Contents of Volume 3
March 1975

Report	Report Number	No. of Leaves	Pub'n. Date
SOFTWARE			
How to Buy Software Packages	70E-010-20	4	10/73
User Ratings of Proprietary Software	70E-010-40	9	8/74
A Buyer's Guide to Data Base Management Systems	70E-010-61	3	11/74
A Buyer's Guide to Data Communications Monitors	70E-010-63	6	12/74
ASI-ST, Applications Software Incorporated	70E-051-01	2	4/74
PI SORT, Applied Data Research, Inc.	70E-052-01	1	7/73
The LIBRARIAN, Applied Data Research, Inc.	70E-052-02	2	2/75
AUTOFLOW II, Applied Data Research, Inc.	70E-052-03	4	9/73
ROSCOE, Applied Data Research, Inc.	70E-052-04	1	12/73
MetaCOBOL, Applied Data Research, Inc.	70E-052-07	2	11/74
SAM (System Analysis Machine), Applied Data Research, Inc.	70E-052-08	2	8/74
Extracto, Aquila BST	70E-058-01	2	9/73
Payroll/Personnel System, Argonaut Information Systems	70E-060-01	1	3/75
Project Control/70, Atlantic Software, Inc.	70E-064-01	1	12/73
QUERY3, Azrex, Inc.	70E-085-01	1	10/73
EZ-Task, Beverly Bancorporation, Inc.	70E-093-01	1	1/74
RELOCATE, Boeing Computer Services, Inc.	70E-094-01	1	10/72
RE-ACT, Boeing Computer Services, Inc.	70E-094-02	1	11/73
Problem Program Evaluator (PPE), Boole and Babbage, Inc.	70E-098-01	1	2/74
Configuration Utilization Evaluator, (CUE), Boole and Babbage, Inc.	70E-098-02	1	2/74
SMS/CAS, Boole & Babbage, Inc.	70E-098-03	1	1/74
SPOOLER, Boothe Management Systems	70E-100-01	1	12/73
DOSRELO, Boothe Computer Corporation	70E-100-02	1	10/72
Fastball-76, Brown Brothers Enterprises	70E-107-01	1	9/73
CROSSTABS, Cambridge Computer Associates, Inc.	70E-123-01	1	12/72
AUTOGRAF, Cambridge Computer Associates, Inc.	70E-123-02	1	2/74
UTILITY-CODER/360, Cambridge Computer Associates, Inc.	70E-123-03	2	2/74
Optimizer II, Capex Corporation	70E-124-01	1	2/74
COTUNE, Capex Corporation	70E-124-02	1	12/73
FORTUNE, Capex Corporation	70E-124-03	1	12/73
Autotab, Capex Corporation	70E-124-04	1	1/74
DUCS-VI, C F S, Inc.	70E-129-01	1	11/73
TOTAL, Cincom Systems, Inc.	70E-132-01	2	11/74
Environ/1, Cincom Systems, Inc.	70E-132-02	2	12/74
3705 Performance Packages, Comm-Pro Associates	70E-158-01	1	10/73
DATACOM/DC, Computer Information Management Co.	70E-180-01	1	1/75
MINIMISER, Computer Services Corporation	70E-206-01	1	10/74
SCERT, Comress, Inc.	70E-242-01	2	12/73
AMIGOS, Comress, Inc.	70E-242-02	1	12/73
PLUS (Program Library Utility System), Cullinane Corporation	70E-272-01	1	2/75
IDMS, Cullinane Corporation	70E-272-02	2	11/74
CULPRIT, Cullinane Corporation	70E-272-03	1	7/73
JASPER, Datachron Corporation	70E-334-01	1	12/73
Data-Man, Data-Man Ltd.	70E-339-01	1	11/73
SAFEGUARD, Digital Solutions	70E-387-01	1	3/74
1130/SORT, DNA Systems, Inc.	70E-396-01	1	10/74
Quantitative Computer Management (QCM), Duquesne Systems, Inc.	70E-398-01	1	8/74
DYL-250 and DYL-260, Dylakor Computer Systems, Inc.	70E-400-01	1	7/73
SWIFT, GBA International	70E-444-01	1	12/74
Multi-DOS2, General Electronics	70E-452-01	1	9/73
Intercomm, GTE Information Systems, Inc.	70E-457-01	2	12/74
Score, GTE Information Systems, Inc.	70E-457-02	1	10/74
Minicomm, GTE Information Systems, Inc.	70E-457-04	1	12/74
ATLAS, GTE Data Services, Inc.	70E-457-21	1	5/74
UCANDU, Gulf Oil Computer Sciences, Inc.	70E-462-01	1	9/73
TLMS, Gulf Oil Computer Sciences, Inc.	70E-462-02	2	5/74
Testmaster/COBOL, Hoskyns, Inc.	70E-488-01	1	11/74
IMS, IBM Corporation	70E-491-01	4	11/74
CICS, IBM Corporation	70E-491-02	4	12/74
Program Products—Applications, IBM Corporation	70E-491-21	14	6/74
Decible III, Independence Computing & Software Corp.	70E-496-01	1	1/74
INQUIRE, Infodata Systems Inc.	70E-498-01	2	1/74
General Ledger System, Infonational	70E-499-01	1	3/75
MARK IV File Management System, Informatics Inc.	70E-500-01	4	3/74
Fast Dump Restore, Innovation Data Processing, Inc.	70E-528-01	1	2/75
PAC 1, International Systems, Inc.	70E-537-01	1	9/73
DOS Job Accounting Report System, Johnson Systems, Inc.	70E-550-01	1	10/74
OS Job Accounting Report System, Johnson Systems, Inc.	70E-550-02	1	10/74
DEEP/360, Macro Services Corporation	70E-591-01	1	10/73
DATAMACS, Management and Computer Services, Inc.	70E-593-01	1	1/74
PULMACS, Management and Computer Services, Inc.	70E-593-02	1	1/74
ALLTAX, Management Information Service	70E-594-01	1	1/75
MSA Payroll/Personnel, Management Science America, Inc.	70E-595-01	2	1/75
Anyplace II, Marcus Powell Associates	70E-603-01	1	10/72
RAMIS, Mathematica, Inc.	70E-610-01	2	11/74
System 2000, MRI Systems Corporation	70E-652-01	2	11/74
Quick-Draw, National Computer Analysts, Inc.	70E-657-01	3	12/73
RSVP, NCI, Inc.	70E-659-01	1	2/74
Work Ten, NCI, Inc.	70E-659-02	1	2/74
SAMSON, On-Line Software, Inc.	70E-674-01	1	2/74
KOMAND Data Acquisition System, Pace Applied Technology, Inc.	70E-676-01	2	10/73
PANVALET, Pansophic Systems, Inc.	70E-677-01	2	2/75
EASYTRIEVE, Pansophic Systems, Inc.	70E-677-02	1	8/74
PAN*SORT, Pansophic Systems, Inc.	70E-677-03	1	10/74
PAN*DA, Pansophic Systems, Inc.	70E-677-04	1	10/74
GRS (General Retrieval System), Program Products Incorporated	70E-691-01	1	10/73
The Data Analyzer, Program Products Incorporated	70E-691-03	1	10/73

NOTE: Please refer to the Index, at the front of Volume 1, for a detailed listing of all the products covered in DATAPRO 70.

datapro
70

Contents of Volume 3
March 1975

Report	Report Number	No. of Leaves	Pub'n. Date
STROBE, Programart Corporation	70E-692-01	1	8/74
Inquiry and Reporting System (IRS), Sigma Data Computing Corp.	70E-753-01	2	7/73
ADABAS, Software AG	70E-757-01	2	11/74
EPAT, Software Design, Inc.	70E-760-02	1	9/74
FMAINT, Software Design, Inc.	70E-760-03	1	4/74
GRASP-Spooling, Software Design, Inc.	70E-760-11	2	11/73
GRASP-Job Accounting, Software Design, Inc.	70E-760-12	1	11/73
GRASP-Relocatability, Software Design, Inc.	70E-760-13	*	11/73
GRASP-Resident Transients, Software Design, Inc.	70E-760-14	1	11/73
GRASP-Partition Balancing, Software Design, Inc.	70E-760-15	*	11/73
GRASP-Dynamic Device Allocation, Software Design, Inc.	70E-760-16	1	11/73
GRASP-Pseudo Clocks, Software Design, Inc.	70E-760-17	*	11/73
MMS General Ledger and Financial Reporting, Software International Corporation	70E-762-01	1	1/75
MMS Accounts Receivable System, Software International Corporation	70E-762-02	1	3/75
SYMBUG, Standard Data Corporation	70E-779-01	1	1/74
PRO/TEST, Synergetics Corporation	70E-793-01	2	10/74
Quikjob I and II, System Support Software, Inc.	70E-798-01	1	8/73
STAGE III, Tesdata Systems Corporation	70E-837-01	1	11/74
CASE, Tesdata Systems Corp.	70E-837-02	2	8/74
DEADLINE II, Tesdata Systems Corporation	70E-837-03	1	8/74
STREAMLINE, Tesdata Systems Corporation	70E-837-04	1	8/74
EDOS, The Computer Software Company	70E-841-01	2	11/74
TASK/MASTER, Turnkey Systems Inc.	70E-866-01	2	12/74
DOS ASAP, Universal Software, Inc.	70E-879-01	1	7/73
RELO-PLUS, Universal Software, Inc.	70E-879-02	1	7/73
UCC TWO, University Computing Company	70E-885-01	1	10/73
UCC ONE, University Computing Company	70E-885-02	1	9/73
UCC TEN, University Computing Company	70E-885-03	1	11/73
UCC FIFTEEN, University Computing Company	70E-885-04	1	11/73
Assembler G, University of Waterloo	70E-886-01	1	11/73
WATBOL, University of Waterloo	70E-866-02	1	12/73
WATFIV, University of Waterloo	70E-866-03	1	12/73
Computer Scheduling and Control System, Value Computing, Inc.	70E-888-01	2	8/74
PHI Payroll II, Wang Computer Services	70E-908-01	2	1/75
DOS Dump/Restore/Plus and Virtual Disk Utility, Westinghouse Electric Corp.	70E-916-01	1	9/74
Teleprocessing Interface System, Westinghouse Electric Corp.	70E-916-02	1	1/75
DOS Job Monitor, Westinghouse Electric Corporation	70E-916-03	1	9/73
SYNCSORT III, Whitlow Computer Systems, Inc.	70E-917-01	1	3/75
FEATURE REPORTS			
How to Get Your Money's Worth with Consultants	70F-050-01	2	1/72
How to Plan and Implement a New Computer Installation	70F-100-01	9	1/72
COMMUNICATIONS			
How to Analyze Your Data Communications Needs	70G-050-01	6	7/71
All About Data Communications Facilities	70G-100-01	14	12/74
All About Communications Processors	70G-400-01	15	10/74
All About Voice Response	70G-420-01	9	9/74
All About Modems	70G-500-01	21	1/75
All About Multiplexors and Concentrators	70G-520-01	10	11/73
All About Remote Computing Services	70G-900-01	20	2/75
SUPPLIERS			
Directory of Suppliers	70H-100-101	62	7/74
MEDIA & SUPPLIES			
All About EDP Media and Supplies	70J-100-01	15	8/74

NOTE: Please refer to the Index, at the front of Volume 1, for a detailed listing of all the products covered in DATAPRO 70.

*Printed on back of preceding report.

datapro 70

the EDP buyer's bible

John R. Hillegass, Executive Editor
Fonnie H. Reagan, Jr., Research Director
David H. Axner, Peripherals Editor
Brian L.J. Callahan, Associate Editor
Mark Flomenhoft, Associate Editor
David M. Hyatt, Associate Editor
Daniel J. Tanner, Associate Editor

John H. Kalbach, Publisher
J. Burt Totaro, Associate Publisher

Robert D. Mervine, General Sales Manager
Charles R. Olson, Western Regional Sales Manager
Robert G. Fritsky, Customer Services Director

CONTENTS

Volume 1—
- Index, section 70A-
- Suppliers, section 70B-
- Computers, section 70C-

Volume 2—
- Peripherals, section 70D-

Volume 3—
- Software, section 70E-
- Feature Reports, section 70F-
- Communications, section 70G-

datapro
research
corporation

1805 UNDERWOOD BOULEVARD
DELRAN, NEW JERSEY 08075
(609) 764-0100

222 W. OSBORN ROAD
SUITE 411
PHOENIX, ARIZONA 85013
(602) 263-7831

Datapro 70 is easy to use . . . and here's how

What is DATAPRO 70?

DATAPRO 70 is a comprehensive hardware/software reference service for EDP management. It's compiled and edited specifically to give you the facts and evaluations you need to understand the computer hardware devices and software packages now on the market and select the ones that best fit your requirements.

What's more, DATAPRO 70 is strongly user-oriented. It's designed to be an everyday working tool–not just a status item on your bookshelf. And it's written in English you can understand–not technical jargon.

How is it organized?

DATAPRO 70 consists of seven major sections, separated and identified by the seven tabs: Index, Suppliers, Computers, Peripherals, Software, Feature Reports, and Communications. A quick scan through any of the sections will show you just what type of information it contains–and how much that information can help you.

When you need facts about specific devices, programs, or concepts, DATAPRO 70's functional Index will guide you quickly to all the pertinent reports. When you get there, you'll find that all the reports have a uniform format. This makes it easy for you to compare the strong and weak points of competitive products and their suppliers.

You'll also find that every DATAPRO 70 product report consists of two major sections: Characteristics and Management Summary. These are arranged in side-by-side columns for your convenience. You'll note that in some reports both sections extend over a number of pages, with the Management Summary in the left-hand column of each page and the Characteristics section in the right-hand column.

The Characteristics section of each equipment and software report gives you all the pertinent details about capabilities, features, configuration, compatibility, pricing, etc. The Management Summary–an exclusive DATAPRO 70 feature– gives you the all-important "big picture": a penetrating evaluation of each device or program. Read them both and you'll have a uniquely clear picture of just what each product is and how you may be able to use it.

What about updating?

Every month you'll receive a supplement that will keep your copy of DATAPRO 70 up to date–and keep you and your staff updated on important new developments in EDP hardware and software.

Each supplement includes easy-to-follow instructions for filing the new and revised reports. (You'll notice that all page numbers in DATAPRO 70 are arranged in straightforward alphanumeric sequence, although many "gaps" have been left in the numbering system to facilitate later insertions.)

How reliable is it?

DATAPRO 70 is compiled and edited by the technical staff of Datapro Research Corporation. The staff consists of experienced analysts, writers, and editors working under the direction of highly qualified professionals in the field of EDP equipment and software evaluation.

All the information in DATAPRO 70 is obtained from the most reliable sources available to our staff. Though the principal source of information for most reports is the manufacturers' own specifications, these are generally clarified and augmented through visits to or correspondence with the manufacturers and (where practical) users of the equipment and software. Prior to publication, reports describing a particular manufacturer's products are normally sent to that manufacturer for review.

Despite all our efforts to keep DATAPRO 70 accurate and up to date, however, the dynamic nature of the computer industry makes it impossible for us to guarantee the accuracy of the information.

What's the best way to use it?

Every reference you make to DATAPRO 70 should begin with the Index. You'll find it behind the Index tab, at the front of Binder 1. It's arranged in straightforward alphabetical order that makes it easy to use.

Naturally, there are Index entries for every company, computer, peripheral device, and software package–and the Index is updated regularly to keep it complete and current. In addition, you'll find "generic" entries for whole classes of equipment and software (such as "key-to-tape recorders" and "data management systems"), followed by the names and locations of all the DATAPRO 70 reports on equipment or software within each class.

Thus, whether you're looking for one specific item or surveying a whole class of equipment or software, the Index is the place to start.

The DATAPRO 70 reports themselves need no explanation. We think you'll agree that they're clear, logical, concise, and helpful. If you need any help in using DATAPRO 70, or if you have any suggestions for making it even more useful, we'll be pleased to hear from you.

How to Buy Software Packages

Here's a set of straightforward guidelines designed to help you understand today's proprietary software market, deal with the vendors, select and install suitable packages, and derive maximum value from your software expenditures.

There are now several thousand separately priced software packages on the market. Packages are available to handle almost every conceivable computer function, at prices ranging from less than $10 to more than $100,000. Some of these packages can probably save you many times their cost, while others will only waste your time and money. Some packages have hundreds of satisfied users, while others have none.

Data processing executives are increasingly recognizing the many potential benefits of proprietary software packages. As a result, sales of standard packages are increasing at the rate of about 30 percent a year and will total an estimated $175 million in 1973. That's more than one-third of the $500 million total market for "outside" (externally supplied) software.

It's easy to understand why proprietary software packages are receiving so much attention from data processing management these days. The costs of in-house programming efforts are climbing behond all reasonable bounds, and it's getting progressively harder to recruit and retain enough competent programmers and analysts. What's more, EDP managers are becoming increasingly aware that there's real money to be saved by using "mass-produced" software wherever practical – and reserving the talents of their in-house programming staffs to concentrate upon the more specialized projects that involve unique requirements or particularly high returns.

The proprietary software industry received a big boost through the "unbundling" moves by IBM and several of the other computer makers. Much of the manufacturer-supplier software that once was "free" is now offered at significant additional costs. As a result, computer users now have far more incentive to "shop around" for the best values for their software dollars.

But how does one shop around for a software package? Is there any rational way to understand today's booming and wildly unstructured software market? How can a buyer be sure that a package will really satisfy his requirements and yield the promised savings? This special report answers these questions and describes an effective acquisition procedure that will help lead you to the best software package for your needs.

Let's Define Our Terms

"Software" can be defined, for our purposes, as the programs that direct computers to perform specific functions. In a larger sense, "software" can also include the whole process of designing, coding, testing, and installing these programs.

A "software package" is a specific computer program or set of programs designed to perform one or more well-defined functions which are considered useful for computer users other than the developer of the package. The package is made available in "canned" form, with associated documentation and maintenance, and is offered either free or at a fixed price.

Though there is an incredible variety of software packages on the market today, all of them can be classified into two basic categories: systems packages and application packages.

"Systems packages" are programs or sets of programs that make it possible to use a computer more conveniently or operate it more efficiently. Included in this broad category are operating systems, data base management systems, report generators, job accounting systems, operating system enhancements, assemblers, compilers, input/output control routines, translators, simulators, diagnostic routines, debugging aids, flowcharters, etc. Systems software is still largely the domain of the computer manufacturers, but independent suppliers are entering the market at an ever-increasing rate and achieving some noteworthy successes. Although only about 25 percent of the currently available software packages fall into the systems category, these systems packages account for more than 50 percent of the total revenues.

"Application packages" are programs or sets of programs that perform certain specific data processing or computational tasks. After getting off to a comparatively slow start, both the computer makers and independent suppliers are now extremely active in the development of packages to handle a broad range of business applications. Payroll packages are by far the fastest-selling type of application package to date, but packages for accounts receivable, accounts payable, general ledger, inventory control, production scheduling, and other common business data processing functions are also being widely accepted. On the engineering and scientific front, application packages are comparatively old stuff; packaged routines to handle matrix inversion, statistical analysis, transcendental functions, and many other common computational tasks have been in widespread use for many years. ▷

How to Buy Software Packages

Sources of Software

There are five basic sources of supply for computer software. Though this report deals primarily with the acquisition of standard, separately priced software packages, it seems appropriate to review the relative advantages and disadvantages of each of the five sources.

1. *In-house development.* Traditionally, nearly all application programming has been done by the user's own staff. This approach has the obvious advantage of giving the user complete control of all aspects of the development process and ensuring that the completed programs will exactly meet the company's needs. But the skyrocketing costs, slipped deadlines, and personnel crises associated with in-house software development are legion, and most data processing managers are now willing to give very serious consideration to the use of software from outside sources.

2. *Computer manufacturers.* Most systems software (assemblers, compilers, operating systems, etc.) has traditionally been supplied by the equipment manufacturers, and more recently many of the manufacturers have developed interesting assortments of application packages as well. Until recently, nearly all of the manufacturer-supplier software was offered at no extra cost, and users were generally inclined to accept it at face value and tolerate delivery delays, frequent bugs, and serious operational inefficiencies. Unbundling changed all that. Users should subject all separately-priced software offered by the computer makers to the same careful scrutiny they give to packages from independent software houses. Even the software that is still "free" should not be used until it has been compared with the available proprietary packages, since the overall costs associated with the installation and operation of a badly-designed free package may far exceed those of a more effective purchased package. Also, users should be fully aware that the computer makers' software facilities are frequently designed to serve as "silent salesmen" by forcing the user to add more storage and/or more peripheral equipment to his system in order to utilize the software.

3. *Users' groups.* Computer users' groups have long performed a useful function by encouraging free exchange of programs developed by thisr members. Unfortunately, these programs are seldom screened or reviewed before they are disseminated. As a result, the quality of the programs and associated documentation tends to be highly variable – but seldom very good. Support for these programs is usually difficult or impossible to come by. Here again, you tend to get what you pay for – and free programs from the users' groups are only rarely worth the time and effort it takes to install and maintain them.

4. *Software houses.* The software field seems destined to be one of the top growth industries of the seventies, and there are some very good reasons for it. Companies that use computers are finding it increasingly difficult, and often economically impossible, to maintain in-house programming groups which are sufficiently large and competent to handle all their software development needs. As a result, they are increasingly looking toward outside suppliers. Many software houses still concentrate on contract programming tasks, in which the programs are custom-designed to meet each customer's specific needs. But an increasingly large number of software suppliers have recognized the potentially greater profits to be gained from "mass-produced" packages, in which the development costs can be spread out over multiple sales. Thus, if a sufficient number of copies can be sold, the supplier benefits from a higher total return on his development costs while the customer benefits from a far lower price (typically 80 to 90 percent lower) than he would have to pay for similar custom-designed programs.

5. *Software brokers.* Numerous companies are now performing a potentially valuable service by acting as brokers between software developers and buyers. Some of their wares are first-class packages, developed specifically for sale to multiple users by independent software suppliers who lack the resources to market them nationally; but others are programs which were initially developed for use in a particular installation and later "jury-rigged" for package sale in the hope of recovering their development costs. Also, some of the brokers are fully staffed to install, support, and maintain the packages they sell, while others look to the original developer to perform these vital support functions. These additional considerations should be kept in mind when surveying the offerings of the software brokers.

Make or Buy?

Before reaching a decision to acquire a software package for a specific purpose, it is advisable to consider carefully the relative costs and benefits of packaged software as compared with those of an in-house development effort. This evaluation process is commonly referred to as a "make or buy" analysis.

Comparing the relative costs involves a straightforward analysis of the total direct and indirect costs of purchasing and installing the package, as described later in this report, versus the total direct and indirect costs of doing the whole job in-house. To make the comparison a realistic one, the true overhead costs associated with all in-house activities, as well as the losses that may be incurred in other areas if the company's programming resources are tied up on this project, should be carefully considered.

How to Buy Software Packages

In addition to the financial picture, the make or buy analysis should include some less tangible considerations such as these:

1. In-house software development projects generally take from a few months to several years to complete and are often subject to serious delays, whereas packaged software can usually be installed and made operational within a month or two.

2. The cost of in-house software development is almost impossible to predict accurately and is often seriously underestimated, whereas packaged software is offered at a fixed price.

3. Some in-house development efrorts never reach completion, for a variety of reasons, whereas packaged software is generally a known quantity.

4. Comprehensive documentation, often sadly neglected in in-house projects, can be demanded as a prerequisite for the sale of most software packages.

5. There may be considerably resistance, in both management and technical circles, to the whole idea of purchasing applications software from outside sources. This resistance can usually be overcome by stressing the clear-cut economic advantages (if any) and the fact that the in-house staff will be relieved of the need to program mundane, commonplace applications and freed to work on more unconventional and challenging projects.

It should be noted that the make or buy decision need not be an absolute one. When a software package does not exactly meet your needs, it will often be far less expensive to buy the package and make the necessary modifications than to "reset to zero" and develop the necessary programs entirely in-house. Moreover, a package can often be used effectively as one component of a larger system or application.

A Suggested Acquisition Procedure

Your chances of finding the right software package and applying it effectively to your requirements will be far greater if you're willing to take the time and trouble to go about it in a systematic, logical manner. The following procedure, while by no means the only one that could rationally be used, is an effective, time-tested recipe that will virtually guarantee a satisfactory solution to your software problems, while protecting you from the huge wastes of time and money that can occur when a company pins its hopes on an ineffectual or unsuitable package.

1. *Determine the requirements.* Just as in the case of in-house software development, the first task is to define the problem to be solved and the environment for its solution. The basic functions to be performed by the package must be clearly defined—usually in terms of the inputs to be provided, the calculations to be performed, the outputs to be produced, and their associated volumes. It is equally important to identify the manner in which the package must interface with all existing systems and procedures. Finally, the environment in which the package will operate needs to be defined in terms of equipment configuration, programming language, operating system, and any other software with which the package must interface.

2. *Gather information about the available packages.* As previously noted, the information required to survey and evaluate software packages can come from a wide range of sources: DATAPRO 70, other trade publications, the software suppliers themselves, consultants, and users of the packages. The specific types of information you'll want to gather are listed in the "Factors to Evaluate" section, which follows.

3. *Narrow the field.* Soon after you begin your survey of the potentially suitable software packages, you'll probably find that some of them simply will not be able to meet your requirements for one or more clear-cut reasons. One may require more core storage or more tape units than your computer provides. Another may turn out to be a "paper tiger" that has yet to demonstrate its effectiveness in any user installation. Still another may be clearly overpriced. These obviously unsuitable packages should be quickly rejected from further consideration so you'll be able to focus all your attention on the real contenders.

4. *Perform a detailed evaluation and comparison.* Apply the tests listed in the "Factors to Evaluate" section, which follows, to each of the competing packages. Decide which of these factors are most important to you, in your application, and weigh them accordingly. Some of these factors must be viewed as "go or no-go" decision points; for example, a package that won't be ready for delivery for six months after your deadline date clearly must be dropped from further consideration. Other factors are much harder to rate objectively; for example, is ease of installation more or less important than the quality of vendor support? Only you, as the user who will ultimately have to live with the package of your choice, can decide.

5. *Talk to users.* The packages which have emerged as leading contenders for your software dollars should now be further checked out by locating and conferring with present users. Ask the software vendor for a list of his customers. If he refuses, take your business elsewhere – he's got something to hide. When you talk with a user, remember that he's not really likely to

admit he made a big mistake by acquiring a bad package. Ask him specific questions about the limitations, delays, errors, and any other problems he's encountered. Ask what modifications he's made, and how much assistance he's received from the supplier. Ask how he thinks the package could and should be improved. Finally, ask for specific details about the performance and throughput he's experienced. The facts and opinions gleaned in this manner will quickly make you a far more enlightened software buyer. Another potentially useful source of user experience is the unique DATAPRO 70 report, *User Ratings of Proprietary Software*, which follows this report.

6. *Conduct benchmark tests.* Preparing and running benchmark tests on your own computer system is by far the most convincing way to assure the selection of a suitable software package. Unfortunately, it is not always a practical thing to do. Designing a really meaningful test, preparing the input data, and evaluating the results can be an expensive, time-consuming process. What's more, the software vendor will often resist the idea of a test on your premises because of both the high cost to him and the attendant security risk. It may be reasonable to waive the requirement for a benchmark test in cases where the package is widely used and has received strongly positive references from its users. Where the package is new and untried, however, a convincing benchmark demonstration should be an absolute requirement and a precondition for any sale.

7. *Make the decision.* Decide which of the available packages will perform the necessary functions and satisfy all your evaluation criteria at the lowest overall cost. Be sure to consider not only the cost of the software itself but all of the accompanying indirect costs: modifications, installation, conversion, training, maintenance, documentation, machine time, etc. Review the previously-discussed "make or buy" criteria to make sure the most economical package truly is a better buy for your company than equivalent software developed in-house. If so, you're just about ready to place the order.

8. *Negotiate a sound contract.* By all means, resist the temptation to simply sign the software vendor's standard contract or order form and get it over with. You'll need to be a fairly tough negotiator to be sure of getting all the protection and support you need, and the assistance of your company's lawyer may be well worth having. You'll find a list of the specific terms and conditions that should be included in a later section of this report called "Contract Negotiations."

9. *Install the package.* Depending upon the complexity of the package and the environment in which it must operate, the installation phase may take anywhere from one day to many months. The vendor will usually consider the installation to be complete when he has supplied the promised documentation, provided the agreed-upon training, and succeeded in making the package run on your computer. The user, however, should not let the vendor "off the hook" until the package is operating at peak efficiency, has delivered bug-free results during at least one complete processing cycle, and has been fully integrated into the overall operational environment. From the user's own viewpoint, the installation may never be complete; there will probably be a continuing need, as with most computer applications, to modify and improve the programs and to train new personnel in their use.

10. *Check the results.* A software package can represent a significant capital expenditure and a vital cog in your company's operations. As such, its performance should be carefully measured to make sure that it is being utilized effectively and is delivering everything you bargained for. Operational efficiencies derived from the use of the package should be analyzed and reported in terms of manhours, machine-time requirements, and overall costs before and after installation of the package. The necessary statistics regarding machine utilization can often be gathered and printed by the package itself or by the associated operating system. Any significant limitations or deficiencies in the package should be identified, and plans made for overcoming them. Finally, a report on the performance and reliability of the software vendor should be prepared and filed for future reference. If the vendor failed to fulfill any commitments, appropriate action should be taken.

Factors to Evaluate

There are numerous criteria which should be carefully considered in evaluating and comparing the merits of specific software packages. The following list, arranged in the form of specific questions, will help guard against the danger of overlooking any significant factors that might make a given package overly costly or otherwise unsuitable for your requirements. You should demand straightforward answers from each of the prospective software vendors, and then try to verify as many of the answers as possible through written agreements, talks with other users, or benchmark tests.

1. *Does the basic function of the package meet your needs?* Are its overall capabilities consistent with the requirements of your application? If you pay most of your employees on a piecework basis, for example, and the payroll package under consideration handles only salaried employees, then no further study will be required. No matter how impressive the package or

How to Buy Software Packages

the salesman promoting it, if it can't meet your basic needs you may as well forget it.

2. *Will the package run on your computer system?* Is a version available for use on your make and model of computer? How much core storage is required? How many disc drives, tape units, card readers, punches, and/or printers? And which models of each? Are any optional hardware features required? How many input/output channels are necessary? Will any off-line equipment, such as key-to-tape data recorders or punched-card tab machines, be needed? If any additional equipment would have to be added to your system to support the package, its cost and availability will naturally need to receive due consideration. If your system configuration is close to the minimum for the package, make sure that hardware limitations will not prevent you from running the programs at peak efficiency or taking advantage of desirable optional facilities.

3. *What are the software requirements?* Can the package be run under the operating system your installation normally uses? Will it interface properly with your input/output control routines and other related software facilities? In what programming language is the package written? Are your programmers familiar with that language, so that they will find the package easy to understand, maintain and modify? Is the language COBOL or FORTRAN, so that the package can be modified for operation on your next computer without undue difficulty?

4. *How well do the detailed capabilities of the package match your requirements?* Are its inputs, computational capabilities, and outputs all consistent with your needs? Is it capable of producing the reports you need? Do the formats and contents of all the input documents, master files, and reports fit your needs—or can they be made to do so without undue cost and difficulty? Are the file organization and data management techniques satisfactory? What input/output and processing options are available, if any? Will special forms be required? What control procedures and audit trails are incorporated? What types of error checks are made, and what happens when errors are detected? What provisions are made for reruns? What about file protection and data security? Only you can judge whether a given package meets—or can be economically adapted to meet—your requirements in each of these important areas.

5. *Is the performance adequate for your needs?* Solid, believable performance figures for software packages are still hard to come by—but essential to have before placing your order. Packages with similar functions and equipment requirements can vary greatly in throughput, and hardware factors such as memory size and tape speed can have major effects on performance. You'll find carefully documented timing data for many of the popular packages right here in DATAPRO 70. Another good source is current users—but make sure their equipment configurations and operational environments are comparable with your own. The very best source of performance data is benchmark tests on your own system—but benchmark testing is an expensive, time-consuming proposition that should be delayed until the field has been narrowed to the two or three strongest contenders.

6. *How flexible is the package?* Are its input, output, and processing capabilities flexible enough to accommodate the changing requirements of your business? Does it include I/O and computational capabilities beyond your present needs, which may be desirable in the future? Are the record lengths and file capacities large enough to meet your requirements for the foreseeable future? Can the system be expanded and/or modified easily to satisfy the needs of the future as well as the present? (It should be noted that this type of flexibility, while highly desirable, usually results in a higher price tag and lower operational efficiency for the package.)

7. *In what form is the package delivered?* For security reasons, some packages are marketed only in object-language form. Whenever possible, however, you should demand a source deck and source-language listing. These make it far easier to understand the inner workings of the package, utilize it effectively, and modify it when necessary.

8. *How difficult will it be to install the package?* What changes will need to be made in your existing systems, procedures, and forms? How many people will be affected, and what type of training or orientation will they need? Will data files need to be converted to new formats and/or media? If so, when and how? How much specialized documentation, in addition to that supplied by the package vendor, will need to be produced? Is it practical to schedule a period of parallel operation, with a gradual cut-over from the old system to the new one? Can the vendor furnish a recommended procedure for the installation and conversion process? How much assistance will the vendor furnish during the process? At what price? While asking these questions, keep in mind that the costs encountered in installing a package often far exceed the direct cost of the package itself.

9. *Will the package be easy to use?* Is it designed for straightforward operation on your computer system, with well-documented operating procedures? Are the input forms and the instructions for preparing them clear and efficient? Are the reports it produces equally clear, comprehensive, and self-evident? Will everyone

How to Buy Software Packages

who is affected by the package find that it satisfies his needs with a minimum of effort on his part?

10. *Is adequate documentation provided?* Good, comprehensive documentation, important in every computer installation, is absolutely vital in the case of proprietary packages, since the writers of the programs will seldom be available when questions and problems arise. A careful check on the quantity and quality of documentation is one of the quickest and most reliable ways to judge the overall quality of a package. In general, four different levels of documentation should be provided, oriented toward the respective needs of the systems analyst, the maintenance programmer, the computer operator, and the package user. Specific elements of documentation that should be available include a comprehensive user's manual, narrative descriptions, system and program flowcharts, source program listings, input document layouts, internal record layouts, report layouts, operating instructions, and instructions for preparing the input and utilizing the output.

11. *What support will the vendor provide?* How much on-site technical assistance is provided under the vendor's standard contract? What specific qualifications will these people have? What is the cost and availability of additional assistance if required? How much training will be supplied for your personnel? What is the nature and level of the training courses, and where and when are they conducted? Will the vendor make any modifications that may be necessary to adapt the package to your specific needs? At what additional cost, if any? Will the vendor correct, without charge, any bugs that are discovered after installation of the package has been completed? For how long? Will the vendor supply improved versions of the package from time to time? What charge, if any, is imposed for continuing maintenance?

12. *What is the operational status of the package?* When, where, by whom, and for what purpose was it originally developed? When was the first installation completed for a user other than the developer? How many companies are currently using the package? One of the biggest potential benefits of a software package is the hope that it has already been tested, debugged, and proven effective in numerous other installations. In many cases this is true. But in many other cases the unwary buyer will commit himself to a package that is still in the early stages of development, with the accompanying likelihood of costly delays and disillusionments. At present, one of the best tests of the quality and effectiveness of a software package is the number of satisfied users—and you can easily find out how satisfied they are by demanding their names and contacting them yourself.

13. *What is the total cost of acquiring and using the package?* This involves careful consideration of both the direct cost (i.e., the price of the package itself) and the indirect costs that may be incurred in modifying the package to fit your requirements, changing existing systems and procedures, training your personnel, converting your files, installing and checking out the package, operating the package on a production basis, and maintaining it after installation. Note that these indirect costs may vary considerably among competitive packages because of differences both in the packages themselves and in their vendors' support policies.

14. *What financial arrangements are offered?* Is the package available for outright sale, for lease, or both? What are the specific terms and conditions for each plan? Are there any objectionable constraints upon the use or modification of the package, or upon the sale of services based on its use? Is a discount offered for multiple installations within the same organization? Are there any extra-cost options? If so, are they worth the asking price, and can they be added at a later date? What additional charges are imposed for support, training, or continued maintenance?

When you've gathered, analyzed, and compared the answers to these questions for each of the packages under consideration, you'll be in a position to make a truly informed, confident buying decision. It's a time-consuming process, of course, but if you're looking for long-term economy and satisfaction, you really can't afford to neglect this type of careful, comprehensive evaluation.

Contract Negotiations

Once you've decided which software package is best for you, it's time to sit down with the vendor and draw up a sound contract (or modify the vendor's standard contract) to ensure that you'll receive everything you've been promised. In general, it's wise to have your company's attorney either sit in during the negotiations or review the final document before you sign it.

Here are some of the key terms and conditions that should be included in the contract when applicable:

1. The guarantee period after installation, during which the supplier will correct all bugs and furnish all required maintenance at no additional cost, should be clearly defined. (This period ranges from 30 days to one year for most of the current packages.)

2. If the package is found unacceptable, the contract should permit its return within a specified period after installation at no cost to the user.

How to Buy Software Packages

▷ 3. A specified penalty should be imposed against the vendor for late or incomplete delivery and installation of the package.

4. If your company will be among the first users of a new package, it may be possible to arrange for a reduced price and/or additional support.

5. All agreed-upon modifications to the package, together with the responsibility and deadlines for making them, should be specified in detail.

6. Objectionable restrictions on the modification or use of the package should be eliminated from the contract.

7. The installation support, training, and maintenance promised by the vendor should be spelled out in detail in the contract, together with the additional costs of these services, if any.

8. The names of the specific individuals who will supply the on-site technical support, or the specific qualifications these individuals will have, should be specified in the contract. (It is particularly desirable in the case of a new, untried package to have the men who developed it present at installation time.)

9. The method to be used for correcting bugs discovered after installation, and the guaranteed response time for correcting them, should be specified.

10. The contract should specify exactly what items, in what forms, will be delivered (e.g., source decks and listings for all programs plus all relevant documentation at installation time, followed by modified or improved versions of the package whenever these are made available to new buyers).

11. The buyer should be willing to agree to reasonable clauses that protect the supplier's proprietary rights by forbidding resale or other unauthorized distribution or use of the package.

12. Payment for the package should be contingent upon delivery of all the promised products and services and successful operation of the package in your installation.

Probably no software package vendor will be willing to make all these concessions; he's in business to make a profit, too. But by playing the role of a tough, sophisticated negotiator and demanding everything you can get, you stand to save a lot of money and get far better support than the buyer who meekly signs the standard contract form.

Summing It Up

Every company that wants to receive maximum value for its data processing expenditures should certainly be taking a long, hard look at proprietary software packages these days. Thousands of packages are now available from hundreds of suppliers, and the wise buyer will shop around very carefully before committing himself to any one package.

A suggested 10-step acquisition procedure that will help to ensure the selection and effective utilization of a truly satisfactory, economical package can be summarized as follows:

- Determine the requirements.
- Gather information about the available packages.
- Narrow the field by rejecting unsuitable packages.
- Perform a detailed evaluation and comparison (see "Factors to Evaluate").
- Talk to users.
- Conduct benchmark tests.
- Make the decision.
- Negotiate a sound contract (see "Contract Negotiations").
- Install the package.
- Check the results. □

User Ratings of Proprietary Software

This first-of-its-kind report capsulizes the experience of 191 users with a total of 174 proprietary software packages. Detailed comparison charts pinpoint the strengths and weaknesses of 40 popular packages rated by 3 or more users. The results should help you get more for your software dollars.

Datapro won't attempt to fix a precise definition for a term as broad as proprietary software for the EDP industry. Instead, this report uses the loose definition that any software that a user pays for, whether it comes from his mainframe manufacturer or an independent vendor, is proprietary software.

Before putting his money on the line for separately priced software, a computer user deserves to know—and should demand to know—how that software is performing in other user installations. But this type of information is normally obtainable only through the arduous and frequently neglected process of reference checking. To help fill this important information gap and guide software buyers toward packages of proven reliability, Datapro Research Corporation conducted the extensive user survey whose results are presented in this report.

Why Use Proprietary Software?

There are many reasons why a user may wish to obtain a proprietary software package. Some of the main ones are:

- Avoidance of an in-house effort.
- Standardization.
- Improvements in the speed or efficiency of program execution.
- Improvements in the control, speed, or ease of system operation.

Avoiding an in-house effort can mean avoiding the hiring of additional programmers or even reducing the size of a programming staff. No wonder programmers call some proprietary software "out-house efforts." Avoiding an in-house effort in order to implement a user-specific application or to solve specific problems is usually one of the least justifiable reasons for buying a proprietary package. Not only can there be considerable resentment from programmers, but the intended application, being user-specific, will probably require modification of the package. This can cause problems with the user's staff and the vendor, and can even cause management to wonder why the package they authorized purchasing hasn't produced the desired results. Further, does the user then dismiss his programming staff and face the inevitable maintenance requirements alone, or should he keep them on and give them "busy-work"? For these and other reasons, more than half of today's proprietary software dollars are going to the approximately 25% of the software vendors who produce general-use packages.

Avoiding an in-house effort can be a valid decision when a general-use need can be met. Most of these general uses fall into the categories of standardization, execution improvements, and operational improvements, and are discussed below. It is in these areas where the vendors have already done the work, making it unwise for a user to re-invent the wheel.

Standardization is often a good reason for a user—especially a user with multiple computer systems at various sites—to purchase proprietary application software. This can guarantee that a particular application, such as accounts receivable, will be processed the same way regardless of location. Additionally, standardization at one or more computer sites can be achieved through the use of proprietary systems or operations software, such as library systems, documentation aids, languages, shorthands, and some accounting and reporting systems. Proper standardization can improve communication among personnel and systems, enable data to be transferred among systems, and permit simplified, standardized maintenance of programs.

Proprietary software used to improve the execution speed of a common application must save the user at least as many dollars worth of computer time as it costs to be worthwhile. Prime examples of packages designed to do this are sorts, utilities, data manipulators, report writers and generators, language optimizers, and data base management systems.

Some proprietary software packages are designed to speed or ease the way in which a system is operated. By their nature, they may also serve to standardize system operations. Additionally, there are packages designed to improve the control management has over computer operations. The various operating system enhancements, accounting packages, library systems, documenters, some reporters, and even some sorts and utilities fall into the group providing services in all of these areas. Cost justification for many of these packages, like that for standardizing packages, can be difficult to assess. Often, a trial is the only way to judge. Can a shift be eliminated when the package is used? Are costly operator errors being reduced? Is the number of production job reruns going down? Are deadlines being met when they weren't before? Do you need an evaluation of whether the deadlines themselves are reasonable? (There are packages for that, too).

User Ratings of Proprietary Software

⊳ What Datapro Asked, and Why

In order to learn and report user experience with proprietary software, Datapro included a Reader Survey Form in the May 1973 supplement to DATAPRO 70. On the postpaid form, we asked users to summarize their experience with proprietary software, including any separately priced software from their mainframe vendor, and return the form by July 1. Subscribers were invited to copy the questionnaire if they wished to report on more than three packages, and specific comments were encouraged. The response was gratifying: Datapro received returns from 191 users. These subscribers reported on the use of 174 different packages. The average number of packages rated by each responding user was 2.7, yielding a total of 467 ratings of packages in use.

We asked when use of the package was begun, so that we could consider the differences in judgments, if any, of those who had been using a package for a long time versus those of the newcomers. As it turned out, such differences were uncommon.

In response to our question as to the make and model of the user's computer system, over 95% of the respondents turned out to be using an IBM System/360, Model 22 or larger, or a System/370, Model 135 or larger. Besides confirming the obvious – most of the proprietary software market is in the 360 and 370 field – the answers helped to show what types of software users of different-sized systems are seeking.

The next question, "How much main storage is used by the package?", provided illuminating comparisons of actual main storage requirements with the figures specified by the package vendors.

After the preliminary questions came the ratings.

Five subjective rating questions were asked, with the answers "excellent," "good," "fair," and "poor" available to be checked. The questions were: "How would you rate the package with respect to (a) your overall satisfaction, (b) job throughput/program efficiency, (c) ease of installation, (d) documentation, and (e) vendor's technical support?" For the programs mentioned by three or more users, their detailed ratings are presented in the accompanying comparison charts.

Following the subjective questions, the questionnaire contained two objective checkoff questions: "Did the package perform as advertised – immediately, eventually, or never?" and "Did the package require modification – not at all, by the vendor, or by the user?" Again, the results are presented in the comparison charts.

Datapro then asked two questions designed to support and amplify the checkoff questions. These were: "Why are you using the package (i.e., principal advantages)?" and "Please note any significant disadvantages of the package." The essence of the replies to these questions is concisely presented in the comments at the bottom of the comparison charts.

The final question was "Please note any other packages you tried for this job and rejected, and explain why." The answers helped to gauge each package's competitive strength, and will increase the value of Datapro's telephone consulting service.

The remainder of the questionnaire asked the user's identity and whether we could contact him for additional information. All but three of the respondents indicated that we could contact them, and those that we did contact proved most helpful. Only signed questionnaires were counted as valid, but only three were returned unsigned.

One user signed a questionnaire and left all but the comment space blank. In this space, he indicated the wish that we'd surveyed users of in-house programs. That reminds us to remind you that we'd be pleased to learn just what additional subjects you'd like surveyed.

What Was Learned?

As of August 1973, DATAPRO 70 contains reports on 102 software packages, not counting the voluminous IBM Program Products report (70E-491-21). In our survey, a total of 40 packages were rated by 3 or more users; of these 40, 31 are covered in individual Datapro reports, 8 are contained within a Datapro report on IBM, and only 1 is no longer covered in DATAPRO 70. Of the 31 packages with 2 users reporting, 13 are covered in individual Datapro reports, 2 are contained in Datapro reports dealing with IBM, and 15 do not currently appear in DATAPRO 70. Finally, the packages rated by only 1 user number 103, with 28 of them covered in individual Datapro reports and 7 described in Datapro reports on IBM.

We considered it appropriate to make up comparison charts on the 40 packages reported on by 3 or more users. These charts are instructive in that they show the raw rating tallies, a computed "grade point average" for each subjective rating category, and brief comments by the reporting users and/or the Datapro staff.

The grade point averages were computed in a manner familiar to most who have been to college: excellent is weighted as 4, good as 3, fair as 2, and poor as 1. Then, the tallied number of each response is multiplied by the corresponding weight, and an average is taken by dividing the sum of the products by the number of responses totalled. ⊳

User Ratings of Proprietary Software

The Top 40

Based on 2.0 (or an average rating of "fair") as passable, how do the 40 packages rated by 3 or more users fare? Only one, IBM's COGS, "fails." On the other hand, 17 of the 40 packages were rated 3.5 or above in the important category of overall user satisfaction, and can be termed "outstanding" on the basis of this survey. The 17 packages on this "Datapro honor roll" include: ALLTAX (Management Information Service), AMIGOS (Comress, Inc.), DUMP/RESTORE/COPY (Westinghouse Tele-Computer System Corp.), DUO 360/370 (whose name is being changed to UCC TWO by University Computing Company), DYL-250 (Dylakor Computer Systems, Inc.), EASYTRIEVE (from Ribek Corporation, but now marketed by Pansophic Systems, Inc.), EPAT and GRASP (Software Design, Inc.), The LIBRARIAN (Applied Data Research, Inc.), PANVALET (Pansophic Systems Inc.), POWER (IBM), QUIKJOB (System Support Software, Inc.), RPG II (IBM), SCORE (Programming Methods, Inc., a GTE Subsidiary), SPOOLER (Boothe Computer Corporation), SyncSort (Whitlow Computer Systems), and TOTAL (Cincom Systems, Inc.). For the detailed ratings of these and the rest of the "top 40" programs, please turn to the comparison charts that follow.

The Others

Observant readers may have noticed that all of the packages rated by 3 or more users run on the IBM System/360 and 370 (and may also happen to run on other systems as well). This is almost equally true of the packages with 2 reporting users; only 3 respondents were using computers other than 360's or 370's. We've placed the 31 packages rated by 2 users in the following listing, arranged alphabetically by package names, with the grade point average (GPA) on overall satisfaction and, where applicable, the DATAPRO 70 report number.

Name & Vendor	GPA	Report Number
ADPAC (Adpac Computing Languages)	4.0	70E-027-01
COBOL ANS Subset (IBM)	2.5	70C-491-04
Coursewriter III (IBM)	1.5	70E-491-21
CS^2 (Value Computing)	3.5	70E-888-01
Data Analyzer (Program Products, Inc.)	2.5	70E-691-03
DATAMACS (Management & Computer Services)	3.0	70E-593-01
DEEP/360 (Macro Services Corp.)	3.5	70E-591-01
DOSRELO (Boothe Management Systems)	3.0	70E-100-02
DOSJARS (Compunetics Corp.)	2.0	–
DSI COBOL (Decision Systems)	3.5	70E-364 - 01
EXCHECK (Arkay)	4.0	–
EXTRACTO (Aquila BST)	4.0	70E-234-01
FASTBALL-71 (Brown Brothers Enterprises)	3.5	–
Financial Info. Control System (Management Science America)	3.0	–
Fixed Assets (Management Science America)	2.5	–
FORTRAN H Extended (IBM)	2.5	70C-491-04
GPSS II (IBM)	3.0	70E-491-21
MPSX (IBM)	3.0	70E-491-21
MULTI-DOS (General Electronics)	4.0	–
PACES (Pace Applied Technology)	4.0	–
PAC-1 (International Systems)	3.0	–
Payroll (PHI Computer Services)	3.0	70E-684-03
Personal Trust Accounting System (General Computer Services)	2.5	–
PI SORT (Applied Data Research)	3.0	70E-052-01
Project Control/70 (Atlantic Software)	2.5	–
QUERY 3 (Azrex)	3.0	–
SELF-RELO (Webster)	3.0	–
SIMSCRIPT (CACI)	3.0	–
SPSS (CHI Corporation)	3.5	–
Tape Management Software (UCC)	3.5	–
UCANDU (Gulf Oil Computer Sciences)	3.5	–

The Data Analyzer and Project Control/70 would have had three reporting users had all the forms been signed. QUERY 3 is running on a CDC 3300 and a Honeywell 2060, while SPSS has a UNIVAC 1108 user. EXTRACTO was developed and is marketed by Aquila BST, a Canadian subsidiary of Cordura Corporation, formerly Computing and Software.

We Know You're Out There; We Can Hear You Breathing

The remainder of the packages mentioned by users were reported on by only one individual each. Nonetheless, we can at least learn what's in the field – and that can often be valuable knowledge. After all, a little information can be valuable information when it's the only information you've got. The packages are listed below, along with each one's E-G-F-P rating on overall satisfaction from the lone reporting user and an occasional report number. The detailed ratings remain in Datapro's files.

Name & Vendor	Rating	Report Number
Accounts Receivable (Software International)	G	70E-762-02
Accounts Receivable (GTE Data Services)	F	–
Accounts Receivable for S/3-10 (IBM)	E	70E-491-21
Accounts Receivable (People's Merchant Trust Co., Canton, OH)	F	–
APL/360 (IBM)	E	70C-491-04
APL*PLUS (Scientific Time Sharing)	E	

User Ratings of Proprietary Software

Name & Vendor	Rating	Report Number
Assembly G (Univ. of Waterloo, Canada)	E	–
Assembly H (IBM)	E	70C-491-04
BASIC (under ITF) (IBM)	G	70C-491-04
Business Management System (Computer Sciences Corp.)	G	–
CalComp Plotter Subroutines (CalComp)	E	–
CATALR (Marcus Powell Associates)	G	–
CFMS (IBM)	F	70E-491-21
COBOL for IBM 1130 (IBM)	E	70C-491-11
COBOL X-REF (Hansco Data Processing)	E	–
COBOL-EZE (General Systems Corp.)	E	–
COMPUMETER (CEI Corp.)	E	–
Controller (Honeywell) – on Series 200	E	–
CORE SETL (Boothe)	G	–
CRJE (McGill Univ., Canada)	E	–
CROSSTABS (Cambridge Computer Assoc.)	G	70E-123-01
CRT Interface (Westinghouse)	–	–
CULPRIT (Cullinane)	G	70E-272-03
CustomAR (Uniroyal Computeristics)	E	–
DATAMAN (Page Raymond Associates)	G	–
DECIBLE III (Independence Computing & Software)	E	–
Demand Deposit Accounting (American National Bank & Trust, Chattanooga)	F	–
Device-Independent Routines for 3330 (IBM)	E	70C-491-04
DISKPLAY (Miniature Software Products)	E	–
DISPLAYALL (Informatics)	–	–
DITTO (IBM)	G	70E-491-21
DOCUMATIC (Data Usage Corp.)	F	70E-352-01
DOS RELOCATE (Boeing Computer Services)	E	70E-094-01
DYL-260 (Dylakor)	G	70E-400-01
DYNAMO II (Pugh-Roberts Assoc.)	G	–
Fast Dump/Restore (Innovation DP)	G	70E-528-01
FASTER-LC (IBM)	G	70E-491-21
FASTER-MT (IBM)	P	70E-491-21
FAST/MASTER for CDC 3000 (Advanced Computer Techniques)	G	–
File Utility (Westinghouse)	E	–
Financial Control System (UCC)	E	–
Fixed Assets (American Appraisal)	F	–
Fixed Assets (McCormack & Dodge)	–	–
FLOWCHART (Westinghouse)	E	–
FORECASTER (Honeywell)–on Series 200	E	–
FORTRAN G (IBM)	F	70C-491-04
GIS-2 (IBM)	P	70E-491-21
IMSL (International Math. & Statistical Libraries, Inc.) – on UNIVAC 1108	E	–
INQUIRE (Infodata)	G	70E-498-01
Installment Loan Accounting System (Cullinane)	E	–
Installment Loan Accounting (Florida Software Services)	E	–
Installment Loan Accounting System (Kranzley & Co.)	E	–
Inventory Control (IBM) – for S/3-10	G	70E-491-21
IRS (Sigma Data)	F	70E-753-01
Item Processing System (Burroughs) – on B 3500	G	–
JASPER (Datachron Corp.)	E	–
Job Accounting (Johnson Data Systems)	G	70E-550-01
Job Monitor, DOS (Westinghouse)	E	–
Letter Writing (Franklin Data Systems)	P	–
Long-Range Planning System (IBM)	E	–
MARGEN (Randolph Data Services)	E	–

Name & Vendor	Rating	Report Number
MINIMIS (Service Bureau Corp.)	G	–
Miracl/COBOL (Republic Software)	G	70E-724-01
MIRADS (NASA) – for UNIVAC 1108	G	–
Mortgage 70 (Sys Con, Inc.)	F	–
OASIS (Stanford Univ., California)	E	–
PANMASTER (Pansophic Systems)	E	–
PAS-1 (Brady-Tower, Inc.)	P	–
Payroll (Florida Software Services)	G	–
Payroll (Generated Systems, Inc.)	G	–
Payroll (IBM) – for S/3-10	–	70E-491-21
Payroll (NCR) – on NCR Century 200	F	–
PAY-O (Phillip Hankins)–on Honeywell 6040 and G-415 of 1 user	E	–
PL/1 (IBM)	F	70C-491-04
PL/1 under ITF (IBM)	G	70C-491-04
PLS (ISC/Pryor Computer Industries)	G	–
PLUS (Cullinane)	E	70E-272-01
PMS (IBM)	G	70E-491-21
PRINCE (Syracuse Univ.)	G	–
PROBE (Computer Resources Corp.)	G	–
Profit-Sharing Accounting (Trust Management Systems)	G	–
Proof of Deposit/Transit (IBM)	E	–
PRO-TEST (Synergetics Corp.)	F	70E-793-01
QUICKDRAW (National Computer Analysts)	G	70E-657-01
QUICKSCAN (Applied Systems Associates)	E	–
RAVE (M. Bryce & Associates)	G	–
Requirements Planning (IBM)	E	70E-491-21
RFCS (University Computing Company)	G	–
Rural Electric Billing (IBM)–S/3	P	70E-491-21
SAV-A-MATIC (Xduc, Inc.)	F	–
Savings Accounting (Florida Software Services)	E	–
SMS/CAS (Boole & Babbage)	G	–
SOFTWARE-1040 (SAB, Inc.)–for S/3, card	E	–
SORT (DNA Systems)–for IBM 1130	E	–
Stockholder Accounting (MSA)	E	–
Stockholder System, Corporate Shareholder (Programming Methods)	F	70E-694-03
SUPDOS-2 (Universal Software)	E	–
SYSTEM/2000 (MRI Systems Corp)–on CDC Cyber 70	E	70E-652-01
TEST PAK (Computer Methods)	G	70E-188-01
Trust Automation System (Ernst & Ernst)	E	–
Trust Package (Wisconsin First National Bank)	G	–
VIGOR (Arkay)	E	–
VISAM (Wieland Computer Group)	G	–

What's Next?

Thanking the 191 users who responded to our survey, we turn to the future. We think that a survey of this type is helpful to all concerned: it informs and aids our subscribers, increases our own expertise, and generates unbiased feedback to package vendors. We plan to make this survey an annual DATAPRO 70 feature, and we look forward to receiving your input next year. In fact, if you've had direct experience that confirms, augments, or contradicts the user ratings in this report, we'd be delighted to hear from you at any time. Just write or phone us at your convenience. □

User Ratings of Proprietary Software

PACKAGE NAME	ALLTAX	AMIGOS	ASAP	ASI-ST	AUTOFLOW
Vendor	Management Information Service	Comress, Inc.	Universal Software, Inc.	Applications Software, Incorporated	Applied Data Research, Inc.
Average memory used	19K	10K (based on 1 report)	10K	82K	70K
Overall satisfaction					
Excellent	4	2	1	1	5
Good	0	1	2	0	3
Fair	0	0	0	1	4
Poor	0	0	0	1	1
Grade point average	4.0	3.7	3.3	2.3	2.9
Throughput/efficiency					
Excellent	3	2	2	1	5
Good	1	1	0	0	4
Fair	0	0	0	1	3
Poor	0	0	0	1	1
Grade point average	3.8	3.7	4.0	2.3	3.0
Ease of installation					
Excellent	4	2	2	1	7
Good	0	1	0	2	5
Fair	0	0	0	0	1
Poor	0	0	0	0	0
Grade point average	4.0	3.7	4.0	2.6	3.5
Documentation					
Excellent	3	1	0	0	6
Good	1	2	2	1	3
Fair	0	0	0	2	4
Poor	0	0	0	0	0
Grade point average	3.8	2.7	3.0	2.3	3.2
Vendor technical support					
Excellent	2	2	1	0	5
Good	0	1	1	1	3
Fair	0	0	0	0	3
Poor	0	0	0	2	1
Grade point average	4.0	3.7	3.5	1.6	3.0
Perform as advertised?					
Immediately	4	1	3	1	10
Eventually	0	2	0	2	2
Never	0	0	0	0	0
Require modification?					
No	3	1	3	1	6
Yes, by vendor	0	1	0	1	1
Yes, by user	1	0	0	0	5
Number of users reporting	4	3	3	3	13
Comments	COBOL subroutine; can be incorporated into most payroll programs, yielding automatic tax table maintenance and tax calculations; especially useful when employees are taxed by several different authorities	Offers better storage utilization and disk access speed to ISAM users reporting	Users like the low core and disk requirements of this DOS spooling package, but one wishes that it were possible to group and print spooled reports by forms type	Data base system that can handle IMS files; some user statements indicated limited design and inadequate vendor support	Automatically generates flowcharts of source-level programs and produces a variety of other documentation aids; those using the package as an enforced standard seem most pleased with it
Report number	70E-594-01	70E-242-01	70E-879-01	70E-051-01	70E-052-03

User Ratings of Proprietary Software

PACKAGE NAME	CICS (Customer Information & Control System)	COBOL (ANS)	COGS (Consumer Goods System)	CUE & PPE	DATA/360
Vendor	IBM	IBM	IBM	Boole and Babbage, Inc.	IBM
Average memory used	124K	54K	69K	41K	64K
Overall satisfaction					
Excellent	1	6	0	3	0
Good	7	12	1	5	3
Fair	7	0	0	0	0
Poor	0	0	2	0	0
Grade point average	2.6	3.3	1.3	3.4	3.0
Throughput/efficiency					
Excellent	1	6	0	2	3
Good	8	8	0	4	0
Fair	5	3	0	0	0
Poor	0	0	2	0	0
Grade point average	2.7	3.2	1.0	3.3	4.0
Ease of installation					
Excellent	1	4	0	4	0
Good	0	8	0	2	3
Fair	6	7	0	2	0
Poor	7	0	2	0	0
Grade point average	1.5	2.8	1.0	3.6	3.0
Documentation					
Excellent	0	5	0	1	0
Good	6	9	0	7	3
Fair	7	4	1	0	0
Poor	2	1	1	0	0
Grade point average	2.3	2.9	1.5	3.1	3.0
Vendor technical support					
Excellent	0	7	0	2	2
Good	7	3	0	5	0
Fair	5	8	1	1	1
Poor	3	0	1	0	0
Grade point average	2.3	2.9	1.5	3.1	3.3
Perform as advertised?					
Immediately	2	9	0	6	1
Eventually	10	8	1	2	2
Never	1	0	2	0	0
Require modification?					
No	3	7	0	4	0
Yes, by vendor	8	8	0	3	1
Yes, by user	9	2	3	1	3
Number of users reporting	14	18	4	8	3
Comments	IBM's major communications monitor, available in DOS entry level, DOS, OS, and VS versions; users praise its ease of use but complain about memory required, bugs, and lack of COBOL support	Widely used source language and compiler for the IBM System/360 and 370; surprisingly, no user mentioned the SORT verb or CROSS-REF facility	Users report package requires excessive runs and "blows up"; one user reported as an ex-user without giving ratings	Configuration Usage Evaluator (CUE) and Problem Program Evaluator (PPE) aid in system tuning and monitoring	IBM on-line key-to-disk data entry system; 2 users dislike disk utilization
Report number	70E-491-02	See 70C-491-04	See 70E-491-21	70E-098-01 and -02	See 70C-491-21

User Ratings of Proprietary Software

PACKAGE NAME	DBOMP (Data Base Organization & Maint. Processor)	DUMP/RESTORE/ COPY	DUO 360/370 (being changed to UCC TWO)	DYL-250	EASYTRIEVE
Vendor	IBM	Westinghouse Tele-Computer Systems Corp.	University Computing Company	Dylakor Computer Systems, Inc.	Ribek Corporation
Average memory used	32K	20K	60K	18K	36K
Overall satisfaction					
Excellent	2	24	4	4	3
Good	4	3	0	1	0
Fair	0	0	0	0	0
Poor	1	0	1	0	0
Grade point average	3.0	3.9	3.6	3.8	4.0
Throughput/efficiency					
Excellent	2	26	4	4	3
Good	4	1	0	0	0
Fair	1	0	0	1	0
Poor	0	0	1	0	0
Grade point average	3.1	4.0	3.6	3.6	4.0
Ease of installation					
Excellent	0	23	3	5	3
Good	2	3	1	0	0
Fair	3	1	0	0	0
Poor	2	0	1	0	0
Grade point average	2.0	3.8	3.2	4.0	4.0
Documentation					
Excellent	2	20	3	0	0
Good	3	7	1	4	0
Fair	1	1	0	0	2
Poor	1	0	1	0	1
Grade point average	2.4	3.7	3.2	3.0	1.7
Vendor technical support					
Excellent	2	21	3	4	3
Good	1	7	1	1	0
Fair	3	0	1	0	0
Poor	0	0	0	0	0
Grade point average	2.8	3.8	3.4	3.8	4.0
Perform as advertised?					
Immediately	6	26	4	4	3
Eventually	2	1	1	1	0
Never	0	0	0	0	0
Require modification?					
No	4	24	4	5	3
Yes, by vendor	1	1	1	0	0
Yes, by user	3	2	0	0	0
Number of users reporting	7	27	5	5	3
Comments	Users say package is modular and easy to use, but slow and has many problems using 3330 disks; some say they use DBOMP only because they see no alternative	Users repeatedly state that this DOS utility package is much faster than the IBM equivalent; the package is an industry leader in number installed	Facilitates DOS-OS conversion by allowing DOS programs to run under OS with access to most OS facilities; one user complains that the interface between OS and the user program isn't well documented	DYL-250 is a popular "quickie" utility program generator with limited editing and calculating capabilities for one-shot reports; it is a $1-per-day package whose shortcomings are largely taken care of in the companion DYL-260 $2.60-per-day package	Simplified, modest-cost information retrieval system now being marketed by Pansophic Systems, Inc.; it earned perfect grades except for the sub-par documentation ratings
Report number	See 70E-491-21	70E-916-01	70E-885-01	70E-400-01	70E-727-01

User Ratings of Proprietary Software

PACKAGE NAME	EDOS	ENVIRON/1	EPAT	GEN'L LEDGER SYSTEM (formerly MARS)	GRASP
Vendor	The Computer Software Company	Cincom Systems, Inc.	Software Design, Inc.	Management Science America, Inc.	Software Design, Inc.
Average memory used	37K (based on 1 report)	56K	No estimate available	123K	25K
Overall satisfaction					
Excellent	3	0	3	0	13
Good	1	1	0	2	2
Fair	0	1	0	1	0
Poor	1	0	0	0	0
Grade point average	3.2	2.5	4.0	2.7	3.9
Throughput/efficiency					
Excellent	2	1	3	0	13
Good	2	0	0	2	2
Fair	0	1	0	1	0
Poor	0	0	0	0	0
Grade point averaeg	3.5	3.0	4.0	2.7	3.9
Ease of installation					
Excellent	3	1	2	0	12
Good	1	1	1	2	3
Fair	0	0	0	1	0
Poor	1	0	0	0	0
Grade point average	3.2	3.5	3.7	2.7	3.8
Documentation					
Excellent	3	0	3	1	8
Good	1	1	0	0	6
Fair	1	1	0	2	1
Poor	0	0	0	0	0
Grade point average	3.4	2.5	4.0	2.7	3.5
Vendor technical support					
Excellent	3	0	3	0	12
Good	1	0	0	1	3
Fair	0	2	0	1	0
Poor	1	0	0	1	0
Grade point average	3.2	2.0	4.0	2.0	3.8
Perform as advertised?					
Immediately	2	1	2	0	12
Eventually	1	1	1	2	3
Never	1	0	0	0	0
Require modification?					
No	1	1	2	0	10
Yes, by vendor	3	1	0	1	2
Yes, by user	1	0	1	2	3
Number of users reporting	5	3 (only 2 gave ratings)	3	3	15
Comments	EDOS (Extended DOS), a collection of DOS enhancements, can include 6-partition support and spooling; it now offers S/370 simulation on the S/360; one displeased user had unique problems; EDOS is used by the Greyhound Phoenix System	Users like the TOTAL interface on this communications monitor, its COBOL capability (not found in IBM's CICS), and its modest core requirements	EPAT (tape spelled in reverse) can help a user keep track of physical tape reels; users praise its automatic volume recognition capability	General ledger package with a comprehensive reporting system; users view it as an application shortcut that has flexibility, but comment that the sequential data base is undesirable and that the package is designed mainly for banking	Widely used spooling supplement for DOS; users rate it faster and more efficient than IBM's POWER and the spooler in EDOS; some also like the fact that GRASP can run in its own FO partition; some users complain about the price tag
Report number	70E-841-01	70E-132-02	70E-760-02	70E-595-01	70E-760-01

User Ratings of Proprietary Software

PACKAGE NAME	IMS (Information Management System)	INTERCOMM	The LIBRARIAN	MARK IV	MURS (Machine Utilization Reporting System)
Vendor	IBM	Programming Methods, Inc. (GTE subsidiary)	Applied Data Research, Inc.	Informatics, Inc.	Webster Computer Corporation
Average memory used	273K (varies widely)	132K	63K	103K (5 models)	0.5K
Overall satisfaction					
Excellent	0	1	13	6	0
Good	4	4	5	7	1
Fair	2	1	0	0	4
Poor	0	1	0	0	0
Grade point average	2.7	2.7	3.7	3.5	2.2
Throughput/efficiency					
Excellent	1	2	14	3	0
Good	1	4	4	5	2
Fair	3	1	0	4	2
Poor	1	0	0	1	0
Grade point average	2.3	3.1	3.8	2.8	2.5
Ease of installation					
Excellent	2	0	12	5	0
Good	1	2	6	7	2
Fair	3	3	1	0	2
Poor	0	2	0	0	1
Grade point average	2.8	2.0	3.6	3.4	2.2
Documentation					
Excellent	1	1	8	4	0
Good	1	1	7	6	0
Fair	3	3	4	1	4
Poor	1	2	0	2	1
Grade point average	2.3	2.1	3.2	2.9	1.8
Vendor technical support					
Excellent	1	1	8	2	0
Good	2	3	6	6	1
Fair	3	2	2	5	2
Poor	0	1	0	0	2
Grade point average	2.7	2.6	3.4	2.8	1.8
Perform as advertised?					
Immediately	2	0	17	9	1
Eventually	4	7	1	4	4
Never	0	0	0	0	0
Require modification?					
No	5	1	15	8	0
Yes, by vendor	0	6	2	4	3
Yes, by user	1	5	1	1	4
Number of users reporting	6	7	18	13	5
Comments	IBM's principal data base management system; has more power than Cincom's TOTAL, but users report excessive core usage and poor response time; IMS is available under OS; a data communications feature (1 user reporting) is available; the DOS version is called DL/1	Users report that this data communications monitor has a flexible data base interface, good output formatting capability, and is smaller than some competitive packages; they also say it is relatively slow, subject to failures, and usually requires modification	Widely used system to maintain source programs on tape rather than on cards; users praise SCAN facility, security and backup provisions, disk space saving features, and control given to management	A powerful file processing system that has users praising its ease of use, even for non-DP personnel; but some feel it is too slow and uses too much memory; MARK IV comes in 5 versions with different capabilities and prices	Provides data collection and reporting on DOS processing activities
Report number	70E-491-01	70E-457-01	70E-052-02	70E-500-01	—

User Ratings of Proprietary Software

PACKAGE NAME	OPTIMIZER	PANVALET	PAYROLL	PL/1 OPTIMIZER	POWER
Vendor	Capex Corporation	Pansophic Systems, Inc.	Management Science America, Inc.	IBM	IBM
Average memory used	100K (based on 1 estimate)	45K	67K	90K	45K
Overall satisfaction					
Excellent	1	20	0	1	2
Good	2	9	4	3	1
Fair	0	0	2	1	0
Poor	0	0	1	1	0
Grade point average	3.3	3.7	2.4	2.7	3.7
Throughput/efficiency					
Excellent	1	17	0	2	1
Good	2	11	4	3	2
Fair	0	1	2	1	0
Poor	0	0	1	0	0
Grade point average	3.3	3.6	2.4	3.2	3.3
Ease of installation					
Excellent	2	18	0	1	2
Good	1	10	3	2	0
Fair	0	1	2	1	1
Poor	0	0	2	2	0
Grade point average	3.7	3.6	2.1	2.3	3.3
Documentation					
Excellent	0	14	1	0	1
Good	2	14	0	3	1
Fair	1	2	4	3	1
Poor	0	0	2	0	0
Grade point average	2.7	3.4	2.0	2.5	3.0
Vendor technical support					
Excellent	0	13	1	0	2
Good	2	13	2	1	1
Fair	1	4	1	3	0
Poor	0	0	3	2	0
Grade point average	2.7	3.3	2.1	1.8	3.7
Perform as advertised?					
Immediately	3	28	3	0	1
Eventually	0	0	3	4	2
Never	0	0	1	1	0
Require modification?					
No	0	25	0	0	1
Yes, by vendor	2	1	2	4	1
Yes, by user	1	2	6	1	1
Number of users reporting	3	29	7	6	3
Comments	Increases COBOL program speed and efficiency; look for announcement of Optimizer II, possibly in 8/73	Users of this widely popular source library system were most enthusiastic about its security and control features; some also mentioned its interface to SCORE	A widely used payroll system, oriented toward bank service organizations	Users report this program is a necessity for 3330 disk operation under DOS, and a help with teleprocessing; it can also reduce a program's execution time; some report it has bugs and frequently aborts	POWER is IBM's spooler for DOS; there is no direct charge for POWER, but its users pay a premium in terms of main storage and machine time requirements; DOS/VS POWER runs in real memory only
Report number	70E-124-01	70-677-01	70E-595-02	See 70C-491-04	See 70C-491-04

User Ratings of Proprietary Software

PACKAGE NAME	QUIKJOB	RPG II (Report Program Generator II)	SCORE	S/M-1 (Sort/Merge-1)	SPOOLER
Vendor	System Support Software, Inc.	IBM	Programming Methods, Inc. (GTE subsidiary)	IBM	Boothe Computer Corporation
Average memory used	25K	21K	56K	80K	5K
Overall satisfaction					
Excellent	2	3	2	9	2
Good	1	1	2	11	1
Fair	0	0	0	1	0
Poor	0	0	0	0	0
Grade point average	3.7	3.8	3.5	3.4	3.7
Throughput/efficiency					
Excellent	1	2	1	7	2
Good	2	2	2	11	1
Fair	0	0	1	2	0
Poor	0	0	0	0	0
Grade point average	3.3	3.5	3.0	3.3	3.7
Ease of installation					
Excellent	2	2	2	10	2
Good	1	1	1	7	1
Fair	0	0	1	3	0
Poor	0	1	0	1	0
Grade point average	3.7	3.0	3.3	3.2	3.7
Documentation					
Excellent	1	2	2	4	2
Good	2	1	2	13	1
Fair	0	0	0	4	0
Poor	0	1	0	0	0
Grade point average	3.3	3.0	3.5	3.0	3.7
Vendor technical support					
Excellent	1	1	2	4	3
Good	1	2	1	10	0
Fair	0	0	1	6	0
Poor	0	1	0	1	0
Grade point average	3.5	2.8	3.3	2.8	4.0
Perform as advertised?					
Immediately	3	3	2	16	2
Eventually	0	1	2	5	1
Never	0	0	0	0	0
Require modification?					
No	2	3	3	15	1
Yes, by vendor	1	1	1	4	1
Yes, by user	0	0	0	1	1
Number of users reporting	3	4	4	21	3
Comments	A "quickie" report generator for $30/month that users like except for the comment that it can use only 1 input file; it has COBOL procedure division capabilities for user control; Quikjob II (7/73) can handle 2 input files and costs $50/month	RPG users seem to strongly favor the language even though IBM's support isn't strong for it; one user rated the RPG II Auto Report Feature as requiring 14K and being excellent overall, but with poor vendor support	Multi-purpose COBOL generator; SCORE enables non-programmers to produce file manipulation programs quickly; users praise this capability; one uses SCORE on a Honeywell 200 system	Sort/Merge was once a free IBM program; now users require this program product for 3330 disk support	This DOS spooling package increases system throughput, users claim, and uses very little core in the process
Report number	70E-798-01	See 70C-491-04	70E-457-02	See 70C-491-04	70E-100-01

User Ratings of Proprietary Software

PACKAGE NAME	SPRINT	SYNCSORT	TASK/MASTER	TOTAL	VANDL-1 (Vancouver Data Language-1)
Vendor	Jason Data Services, Inc.	Whitlow Computer Systems, Inc.	Turnkey Systems, Inc.	Cincom Systems, Inc.	IBM
Average memory used	11K	59K	93K	15K (plus buffers)	70K
Overall satisfaction					
Excellent	1	2	0	9	0
Good	4	1	3	3	1
Fair	0	0	0	1	0
Poor	0	0	0	0	0
Grade point average	3.2	3.7	3.0	3.6	3.0
Throughput/efficiency					
Excellent	2	2	0	5	0
Good	2	1	3	4	1
Fair	0	0	0	4	0
Poor	0	0	0	0	0
Grade point average	3.5	3.7	3.0	2.8	3.0
Ease of installation					
Excellent	3	1	1	9	0
Good	2	1	1	4	1
Fair	0	0	0	0	0
Poor	0	1	1	0	0
Grade point average	3.6	2.7	2.7	3.7	3.0
Documentation					
Excellent	0	1	1	0	0
Good	2	2	1	7	0
Fair	2	0	1	3	1
Poor	1	0	0	2	0
Grade point average	2.2	3.3	3.0	2.4	2.0
Vendor technical support					
Excellent	1	3	1	7	0
Good	3	0	2	4	1
Fair	1	0	0	2	0
Poor	0	0	0	0	0
Grade point average	3.0	4.0	3.3	3.4	3.0
Perform as advertised?					
Immediately	5	2	0	11	1
Eventually	0	1	3	2	0
Never	0	0	0	0	0
Require modification?					
No	4	1	0	11	1
Yes, by vendor	0	2	3	2	0
Yes, by user	1	0	2	0	0
Number of users reporting	5	3	3	13	3 (rated by only 1)
Comments	A low-cost spooling package for DOS systems that one user says redcues set-up time to near zero; but another complains it won't drive IBM's 1000-cpm reader at full speed; since the initial Datapro report, SPRINT has been made self-relocating	SyncSort is a disk sort package for large OS systems that comes with a money-back guarantee if it is ever outperformed	Users rate this data communications monitor as well suited for teleprocessing applications; one says it's better than CICS; another tried and rejected an in-house system as unreliable	A data base system that rivals IBM's IMS in total sales, TOTAL is praised for its low core usage and simplicity; users complain about its lack of hierarchical file structure and inability to handle sequential files; Honeywell is now marketing TOTAL for its Series 200/2000 systems	VANDL-1 is an IBM RPQ package acting as a stopgap until DL/1 DOS/VS is delivered; it is thus IBM's current data base system for DOS users
Report number	70E-549-01	70E-971-01	70E-866-01	70E-132-01	See 70E-491-01

Adpac-II Programming System
Adpac Computing Languages Corporation

MANAGEMENT SUMMARY

The Adpac-II Programming System is advertised as a revolutionary programming system for commercial programming applications. The system consists of an Adpac language compiler, an Adpac-to-COBOL translator (Polypac), a source library management system (LIBRA), a documentation subsystem (UNCODE), and a text editor (EDITOR). Among the claims made for the Adpac language and other system modules, as compared to COBOL, are these:

- Easier to learn.
- Faster to program.
- Faster compilation and execution.
- Furnishes better documentation.
- Enables better control of programming tasks.

Based upon the results of a Datapro survey, these claims are well justified. Users confirm that the Adpac Language is very easy to use and does compile faster, although improved control of programming depends heavily upon the skill of user personnel.

The Adpac Language is structured into four divisions: I/O, Data, Process Control, and Instruction. This language structure is reminiscent of that found in COBOL, and the I/O and Data Divisions are very similar to comparable COBOL structures.

The Process Control Division identifies the types of processing and the conditions under which this processing is to be done. For example, statements in this division can identify the conditions for accepting or skipping a particular master record, for matching master and transaction records, for identifying different types of transactions, for identifying non-regular records in a detail file (such as batch total cards), and for identifying control breaks for outputting totals.

The Instruction Division contains six sections that specify the processing for subroutines, any special system control (such as checking non-standard labels or interfacing non-standard I/O routines), any program opening details, update processing, detail processing, and control break processing.

Adpac's coding form is fixed-format and compact. In all divisions except the Instruction Division, each statement specifies one item, with room for comments to identify the operation, file, data name, etc. In the Instruction Division, two instructions plus the output of a record to a file can be specified.

Adpac is a complete business-oriented language and compiler that has gained a respectable number of users. The language features conciseness of coding and a highly structured format. An Adpac-to-COBOL translator allows users to convert Adpac programs to ANS COBOL at any time.

CHARACTERISTICS

SUPPLIER: Adpac Computing Languages Corporation, 101 Howard Street, San Francisco, California 94105. Telephone (415) 981-2710.

BASIC FUNCTION: Programming and documentation of business data processing problems. The Adpac compiler supports the highly structured Adpac language.

LANGUAGE: Instructions are grouped according to function in four divisions: I/O, Data, Process Control, and Instruction. The I/O Division identifies the files to be used, but not their structure. The Data Division specifies headings for reports, constants, lists or dimensioned arrays, working storage areas, and dictionaries (file structures). The Process Control Division specifies the conditions for selecting master records to be processed, and identifies updating, detail processing, or summary totals to be taken. The Instruction Division provides the detail coding for subroutines, label checking, program initialization, updating, detail processing, and totaling. Not all divisions need be present—and, in fact, seldom are for simple master-file update problems.

Adpac is a curious blend of program generator (RPG), high-level processing language (FORTRAN or COBOL), and macro-assembler. Much of the simplicity of the language is derived by positional implication; i.e., the location of character strings within the statement determines the function.

The Process Control Division is the key to Adpac. Here the structure of the program and the use of the data is defined.

The Adpac compiler generates the complete file-handling coding from specifications that are, in effect, parameters for subroutines carried in the Adpac compiler. Linkages to standard IBM I/O routines are automatically included, although user-coded routines can be accommodated if desired.

Within the Instruction Division, a group of 100 instructions provides the programmer with the ability to specify detail processing. The instructions have the feel of an assembly language generously interspersed with simple macros. Operands can be data names or literals, and can be specified by their relative addresses within a record.

The coding form is highly structured, with many processing definitions implied by a single character in a specific location. In general, one statement (card) represents one step in the processing. Statements within the Instruction Division can group two instructions within the same card. Any number of cards containing remarks only can be interspersed throughout the program.

Adpac-II Programming System
Adpac Computing Languages Corporation

▷ Adpac possesses sufficient flexibility to be used as a general-purpose programming language for commercial processing tasks. Because of the ease of coding file operations, it can also be used primarily for report generation or file maintenance. In this capacity, it is more flexible than the conventional report generator/file maintenance systems on the market. This flexibility, however, also means that Adpac does not include a set of default conditions; all items must be coded. In a sense, the relationship between COBOL and Adpac can be summed up by saying that COBOL is more flexible, but also more difficult to code.

Of particular note is a widespread concern of users about being locked in to a non-standard, non-compatible language. To overcome this concern, Adpac Corporation has introduced Polypac, a meta-translator that is used to translate programs into ANS COBOL programs. Polypac is included in the standard Adpac-II Programming System.

Adpac Corporation heralds the Adpac-II Programming System as the coming commercial programming system. Although the industry has been slow to agree, the role of systems that simplify applications programming with higher-level languages (whether of the COBOL preprocessor type or the more advanced Adpac concept) has been firmly established. The potential benefits of programming systems such as Adpac-II cannot be ignored by responsible commercial programming shops any longer.□

➤ **Algebraic expressions can be coded in free form, with subscripting to two levels. Extensive move and edit instructions facilitate manipulation of data. Complex processing can be programmed through the nesting of true/false comparison instructions, conditional execution of free-form algebraic expressions, and the capability to make any instruction conditional through the inclusion of modifiers in the statement.**

Subroutines are an integral part of the Adpac concept of programming. The neatness with which the program can be developed through extensive use of subroutines is very conducive to their use. All subroutines are closed; i.e., a return to the instruction following the one that caused the branch to the subroutine is enforced. Nesting of subroutines is permitted, but can get tricky. Subroutines can be generalized and modified at execution time through the use of parameter lists. The use of subroutines also tends to give a more tidy program listing that promotes readability.

Adpac's compiler is held in the systems library and is called in much the same way as the COBOL or FORTRAN compiler would be called. The normal mode of operation is "load and go"; i.e., an attempt to execute the program is made immediately following compilation. If desired, a copy of the object program can be produced and cataloged in a library.

HARDWARE/SOFTWARE REQUIREMENTS: Versions of the Adpac-II Programming System are available for an IBM System/360 or 370, Model 25 or larger, operating under DOS, OS (or their VS counterparts), or CMS.

PRICING: Adpac is offered in several versions under a complex permanent-license pricing structure that includes the number of programmers in the user installation, payout period, and condition of use (service bureau, educational, etc.). Typically, rates for shops with more than a dozen programmers are about 2/3 higher than those for shops with one to three programmers, based upon equivalent payout periods. A smaller Custom Adpac-II and a still smaller Quicpac, both with certain functions removed, are also available. Either subset provides less power than full Adpac, and the Quicpac System is intended to compete with IBM's RPG-II. Substantial discounts are offered for educational institutions and for multiple copies of the system. Additional programming, training, or special modifications are available for separately negotiated charges.

	Monthly Rental*	
	DOS	OS
Full Adpac-II	$750	$880
Custom Adpac-II**	450	615
Quicpac**	90	90

* Prices shown are for installations with more than a dozen programmers; smaller installations pay substantially less.

**Standard system without options; optional features are individually priced.

INITIAL DELIVERY: 1965.

CURRENT USERS: Approximately 100. ■

General Ledger System
Ancom Systems

MANAGEMENT SUMMARY

Ancom Systems is a company engaged in service bureau operations, facilities management, and the production and marketing of proprietary software. The company's software efforts are concentrated in the area of financial systems. With the current packages and the ones due to be announced shortly, a comprehensive line of financial packages will be available from one source, a rarity in the software market.

The General Ledger System is one of the oldest of the Ancom systems. Notably, the Ancom products form an integrated set of facilities. Compatibility among the systems is the rule. In fact, the same file maintenance programs are used for files in all the systems. Output from the Accounts Payable System, for example, can be input directly into the General Ledger System, bypassing the normal edit cycle. Naturally, some planning and foresight are required for the results to make sense. Account numbers and organizational codes should agree, for example.

The Ancom General Ledger System comes close to providing the flexibility formerly associated with manual ledgers. (Remember the good old days?) It is designed to accommodate multiple companies with reporting at the company level and two subsidiary levels. In addition, up to 99 levels of intercompany organization can be established for reporting purposes. (That would be some conglomerate!) An excellent series of conventional reports is provided, with extensive provisions for comparative reporting. In addition, analysis reports of various types provide all levels of management with operating performance information.

Many of the reports overlap and interlock, giving a choice of information presentation. For example, detail can be included in the profit and loss statement, or separate expense analysis reports can be generated. You can have your ledger listing complete, showing all months, each time you run, or you can get it in balance-forward form and save last month's report.

A potentially very useful feature is the provision for separate budget and forecast entries. You can hold one constant for reference and modify the other to show the best estimate (guess?) of performance for the remainder of the year. Comparative information at three levels (if you toss in last year's performance) requires some rather dexterous mental gymnastics to maintain proper perspective and to constructively relate to the wealth of information available for determining trends, trouble spots, etc. The situation can be further complicated by the options for changing comparative bases from run to run. The flexibility available is to be lauded, but it is not automatic; and careful planning and stern discipline are ▷

Designed for extensive comparative reporting and compatibility with other Ancom packages, the General Ledger System is provided in COBOL source-code form and has been implemented on IBM, Honeywell, and Burroughs computers.

CHARACTERISTICS

SUPPLIER: Ancom Systems, 1250 Sixth Avenue, San Diego, California 92101.

BASIC FUNCTION: To create and maintain current and historical files of general ledgers by account for reporting the financial picture of a company and the organizational operating results. Emphasis is placed on simplicity of data input, compatibility of file structure with other Ancom proprietary accounting packages, and comparative reporting at all levels of company and intercompany organization.

OPERATION: The General Ledger System is composed of two distinct phases: (1) data input and validation, and (2) master file update and report generation. The system is designed to accommodate information from multiple companies. Within the company, division and department organization can be identified and used for the basis of reports. In addition, up to 99 levels of organizational structure above the company level can be identified for reporting purposes. The standard reports provided with the system are heavily oriented toward showing comparative information (actual, budget, forecast, and prior years). Amount of detail included in the reports is highly flexible. In addition to the conventional reports, a number of reports deal with displaying organizational expenses. Excluding the input edit listings and registers, a total of 10 different types of reports are produced.

All programs are written entirely in COBOL. The system is designed for operation under DOS or OS on an IBM System/360, but has also been installed on several other computer systems.

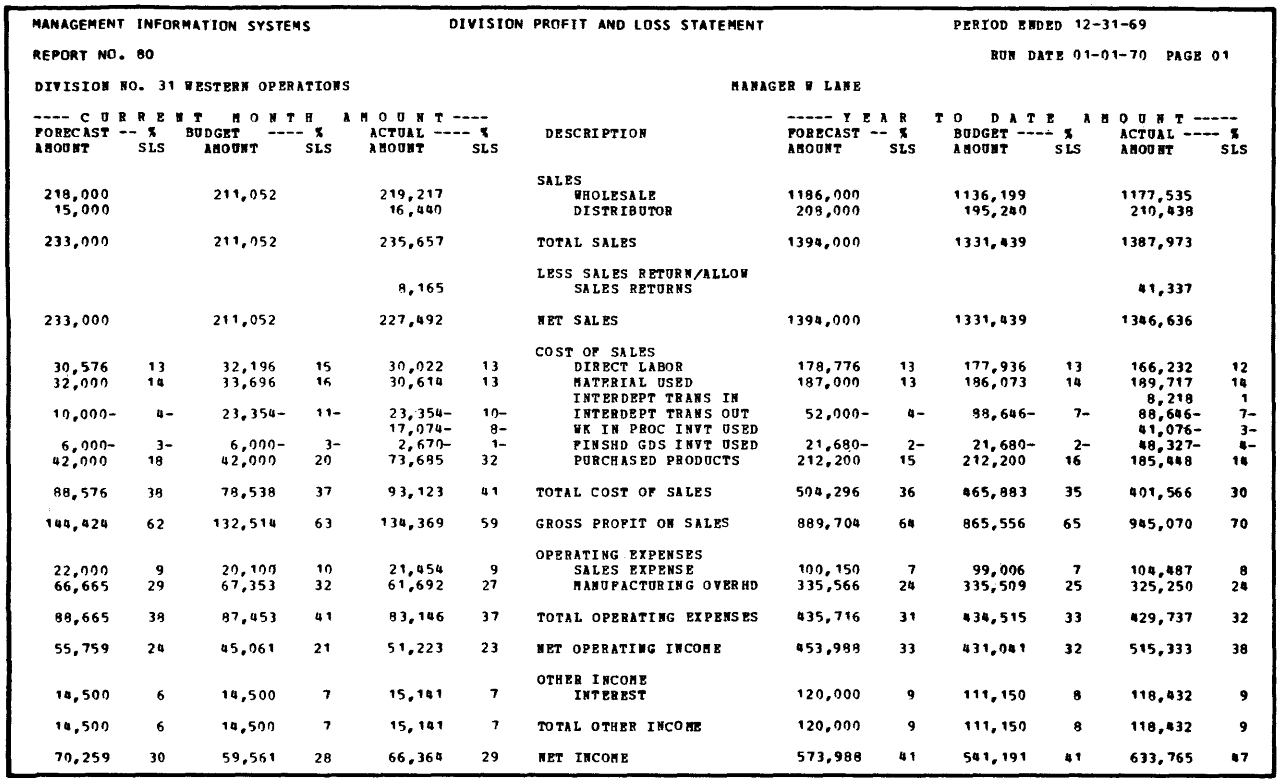
MASTER FILES: There are only two master files maintained by the system. The General Ledger master file contains individual ledger transactions as well as all budget, forecast, or other monetary items pertaining to individual accounts. The Standard Format master file is actually a set of tables containing the chart of accounts, organizational codes and relationships, budgetary calculation factors, report parameters, and descriptive identifiers for the account groups reported in various reports such as the balance sheet and profit and loss statement.

The General Ledger master file is organized sequentially according to the following order (major to minor): company, account, ledger type, month, division, and department. All data in the system is maintained in an active status. Each detail record holds 120 bytes of data, including the keys just mentioned and all data from the input journal voucher.

A total of nine different ledger types can be processed. Normally, only four are employed: actual or current transactions, budget amounts, forecast amounts, and prior year amounts. Each type is keyed to the company, division, department, account, and month.

Maintenance of separate budget and forecast information for all accounts is discussed under Processing; essentially, it permits recasting projections while retaining the original forecast for comparison purposes. ▶

General Ledger System
Ancom Systems



```
MANAGEMENT INFORMATION SYSTEMS          DIVISION PROFIT AND LOSS STATEMENT                          PERIOD ENDED 12-31-69

REPORT NO. 80                                                                                         RUN DATE 01-01-70  PAGE 01

DIVISION NO. 31 WESTERN OPERATIONS                                                  MANAGER W LANE

---- C U R R E N T   M O N T H   A M O U N T ----                                   ----- Y E A R   T O   D A T E   A M O U N T -----
FORECAST -- %   BUDGET   ---- %   ACTUAL ---- %     DESCRIPTION                  FORECAST -- %    BUDGET ---- %    ACTUAL ---- %
AMOUNT    SLS   AMOUNT     SLS   AMOUNT     SLS                                  AMOUNT     SLS   AMOUNT     SLS   AMOUNT     SLS

                                                    SALES
 218,000          211,052          219,217             WHOLESALE                 1186,000         1136,199         1177,535
  15,000                            16,440             DISTRIBUTOR                208,000          195,240          210,438

 233,000          211,052          235,657          TOTAL SALES                  1394,000         1331,439         1387,973

                                                    LESS SALES RETURN/ALLOW
                                     8,165             SALES RETURNS                                                 41,337

 233,000          211,052          227,492          NET SALES                    1394,000         1331,439         1346,636

                                                    COST OF SALES
  30,576   13      32,196   15      30,022   13        DIRECT LABOR               178,776   13     177,936   13     166,232   12
  32,000   14      33,696   16      30,614   13        MATERIAL USED              187,000   13     186,073   14     189,717   14
                                                       INTERDEPT TRANS IN                                              8,218    1
  10,000-   4-     23,354-  11-     23,354-  10-       INTERDEPT TRANS OUT         52,000-   4-     98,646-   7-      88,646-   7-
                                    17,074-   8-       WK IN PROC INVT USED                                           41,076-   3-
   6,000-   3-      6,000-   3-      2,670-   1-       FINSHD GDS INVT USED        21,680-   2-     21,680-   2-      48,327-   4-
  42,000   18      42,000   20      73,685   32        PURCHASED PRODUCTS         212,200   15     212,200   16     185,448   14

  88,576   38      78,538   37      93,123   41     TOTAL COST OF SALES           504,296   36     465,883   35     401,566   30

 144,424   62     132,514   63     134,369   59     GROSS PROFIT ON SALES         889,704   64     865,556   65     945,070   70

                                                    OPERATING EXPENSES
  22,000    9      20,100   10      21,454    9        SALES EXPENSE              100,150    7      99,006    7     104,487    8
  66,665   29      67,353   32      61,692   27        MANUFACTURING OVERHD       335,566   24     335,509   25     325,250   24

  88,665   38      87,453   41      83,146   37     TOTAL OPERATING EXPENSES      435,716   31     434,515   33     429,737   32

  55,759   24      45,061   21      51,223   23     NET OPERATING INCOME          453,988   33     431,041   32     515,333   38

                                                    OTHER INCOME
  14,500    6      14,500    7      15,141    7        INTEREST                   120,000    9     111,150    8     118,432    9

  14,500    6      14,500    7      15,141    7     TOTAL OTHER INCOME            120,000    9     111,150    8     118,432    9

  70,259   30      59,561   28      66,364   29     NET INCOME                    573,988   41     541,191   41     633,765   47
```

This profit and loss statement, one of 10 different types of reports produced by the Ancom General Ledger System, illustrates the system's extensive facilities for presenting comparative information.

required to keep management from running amok and creating heavy strains on the computer system in producing the multitude of reports this system is capable of and on themselves in trying to digest too rich a diet of information.

Even though the standard reports are excellent, Ancom is preparing a generalized report generator for delivery in mid-January 1972. The primary purpose of this generator is to allow initial tailoring of reports to an installation's specifications, but it can also be used for "one-shot" report preparation.

Without such a generalized report generator some relationships among the data recorded in the master file are "hidden." Two comments are applicable. Intelligent planning of the chart of accounts will go a long way toward providing much of the "missing" dimension within the ledger listing and account analysis. Secondly, Ancom is aiming towards an integrated group of financial software systems, and much of what some might call missing is supplied within the framework of the other packages. (The term "data base processing" rears its head, and the commonality of the master files in the various Ancom packages surely points this way.)

The flexibility gained in the Ancom system is not free. To get it, you must give up processing speed, in terms of

Alternatively, a balance-forward organization can be elected.

The Standard Format master file is maintained on a disk file in indexed sequential organization.

PROCESSING: Any chart of accounts can be accommodated, and different companies can use different sets with no restrictions. Changes to the chart of accounts are not difficult to make.

Provision of two types of budgets is an intriguing and very useful concept if handled properly. The normal way is to prepare a 12-month forecast at the beginning of a fiscal year. A budget is then prepared based on the forecast and other factors. A number of aids are available to assist in calculating expense elements of the budget, including units times a factor, percent of sales (for months subsequent to beginning of the year), and percent of forecast. It is difficult to imagine why budget and forecast should differ at the beginning of a year, but the strength of the concept lies in the ease with which the budget can be updated to reflect unforeseen changes in operations while still retaining the original forecast for comparison purposes and to serve as a constant base from which to make the new budget estimations. If changing the forecast while keeping the budget constant appeals to your sense of nomenclature better, it can be done that way because of the flexibility in specifying comparative elements in the various reports. However, calculation of the new forecast would be entirely a manual operation. If you want to modify both, you are on your own as far as interpreting the results. A detailed listing of the way in which each budget element was arrived at is output—a nice touch that is often overlooked.

The system is designed for monthly reporting of information accumulated in batches during the month. Several features promote speedy closing and reporting of monthly computer time, and restrain your impulses for one-shot reports. Normally, the master file retains all transaction records for each month of the year, along with monthly entries by account for the budget, the forecast, and the prior year (plus up to five other categories if you so desire). The file is in sequential form. To run the system, you pass the whole file, as is the nature of sequential processing. If you confine your reports to the regular monthly run, the situation is tolerable, but one-shots can be deadly. However, the monthly reports obtainable will satisfy most needs. Alternatively, you can set up the master file in a balance-forward mode, which will reduce running time but rule out some of the possibilities for reporting.

A brief DATAPRO 70 survey of users showed a general satisfaction with the system and with Ancom's support. □

results at the expense of slight inaccuracies, which are picked up as a matter of course and reflected in subsequent months. All information is available at any time. Reports can include detail data from previous months if desired. Input transactions that did not pass the editing and validation checks are entered into an addressable error suspense account. If a batch is out of balance, a balancing entry is placed against this account. This account is reported in the same way as other accounts, so a complete audit trail is provided. If handled properly, a great deal of information about the sources of errors can be gleaned. Accrual accounts (temporary accounts for which exact information is not known at closing time) can be identified for automatic reversal during the next month. Both sides of the entry are reversed. Thus, a great deal of flexibility is provided for getting reports to management quickly, without losing control. However, undisciplined use of these facilities can render the monthly reports virtually useless. Because the system contains no checks for the items mentioned, it is up to the data processing manager to maintain integrity of operations.

Reports can be produced at any time, but because the whole file has to be passed for each run, good practice would be to produce all desired reports during the monthly run. Any combination of the basic reports can be printed for each company represented during one run. But only one format for each report for each company can be produced in one run.

INPUT: The basic form of input is the journal voucher. The standard forms provided permit up to 23 items to be entered per form. Each line is numbered, as well as each page; several pages can be grouped under one voucher number. In addition, two 4-character references can be cited for each item, and a source code can be associated with each page. All of this information is carried in the general ledger transaction record, permitting complete tracing of information.

A 20-character description of each item can be input. All account numbers are assigned manually.

The data input phase consists of the edit/validation check of required fields, data types, and real account, company, department, and other codes, followed by a voucher register. An edit listing and batch balance summary are output from the edit cycle. Multiple batches can be combined for the voucher listing. Data in appropriate format can be entered at any point in the input phase; i.e., data can be processed fully, data can skip the edit cycle but appear in the voucher register, or data can skip the whole process and enter the system directly. (The output from other Ancom accounting packages is in appropriate form for direct entry.) One or more batches are then sorted for entry to the system. Automatic entry of recurring entries, such as rent and depreciation, can be set up.

REPORTS: The reports furnished by the system are particularly comprehensive and approach the flexibility of manual accounting practice.

The trail balance produced is an actual working paper with space for adjusting entries.

A listing of accounts can be produced in balance-forward form or with detail for all months shown. The user receives ledgers to work with.

The balance sheet and profit and loss statements are particularly flexible in permitting the selection of comparative bases. Detail or summary forms can be used, with individual control over which items (gross sales, net income, current assets, etc.) are summarized or detailed.

A total of six different types of analysis reports can be produced. The account analysis shows monthly balances by account number with control totals for accounts and subaccounts. (The eight-digit account number is divided into a four-digit control account and a four-digit sub-account.) Three expense analysis reports are organized by organizational group (company combinations, company, division, or department), by month with account, and by department within account. In general, the expense analysis reports show actual, budget, and forecast amounts for the current month; organizational reports also include year-to-date information for all three.

The remaining two analysis reports are more specialized. The current outlook report combines actual year-to-date amounts with the amounts forecast for the rest of the year to produce a profit and loss statement projection. The performance analysis report is a listing of a particular level of organization (division, department, etc.) showing net sales, cost of sales, operating expense, other income/expense, and net income. Variance in actual amount and percent is shown. The order of listing (ranking) can be based on any data column in the report.

Thus, the reporting facilities are very extensive. Report generation is selective, and only those reports desired need be produced. The output from a typical run can easily produce several hundred pages of reports, but remember that these are the working documents for management for the next month.

HARDWARE/SOFTWARE REQUIREMENTS: The Ancom General Ledger System requires a 24K-byte partition of main memory and three file units, at least one of which must be a disk. The system will run under OS or DOS on an IBM System/360. The system is supplied as a COBOL source deck. It has been successfully implemented on computers other than the IBM System/360, including the IBM System/370 Model 155, Burroughs B 2500 and B 3500, and Honeywell Series 200.

PRICING: The system can be purchased for $12,500 or leased for monthly charges of $1,250 (12 months), $650 (24 months), $450 (36 months), or $350 (48 months). One week of on-site installation assistance is included in the price.

FIRST DELIVERY: Spring 1969.

CURRENT USERS: About 20 as of January 1972. ■

ASI-ST
Applications Software Incorporated

MANAGEMENT SUMMARY

ASI-ST is billed as a complete data management system that can be used in conjunction with existing data base management or host operating system file structures. Furthermore, ASI-ST can replace most of the procedural-language programs used to access the data base, with attendant savings in program preparation time and improvements in processing efficiency.

Now, what does that make ASI-ST? And where can it fit into the scheme of your data processing organization, if anywhere? In order to fully understand what ASI-ST does, certain concepts must be recognized. For IBM operating systems, certain file access methods are provided, such as BSAM, ISAM, QSAM, etc. These methods are used to structure data files, and the resulting files are known as "conventional" files. While conventional file structures do store data on various physical media in ways calculated to improve data access, extensive room for improvement is possible. Most conventional file structures are rather inefficient, which results in burdens in processing time as well as excessive auxiliary storage requirements.

A number of program products have been developed in recent years that aim at managing the physical storage of data more efficiently than the conventional methods. These systems tend to produce "data bases", in which data redundancy is minimized and data relationships are maximized. The primary objective of such data base managers (DBM's) is to structure, organize, and provide logical- and physical-level access to logical data bases on physical devices. Examples of such systems are IBM's Version 2 of the Information Management System (IMS-2) and Cincom's TOTAL (Report 70E-132-01).

In order to prepare data for input (editing, audit trails, recording, accumulation of summaries of update activity, etc.), modify the resulting data base (alter the definition, etc.), or retrieve information (report generation, data base queries, etc.), the user must develop application programs in a procedural or host language such as COBOL, PL/1, FORTRAN, or BAL. This typically involves a sizable effort for a trained programmer, who not only describes to the DBM what he wants, but also how to do it to a considerable degree.

This is where a data management system (DMS) steps in. One major role of such a system is to provide a flexible, high-level user language to process the data base, thus replacing much or all of the specially written COBOL, PL/1 or other procedural language programs required for updating, manipulating, or retrieving information from the data base. A full-scale DMS handles user functions from the raw input data through the final output reports. ▷

ASI-ST, a full-scale data management system for IBM 360/370 and UNIVAC Series 70 computers, features a high-level user language that simplifies the processing of files and/or data bases. The system can work with files in conventional formats (such as BSAM or ISAM) or it can supplement the IMS or TOTAL data base management system.

CHARACTERISTICS

SUPPLIER: Applications Software Incorporated, 21515 Hawthorne Blvd., Torrance, California 90503. Telephone (213) 542-4381.

BASIC FUNCTION: ASI-ST provides a high-level user language that permits flexible updating, retrieval, and/or report generation using existing data bases, including IBM's IMS and Cincom's TOTAL. Capability is also provided to define, create, modify, and produce reports from other data bases using the inherent file structures present in the IBM System/360 or 370 or UNIVAC Series 70 host operating system, such as ISAM, etc. As a data management system, ASI-ST is applied to the user's master file(s) or data base by replacing procedural "host language" programs in PL/1, COBOL, FORTRAN, or even BAL that are required to update, manipulate, or generate reports from the data base. ASI-ST does not supplant the data base manager or conventional file access method itself (ISAM, IMS, TOTAL, etc.), but rather is used in conjunction with it.

OPERATION: ASI-ST processes multiple user-defined tasks in parallel in a single pass of the master file(s) or a single run through the data base(s). The data base itself can consist of multiple independent or related files. ASI-ST coordinates multiple related files, while independent files are processed in parallel in a single job step. The standard mode of operation is batch processing or remote job entry, but on-line interfaces to a variety of popular data communications monitors are planned for future release to permit conversational operation.

modules:

- ASI-ST Executive
- Dictionary Processor
- File Creation/Maintenance and Retrieval Processor
- Report Generator

Applications involving the last three modules above can be processed in parallel in one job step—a capability not found in other data managment systems.

Optional facilities can be added to allow: user "own-code" linkage; RCA 301, 501, or 3301 emulation processing; automatic JCL generation for System/360 or 370 OS; and interfaces to IBM's IMS or Cincom's TOTAL data base managers. A somewhat restricted subset of ASI-ST—COMPACT ASI-ST—is also available. ➤

ASI-ST
Applications Software Incorporated

Typically, three overlapping families of systems are concerned with these functions in anything less than a full DMS: file maintenance systems (from input data through updated files or data bases), retrieval systems (which retrieve data from files or data bases for reports or reformatted files/data bases), and report generators (which retrieve data and produce user-specified reports from exising files or data bases).

As a full-scale DMS, ASI-ST is not new or even unique; it shares the spotlight with other widely used systems such as Mark IV from Informatics (Report 70E-500-01). In fact, at the functional level, there are relatively few significant differences between ASI-ST and Mark IV. (It is of interest to note that the developers of the Mark I, II, and III predecessors to Mark IV worked on ASI-ST.) But ASI-ST does go substantially further than many other data management systems.

A full range of features designed for user convenience and ease of use is built into ASI-ST. Among these features are:

- Interfaces to IBM's IMS and Cincom's TOTAL data base management systems.
- Well over 300 easy-to-understand diagnostic messages.
- Ability for the user to perform calculations at any point in the processing.
- Capability to process a total of 26 input and/or output files and/or data bases in parallel.
- Processing efficiency on a level consistently as good and usually better than that of COBOL for the same job.

Of the wide range of systems that purport to handle part or all of the data management requirements associated with creating, maintaining, and/or retrieving information from a data base, ASI-ST seems to have a well-balanced mixture of all the right elements to do the job as well or better than its competition. Users with existing files in conventional structures (ISAM, etc.) or with IMS or TOTAL data bases are well advised to give serious consideration to ASI-ST to simplify the job of processing their data at a high level.

Users of ASI-ST for IMS(2), TOTAL, and conventional file structure interfacing contacted by Datapro all reported that the system performs according to expectations and is very effective. One user indicated that ASI-ST has proven to be his most valuable purchase of outside software – and this was in a sophisticated environment where several other reputable program products were in use. In general, ASI-ST's users reported that the system has been particularly effective in allowing non-programmers to manipulate and process data where only special programs developed by programmers had previously been in use. □

The ASI-ST Executive controls and directs the operation of the other three ASI-ST segments of subprocessors. The primary function of the Executive is to provide an interface between the ASI-ST user and the resident supervisor of the host operating system. Input/output calls are directed to the host operating system file organization procedures (ISAM, etc.), or to the applicable data base management system (IMS or TOTAL). Other Executive functions include:

- Processing ASI-ST user statements.
- Fetching ASI-ST dictionaries.
- Cataloging and calling ASI-ST procedures.
- Providing housekeeping for global parameters or temporary work areas passed between the other ASI-ST segments.
- Dynamically allocating disc and main memory work areas.

The Dictionary Processor builds and maintains disc files that contain complete descriptions of the user's data plus a variety of other handy information such as values used as non-integer data for calculations and/or print purposes, column headings used in output reports, input editing parameters, etc. Equivalence names, hierarchical or network data relationships, comments, and the maintenance sequence of the file(s) are also stored in the ASI-ST dictionary. Whenever a user statement is encountered that pertains to dictionary definitions, the Dictionary Processor is invoked. Whenever the Dictionary is established or modified, a glossary is produced containing descriptive diagnostic messages that can be used for audit trail purposes, as well as a cross-reference listing by relative field position within hierarchical segment level. Because the Dictionary contains the only "hard" description of the data base or master file(s) for ASI-ST, cataloged ASI-ST user requests can be entirely independent of the actual definitions of the structure of the data base and its records and fields. Changes to the data base structure do not necessitate changes to the user's processing requests. For this reason, plus the fact that a great quantity of additional information is stored by the Dictionary Processor to make life easier for the user, this module and the resulting dictionary is properly called the heart of the ASI-ST system.

The File Maintenance and Retrieval Processor handles any files that can be defined by an ASI-ST dictionary. Update transactions may be in any format and on punched cards, magnetic tape, or a direct-access device.

For updating, two modes are provided:

- Transaction-driven mode for sequential processing of a stream of transaction records. (This mode is normally used for batch processing.)
- Group maintenance ("event-driven") mode that allows application of a set of qualification criteria against the data base for common updating of a large qualifying subset of the data base. (This mode is generally used to set initial values in newly established files or to perform wholesale modification of existing files with common data.)

For retrieval, the Retrieval Processor can produce information to be run through the Report Generator or sub-files in

ASI-ST
Applications Software Incorporated

a non-report mode for subsequent processing by special-purpose procedural language programs (COBOL, PL/1, FORTRAN, etc.) or separate ASI-ST runs. Extracted data for reports is conditionally sorted prior to handling by the Report Generator.

The Report Generator provides automatic preparation of any number of printed reports following a single run through the data base(s) or master file(s). All reports using the same printer forms are batched together. In a non-spooled environment, operator console messages are provided by ASI-ST to indicate required operator intervention for the printer. In a spooled environment, that function is handled by the spooler.

ASI-ST uses a series of highly formatted specification sheets to set up and control user request streams. Extensive default parameters are used for blanks, with full flexibility to override these defaults with specific inputs. A total of 21 different types of language specifications can be represented on 7 types of work-sheets to control one or more of the following logical groups of functions: dictionary operations, file maintenance, retrieval/report generator subfile or non-report outputs, record selection/data manipulation/calculations, procedural directives and data base interface, group cataloging, and comments.

During the processing of an ASI-ST run, the following operations are performed in the sequence indicated: dictionary processing and file maintenance/retrieval processing, sorting of retrieved data, and report generation.

For each ASI-ST run, all user requests are compiled and executed on a compile-and-go basis. This is true even for cataloged requests, which are merely stored lists of user parameters. Following each ASI-ST run, complete documentation can be produced that lists the requests processed, any diagnostics issued, validation of changes, updated dictionary glossary, etc.

Standard relational operators provided in ASI-ST are equal, greater than, less than, not equal, not greater than, and not less than; these can be combined with five logical operators to form nested Boolean logic: IF (implies "OR IF"), WHILE (implies "AND IF" or "AND WHILE"), OR, AND, and Comma (implies "OR" or "AND"). Any set of selection criteria can be connected with any of up to five logic levels to express complex processing logic for accessing the data base.

A total of 322 diagnostic messages are standard in ASI-ST, including those for the IMS and TOTAL interface, and each is written out in full whenever it is encountered.

PERFORMANCE: Each ASI-ST run involves a concurrent compilation of multiple jobs, followed by the execution of the code to process the applicable files and data bases. Even with this compilation (whose purpose is to provide the user with flexibility that can come only from full independence of the data and user parameters), users report that processing time for a job done with ASI-ST is as fast, and in many cases faster, than the direct execution time of a COBOL object program for the same computer system.

HARDWARE/SOFTWARE REQUIREMENTS: ASI-ST runs on IBM System/360 or 370 computers under DOS or OS (MFT or MVT), under HASP or ASP; and on Univac Series 70 computer under TDOS. Under OS or TDOS a 60K to 90K partition or region is required for batch operation, while under DOS a 42K to 60K partition is needed. With either the IMS or TOTAL interface, ASI-ST requires a larger main memory area – typically about 85K to 100K bytes. ASI-ST runs under Remote Job Entry or CRJE modes on IBM systems as well as in standard batch processing mode. The ASI-ST system library disc residence requirement is about 320K bytes. ASI-ST interfaces are planned to the following data communications monitors: CICS, IMS Teleprocessing, and Intercomm (Report 70E-694-01).

PRICING: The full-fledged ASI-ST System or a COMPACT ASI-ST are both available on a perpetual license basis. The COMPACT version excludes the Transaction and Group File Maintenance capability, floating-point processing, and the procedural directives feature that permits the ASI-ST user to express directly some of the commands found in procedural languages such as COBOL. Users of COMPACT ASI-ST can upgrade to the full system within the first year of usage by paying the price differential only. Multiple copies of the full-scale version cost 35% of the basic price for the second or third copy; 30% for the fourth or fifth copy; and 25% for the sixth and subsequent copies. Multiple copies of the COMPACT version cost 40% of the basic price for the second or third copy; 35% for the fourth or fifth copy; and 30% thereafter.

	Perpetual License	Annual Maint.*
Full-Scale ASI-ST (includes warranty, one week of on-site training, use on an unlimited number of CPU's at one site, and a one-year maintenance contract)	$33,000	$3,000
ASI-ST Options (for full-scale version):		
Own-Code linkage	6,000	0
RCA 301, 501, or 3301 compatibility	3,000	0
Automatic JCL generator (for IBM OS only)	4,000	0
Interface to Total	10,000	0
Interface to IMS	10,000	0
COMPACT ASI-ST (includes warranty, 3 days of on-site training, use on an unlimited number of CPU's at one site, and a one-year maintenance contract)	20,000	2,000
ASI-ST options (for COMPACT version):		
Own-Code Linkage	4,000	0
RCA 301, 501, or 3301 Compatibility	3,000	0
Automatic JCL generator (for IBM OS only)	4,000	0
Interface to TOTAL	12,000	0
Interface to IMS	12,000	0
Report Generator Subsystem of ASI-ST (does not include training)	10,000	0
Use of ASI-ST at each geographical RJE terminal site	1,000	0

*After the first year.

Installment purchase plans are available on a negotiated basis. A standard lease (with purchase option) plan is also available.

INITIAL DELIVERY: March 1969.

CURRENT USERS: 68 installations in 43 companies. ■

PI SORT
Applied Data Research, Inc.

MANAGEMENT SUMMARY

As of June 1973, PI SORT is the only available sort program designed solely for sorting fixed-length records on IBM System/360 and 370 DOS systems. Partly because of its tight conformance to its defined objectives, PI SORT can, in most cases, outperform the free IBM-provided Sort 483. Also, PI SORT supports System/370 peripheral devices whereas 483 does not. (It should be noted, however, that Sort 483 can support certain 370 devices when they are Sysgened as 360 devices.) Additionally, users of System/370 DOS (Release 27) cannot use 483 to support their 3330 disks. Instead, they must license IBM's Sort/Merge 1 (S/M 1), for a monthly fee of $40.

PI SORT reads the same control cards as Sort 483. As a result, its effectiveness versus that of 483 can be tested by running simple side-by-side comparisons and drawing a conclusion from the respective run times. PI SORT is also installed easily by means of a simple cataloging procedure. Moreover, using PI SORT involves no operational changes in an installation.

The PI in PI SORT derives from Programmatics Incorporated, which at one time in its history was a wholly owned subsidiary of Applied Data Research. David Ferguson, founder and one-time president of Programmatics, developed the program. It was once known as PI SORT 1 and was a replacement for the then-popular IBM DSORT 450. PI SORT 1 is no longer supported, but ADR is now, as always, fully supporting PI SORT 2, now simply called PI SORT. The first PI SORT was delivered in February 1969. It was continually enhanced and improved, and the current PI SORT was delivered in November 1970.

PI SORT has received past publicity due to its involvement in complex legal maneuvers. The questions and claims have long since been settled; but while they were active, during much of 1970, PI SORT was not directly marketed. Since then, though, ADR has marketed the program strongly, in competition with both Sort 483 and now S/M 1. Support for IBM 3330 disks, already present in S/M 1, will be released for PI SORT during the third quarter of 1973.

PI SORT already supports the following System/370 devices which Sort 483 does not: 3400 series magnetic tape units, 3505 card equipment, and the 3211 printer. It also supports devices Sysgened as 370 devices and is able to accept sort input from SYSIPT, both things that 483 won't do. Perhaps the only remaining question regarding PI SORT support is about its use under DOS/VS; this will be evaluated for feasibility by ADR ▷

PI SORT enables faster sorting of fixed-length records under IBM System/360 or 370 DOS. It also allows increased sorting capacity for particular user configurations. PI SORT requires a minimum of 20K bytes of memory.

CHARACTERISTICS

SUPPLIER: Applied Data Research, Inc., Route 206 Center, Princeton, New Jersey 08540. Telephone: (609) 921-8550.

BASIC FUNCTION: PI SORT is a stand-alone, self-relocatable, fixed-length record disk sort program for IBM System/360 or 370 DOS users. Its sorting technique reduces work-space requirements for large files and generally provides faster results than comparable IBM sorts.

OPERATION: PI SORT is installed using a simple cataloging procedure that does not affect an installation's standard operations. PI SORT reads the same control cards as IBM Sort 483, so retraining and reorientation are not necessary. It supports all System/360 devices that Sort 483 does (e.g., 2311, 2314, and 2319 disks, 2400 series tape drives, etc.), as well as some actual System/370 devices (3400 series tape drives, 3505 card equipment, and the 3211 Printer), which 483 will not do. PI SORT will also accept input from SYSIPT, another thing that 483 won't do. IBM 3330 disk support for PI SORT will be released in the third quarter of 1973. DOS/VS support will be evaluated for feasibility when comprehensive documentation of that operating system is released by IBM.

In situations that PI SORT can't handle (e.g., variable-length records, tape work files, merge-only, COBOL or PL/1 SORT verb processing, etc.), control is automatically passed over to Sort 483. PI sort has a replacement module for 483 device code checking, so that, if an exit to 483 is made, 3400 series tape drives can still be handled.

PI SORT gains speed from its fixed-length record optimization philosophy, which allows it to calculate record addresses rather than having to move and store records, as is often necessary when variable-length records are sorted.

PI SORT contains two sorting programs: a linear sort for use when work space is at a premium or when the file is sufficiently ordered to warrant single-pass sorting, and a non-linear sort for higher performance when adequate disk space is available. The non-linear sort is always chosen when space is available. The non-linear sort has the same disk space requirements as Sort 483: twice the size of the file being sorted.

If this amount of space can't be made available, the linear sort is used. It only requires disk space exactly equal to the size of the file being sorted. While its use may represent a sacrifice in performance, a user can sort his file even when space is limited. Moreover, if the size parameter is specified incorrectly, so as to exhaust work ▶

PI SORT
Applied Data Research, Inc.

▷ once IBM makes full DOS/VS documentation generally available.

Sorting is an essential operation in virtually all commercial computer installations. It has been estimated that up to 40 percent of all computer run time involves sorting. Most of this time is visible, but some applications often hide sorts within programs. When sorts are based upon fixed-length records and the sort is a phase between application programs, PI SORT can help users economize. Often these economies can be substantial, as indicated in the user experiences reported below.

PI SORT does not, however, come into use when the ANS COBOL SORT or PL/1 SORT verbs are used. In those cases, PI SORT calls the IBM sort. However, PI SORT calls the IBM sort. However, PI SORT contains a replacement module for 483 device code checking so that, should the user's program exit to Sort 483, IBM 3400 series tapes can be handled.

Overall, since economies can be achieved so simply in the area of sorting, PI SORT should receive careful consideration in comparison to both IBM's free Sort 483 and its separately priced S/M 1.

USER REACTION

PI SORT users are a satisfied group. An insurance company in Florida has been using PI SORT on its 65K and 131K Model 30's (soon to be upgraded to a Model 40) ever since it discovered in January 1973 that a particularly large sort job took only 40 minutes with PI SORT versus 1 hour and 40 minutes using Sort 483. In an Ohio manufacturing company, PI SORT has been in use since April 1971. Benchmarked on that firm's 208K Model 145, and using 54,000 records, times for SORT 483, S/M 1, and PI SORT were, respectively, 11 minutes and 6 seconds, 11 minutes and 51 seconds, and 7 minutes and 21 seconds. The disk drives used were IBM 2319's. The customer is anxiously awaiting 3330 and DOS/VS support in PI SORT. □

➤ space before the entire file is processed, PI SORT supports an automatic cutoff feature which enables the user to sort at least a portion of his file.

Significantly, PI SORT has the ability to make use of any file sequencing currently in effect. This is handled by the creation of strings during the first phase. Data is read into core, and when available core is full, the lowest item is read out. At this point a new item is read in, and, provided it has a later key than the item being read out, it is melded into the current string rather than being held back for the next one. This continues as long as possible.

A group of records whose keys are already in order, therefore, will be included in the same output string. For instance, when record 235 is followed by 236, 237, and so on up to 250, then even if the memory space only allows record 235 to be read in, as soon as it is read out record 236 will come in and then be read out, then 237, etc. This can greatly reduce the number of strings and, consequently, the sorting time.

PERFORMANCE: Various benchmarks and user experience have demonstrated PI SORT's ability to outperform Sort 483 or S/M 1 by an average of 20 to 40 percent in usual cases. In unusual situations even greater gains in performance (sometimes over 50 percent) have been reported.

It is not always possible to simply estimate PI SORT versus IBM sort performance by using IBM timing tables. The tables assume that no multiprogramming is involved and that data placement is totally random on the files. In actual practice, either condition may or may not be the case. Since actual testing is comparatively easy, it is advisable (from both the theoretical and the practical standpoint. In real-life use, multiprogramming is usually in progress and files are somewhat ordered.

HARDWARE/SOFTWARE REQUIREMENTS: PI SORT requires an IBM System/360 or 370 DOS system with at least 32K and one 2311 disk drive (2311 version) or 65K and one 2314 or 3330 disk drive (2314/2319 version). Partition requirements are a minimum of 20K for 2311's, 30K for 2314's or 2319's, and 48K for 3330's.

PRICING: PI SORT can be leased for $100 per month with a 30-day cancellation clause. Program maintenance is included in the lease charges for the duration of the contract. A free 30-day trial period permits thorough user evaluation.

INITIAL DELIVERY: The first PI SORT package was installed in February 1969. In its current version, PI SORT was first delivered in November 1970.

CURRENT USERS: 104 as of March 1973. ■

The LIBRARIAN
Applied Data Research, Inc.

MANAGEMENT SUMMARY

The LIBRARIAN is a source-program maintenance and retrieval system for IBM System/360 and 370 DOS, OS, and VS users. In addition, it automatically produces audit-trail documentation of individual programs, as well as the entire source library containing the programs. Once the system is installed, programmers use LIBRARIAN control cards to access source programs for updates or compilations.

The LIBRARIAN eliminates cumbersome card decks, provides increased security for program maintenance and storage, and reduces programmer time used for modifications to programs—all usual functions of library software. The LIBRARIAN also provides:

- Improved and/or cheaper audit-trail documentation.
- Reduced disk storage space requirements.

The second advantage is especially significant because The LIBRARIAN can be used to store data, not just programs.

The LIBRARIAN approaches the audit-trail documentation problem by providing several types of standard documentation for each program and library of programs. These are produced automatically as a by-product of typical updates and program modifications, rather than as a separate effort, however slight, by the programmer. The audit trails provide a historical record of individual programs, reflecting the successes or failures of all LIBRARIAN operations as well as providing helpful data for the programmers to assist them in using the system.

Reduction of programmer effort required for modification is a significant justification for the use of The LIBRARIAN. An important point to consider is that each individual programmer retains control over his own programs through the use of control cards, rather than instructions to the computer operator. Once created, the deck can be run with other programmers' update decks in a single batch. The elimination of a series of individual runs will save considerable computer time.

Besides the obvious aid The LIBRARIAN provides in program development, program maintenance is simplified by the audit-trail provision. History cards are listed along with the program listing showing reasons for specific program changes and their corresponding dates. These are automatically placed as history comments next to affected program statements by The LIBRARIAN.

In attempting to determine possible savings from using The LIBRARIAN, an installation should estimate the frequency of source program updates and compilations. Typically, they will be run once or twice a day, but this figure may vary for different installations. ▷

The LIBRARIAN is a widely used software system that maintains programs on disk or tape rather than on cards. COBOL, FORTRAN, PL/1, and Assembly language modules can be intermixed in a single user library. Versions are available for System/360 and 370 DOS, OS, and VS users.

CHARACTERISTICS

SUPPLIER: Applied Data Research, Inc., Route 206 Center, Princeton, New Jersey. Telephone: (609) 921-8550.

BASIC FUNCTION: The LIBRARIAN maintains and controls source program libraries on tape and/or disk for installations using an IBM System/360 or 370 (DOS, OS, or VS). The LIBRARIAN documents all update runs and records the date each card of an existing program was last modified. It also includes the facility to interface with ADR's AUTOFLOW (Report 70E-052-03).

A feature of the program is that it will maintain a library containing any combination of programs (in any language) and/or data, so long as it was originally supplied in 80-column card image format.

OPERATION: The user of The LIBRARIAN system will instruct his programmers to use the 12 LIBRARIAN control cards in order to modify or retrieve any source program. Each programmer will produce his own card input to achieve his particular purposes—e.g., to modify or document a program, or to produce a workstream involving a compilation and perhaps a program execution. His card deck then can be batched with others and run against the library.

In order to modify a program already on in the tape or disk library, the programmer uses INSert, DELete, or REPlace control cards. The required parameters specify the sequence number of the statement after which the insert is to be made, or the sequence numbers of the first and last statements to be replaced or deleted. Optionally, updates can be made based on data card sequence numbers (when they are present) rather than using the explicit LIB commands.

Three commands permit character string operations, within individual card images. The LIBRARIAN can search a module or group of cards within a module for the occurrence of a particular string and replace that string with another (EDIT) or list the occurrences (SCAN). The FILL command causes a specified string to be forced into particular card columns (used primarily to insert or modify program ID fields).

In addition to actually modifying his program, the programmer can also document the modification as he desires. Documentation consists of either listing the program or punching it out. Under The LIBRARIAN system, the programmer must specify his output requirements only if they differ from the standard documentation produced for each run. For example, if a typical run produces a listing of any modified program but does not punch programs out unless a special request is submitted, a programmer will receive a listing automatically without using any control cards. He can additionally request a punch-out or suppress the listing simply by entering a SEL (select) control card. This card initiates the run and identifies the program module to be selected by The LIBRARIAN. ➤

The LIBRARIAN
Applied Data Reserach, Inc.

Elimination of source program card decks is another advantage for several reasons: reduced storage requirements, fewer accidents such as dropping or tearing cards, and reduced machine time otherwise wasted in reading cards. In some situations, however, the ability to change a single card in a particular deck directly is desirable or the machine time saved by eliminating decks is negligible.

Additionally, The LIBRARIAN can generate savings in file space by significantly compressing, by as much as an 80 into 25 ratio (80 into 50 is more typical), any card images, be they source programs or data. Besides this, The LIBRARIAN, using disk master files and its special direct-access techniques, can maintain disk file space effectively and can permanently eliminate one hated procedure, the file REORG.

The key elements in determining the potential value of reducing or eliminating the use of card libraries are the amount of time which can be saved and the need to reduce accidental mishaps in card handling. Again, this depends on local factors and individual circumstances. The important characteristic of the LIBRARIAN is that it offers an alternative to current practices, thereby providing the opportunity to investigate and improve current performance.

Sales and support for the program are handled from the Princeton headquarters of Applied Data Research. Other ADR offices are located in principal cities throughout the United States. Overseas offices and representatives of ADR are located in Canada, Japan, France, and Sweden.

USER REACTION

Datapro contacted LIBRARIAN users who are using both the DOS and OS versions of the system. Without exception, all initially acquired The LIBRARIAN because it eliminated cards. They continue to use the product for a variety of reasons: for project control (one user has eight groups with all their program modules on The LIBRARIAN), for source program manipulation with all source modules in one location, for the SCAN and EDIT features unavailable in IBM source library software, for its "efficiency and stability," and for its backup security. One user claims that its versatility is such that he has executed half-day schedules directly from The LIBRARIAN–a thing he'd never dream of attempting with IBM program libraries.

The users interviewed had LIBRARIAN experience ranging from six months to two and a half years, and were using such diverse systems as DOS 360/30's, 40's, and 50's, DOS 370/135's and 145's, OS 360/50's and 65's, and OS 370/135's, 145's, and 155's. Only one user reported a problem, and that was quickly solved. The problem was with initial use of 3330 disk drives with The LIBRARIAN. □

A Utility feature is included that gives the user much of the flexibility in rearranging card images that he had when working directly with the cards. Groups of cards can be moved around in the same module, transferred to a new master file, set up as a separate module, placed into card-image data sets on disk or tape, and punched and/or printed.

Optionally, users can employ the COBOL syntax checker to examine modules when added and during subsequent changes. The syntax checker all levels of IBM COBOL, including ANS, and checks for correct sentence structure, punctuation, reserved words, picture clauses, etc., but not for validity of data or for procedure references. The user can specify that updating and compilation for the particular module be suppressed if any syntax errors are found. By flagging errors at update time, unsuccessful compilations can often be avoided.

The LIBRARIAN contains automatic safeguards against making changes to the wrong program. The main protective feature is the password facility. As each program is placed on the file, a unique random four-character "passwork" is assigned to it. A user can require that the proper password, as well as the program name, be specified before any updates will be applied.

The password facility is basically a protection against keypunch errors, rather than against any security breaches. ADR feels that the module names used by programmers are so alike that clerical errors frequently occur. The password technique will prevent a single keypunch error from causing serious damage to the wrong program module. Additional security is provided by a cryptically encoded software lock.

Another security feature that can be optionally used by the programmer is the VERS feature, which offers protection against using an outdated listing. The programmer supplies the date (or date and time) of the listing he's using and The LIBRARIAN will automatically compare this information to the date (and, if necessary, time) when the module was last updated. If the two do not agree, the programmer is assumed to be updating an outdated module, and updating for that module is bypassed.

A Debugging facility copies existing modules and assigns the copied module a new name. This temporary module can be used for testing and debugging purposes, while the original module simultaneously is maintained as a production version. Once the copied version has been thoroughly tested, it can be renamed to replace the original version. Alternatively, a temporary change can be made to the production module.

If a user has tape master files, Cycle Control is an optional facility which relieves programmers of the burden of cycling the tapes. A Cycle Control File is set up on disk storage for each source library. When The LIBRARIAN is called to do an update, all tape mounting instructions are printed out automatically by the program. The Cycle Control feature also checks to ensure that its instructions have been followed properly when tape master files are used. This prevents the operator from mounting the incorrect tape and applying updates to the wrong file. In addition, a specific number of tapes are allocated for each library concerned and assigned appropriate sequence numbers.

Default JCL parameters are stored for effortless standard operating procedures with both tape and disk master files. The LIBRARIAN uses these optional parameters to automatically generate error-free JCL at the user's direction.

In creating a job stream, the latest versions of the programs in the library are merged as instructed by the programmer's control cards and passed, together with additional control cards and data, into the operating system. Compile-only or compile/link/go operations can be initiated as necessary.

Card images are compressed when entered on The LIBRARIAN master file. Compression is achieved by eliminating contiguous blanks, by eliminating the program ID field of each card (optional), and by taking advantage of certain characteristics of each language. Maximum compression is typically achieved with higher-level languages, such as FORTRAN and COBOL, rather than with

The LIBRARIAN
Applied Data Research, Inc.

➤ Assembly language code. ADR estimates that, typically, one source card will occupy only about 25 bytes, including the control codes necessary for expansion to the full form.

Master file setup or adding to the master file requires preparing a card, tape, or SYSSLIS deck for each program to be placed on the master file (if none exists), determining the number of individual libraries to be established, and running an initialization pass. In addition, all programmers and operators are provided with introductory training in the use of the system. A standard for using and filing the program documentation produced during the runs also should be established.

REPORTS: The LIBRARIAN produces several levels of documentation, designed for immediate reference purposes as well as a history. A record of all actions taken on any of the programs is permanently kept on the file itself as standard history. Four versions of master file documentation are: an updated Master File Index (showing the details of all modules on file, when they were last altered, etc.), a Summary of Activity, a detailed Update Record, and, optionally, a listing of any particular module(s).

The Summary of Activity lists the name of each module selected for updating, listing, or any other purpose during a specific LIBRARIAN run. This listing notes whether the programmer's instructions were correct–i.e., whether the selection was successful or not–but does not go into any further detail. If an error occurs, no changes are made to the particular module involved. Operations on all other modules, if correctly specified, are performed. Functionally, this Summary can be used as a management report on program activity.

The Update Record, when used properly, can be an invaluable record of program history. It lists all instructions supplied by the programmer, all cards which have been inserted, deleted, or replaced, and any errors that were found, together with the reasons for them and the corrective action taken. It also contains comments from the programmer, such as justification for making certain updates or modifications.

This ability to include commentary on a specific change is particularly useful as timely supplementary documentation which facilitates an understanding of the program when reviewed at a later date. The LIBRARIAN provides two types of historical documentation via the history card and the comment card. Any comments included on a history card will be permanently stored on the master file. The explanations or descriptive information presented on a comment card will appear only on the Update Record produced for that particular run. Thus, a programmer who wishes to temporarily record information of a transitory nature may do so through the comment card without cluttering up the master file with unnecessary historical details.

The Master File Index Listing is another type of documentation produced for each LIBRARIAN run. It contains details of the date each module was last updated, thereby providing the audit trail mentioned above. It also lists–using one line per program module–the vital statistics of the program, including its name, date added to the library, number of cards involved, responsible programmer, the data set used, and the job name. Any print listings stored in the library are also itemized. The Index can be used for record purposes and also for posting on a convenient bulletin board for programmer reference to ensure that the material they are using is completely up-to-date.

The fourth type of output, the Module Listing, consists of a complete list of each card comprising an individual module, together with the date the card was last modified. Absence of a date indicates the card was part of the original module and has never been changed.

OPTIONAL FEATURE: The OS version of The LIBRARIAN has an optional feature called SpaceSaver, which is designed to produce management reports aimed at helping to achieve improved utilization of disk storage space. Due to the limited OS facilities for disk space utilization control, OS installations typically have substantial amounts of unused, inaccessible disk space. SpaceSaver can go further to help users make use of this space than the existing IBM utilities.

SpaceSaver can: report the amount of space allotted versus the amount actually used, call attention to "stray" data sets (i.e., data sets created but never used again), determine the number of directory blocks used, and report data set usage by user and volume. It can provide management with information that can help avoid a disk storage shortage crisis and help to optimize the use of disk storage resources. It can produce two disk analysis reports. It can also allow users to establish naming conventions for data sets, and its reports can be helpful even when conventions haven't been established in the past.

HARDWARE/SOFTWARE REQUIREMENTS: The LIBRARIAN operates under DOS, OS, or their virtual storage counterparts on any IBM System/360 or 370 except the Model 20 and Model 44.

Two tape drives are required to handle tape library files. A 2311 Disk, 2314 Disk, or 3330 Disk can be used for disk files. A tape-oriented LIBRARIAN installation needs a disk to accommodate the Cycle Control feature if desired.

Main memory requirements depend on the operating system, features implemented, and type of library master files used. Approximate requirements are: DOS tape–34K bytes; DOS disk–38K bytes; OS tape–40K bytes; OS disk–55K bytes. An additional 12K bytes are required if the COBOL Syntax Checker is used. By leaving out certain options, such as EDIT, SCAN, and COBOL Syntax Checker, a DOS user employing tape master files only can operate in just 24K bytes. The above main memory figures do not include operating system residency requirements.

PRICING: The LIBRARIAN can be obtained for permanent license under three payment plans, with annual maintenance paid separately after the first year, or on a monthly license basis. Prices are different for the DOS and OS versions.

The DOS version costs $3,960 in a single payment for permanent license, or the payments can be made at $1,518 per year for three years (total $4,554) or at $143 per month for 36 months (total $5,148). The monthly license cost for the DOS version is $220 per month. Annual maintenance after the first year is $495.

The OS version costs more, and also offers the separately priced SpaceSaver feature. Permanent license fee in a single payment is $4,800 for The LIBRARIAN and $990 for the SpaceSaver. Paid in three yearly installments, these figures are, respectively, $1,860 (total $5,580) and $380 (total $1,140). In 36 monthly payments, the respective costs are $175 (total $6,300) and $36 (total $1,296). Monthly license cost for The LIBRARIAN (OS) is $270, and for the SpaceSaver it is $55. Annual maintenance after the first year costs $650 for The LIBRARIAN and $150 for the SpaceSaver.

INITIAL DELIVERY: October 1969 (both DOS and OS versions).

CURRENT USERS: About 850 as of April 1973. ■

AUTOFLOW II
Applied Data Research, Inc.

MANAGEMENT SUMMARY

AUTOFLOW produces flowcharts automatically on a high-speed printer. Nearly everyone knows that. Nearly everyone also knows that AUTOFLOW is enormously successful and is one of the "grand-daddy" proprietary software packages. Well then, what's to say about AUTOFLOW today? The answer is a lot or a little, depending on how you use your computer today.

Behind the April 1973 announcement of ADR's AUTOFLOW II with optional Extended Text Compositor (ETC) and Cross Program Auditor (CPA) lies a major new marketing thrust for the package in the future. That is, AUTOFLOW II, with CPA and ETC, is being marketed as a total program development system, while the more than 1800 present AUTOFLOW users have, for the most part, become accustomed to regarding the package as a tool for gathering and tying up the hanging threads of prior poor or nonexistent program documentation. Since word-of-mouth in the EDP industry has further propagated this viewpoint, ADR is faced with a big reorientation task in its AUTOFLOW II marketing program.

Some users have already "discovered" AUTOFLOW's capabilities as a program development system on their own, and won't require reorientation. For the rest, ADR encourages this new outlook so that it can garner repeat AUTOFLOW business or (where the system has been purchased) at least convince the customer that the system has everyday uses (and thus, everyday payoffs in system efficiency).

Since no user has AUTOFLOW II yet, how can ADR's claims be judged? Positively, we think. Two facts support this judgment: (1) the happiest AUTOFLOW users are the ones who are presently enforcing its use in program design and documentation, and (2) ADR can be counted among the foremost in expertise in program development and maintenance administration.

In summary, we can say that although ADR clearly has a vested interest in promoting the use of AUTOFLOW (and LIBRARIAN, etc.) in program development and maintenance, it is nonetheless correct in stating that users with the best documentation are best off, and that AUTOFLOW II fills the bill. The only question remaining, then, is how do you measure the system's value in dollars and cents?

Perhaps you can't measure savings of this sort directly. Maybe it's more like buying a house or an insurance policy; the money spent can better be regarded as a sensible investment rather than as direct savings. In weighing AUTOFLOW's value, let's consider the steps in program development and maintenance (or program pre- and post-development) to which ADR claims the AUTOFLOW II system can be applied. They are:

- Design – consisting of specifications, review, modification, and documentation;

AUTOFLOW automatically generates flowcharts from source language and/or COBOL program notes on a variety of computers. AUTOFLOW II is a complete documentation system, consisting of the flowcharting facilities plus the optional Cross Program Auditor (CPA) and Extended Text Compositor (ETC), that can aid in the management of data-base-oriented systems and enhance textual documentation.

CHARACTERISTICS

SUPPLIER: Applied Data Research, Inc., Route 206 Center, Princeton, New Jersey 08540. Telephone (609) 921-8550.

BASIC FUNCTION: To form a complete program pre- and post-development system in the form of specification flowcharts, program logic and documentation flowcharts, cross-listings within and among programs and documentary text. (In actual use, however, AUTOFLOW's basic facilities are most commonly used to explain the logic of a given program for use as a conversion aid, to initialize documentation of a previously undocumented program, or as a "post-mortem" debugging aid.)

Versions of the AUTOFLOW system are available for many computer systems and many source languages. Principal support, however, is directed toward the IBM System/360 and 370, the UNIVAC Series 70 and 9000 family, and the Honeywell Series 200 and 2000 computers, and toward the COBOL, Assembler, FORTRAN, and PL/1 languages (in that approximate order of popularity) and the CHART language of AUTOFLOW.

In popular usage, COBOL, FORTRAN, Assembly, and PL/1 AUTOFLOW are used to produce flowcharts of the program input. A Compress option can cause the output flowcharts to eliminate much detailed logic and thus present a more general view of the program logic. Chart/AUTOFLOW allows creation of pre-implementation flowcharts. Compress is available with COBOL AUTOFLOW and Chart is available with COBOL, FORTRAN, Assembly, and PL/1 AUTOFLOW. FORTRAN/AUTOFLOW has an optional Compress-like "IGNORE" facility.

The basic chart set output of AUTOFLOW is a title sheet, cross-reference table of contents and references, and, when necessary, a table of diagnostics. Output optionally specified at AUTOFLOW run time includes an input listing, procedural statement label index, flowcharts, and special listings.

In all, System/360 AUTOFLOW can accept over 20 languages in addition to the standard System/360 and 370 languages and produce output on the System/360 and 370 devices, or, optionally, on a Stromberg-Carlson 4020 or CalComp plotter. Some of these languages are: IBM 1401 SPS, 1401/1440 Autocoder, 1410/7010 Autocoder, 7070 Autocoder, 7080 Autocoder, Control Data 3000 Series COMPASS assembly language (except for the CDC 3600 and 3800), Xerox 900 Series assembly language, Xerox Sigma Series assembly language, UNIVAC 1107 and 1108 SLEUTH assembly language, UNIVAC 418 ART assembly language, and Honeywell DDP-24 Series DAP assembly language.

In addition, special COBOL support exists for Burroughs B 3500, B 5500, and B 6500 and CDC 6000 Series

AUTOFLOW II
Applied Data Research, Inc.

- Implementation – consisting of coding, desk-checking, testing, and debugging;
- Verification – consisting of system integration, system testing, and operations documentation; and
- Maintenance – consisting of modifications, enhancements, conversions, and system redesign.

By the way, there is growing evidence that the post-production (maintenance) costs for a given program tend to equal or exceed the preproduction costs after a time. This is particularly true when a need arises for conversions (new machines, vendors, peripherals, or operating systems), major modifications (new requirements and/or objectives), or redesign (new objectives or administration). Since application programs tend to have a life in the range of ten years while the lifetimes for systems (and, sadly, sometimes vendors) are generally shorter, the possibility of these needs arising should not be taken lightly.

In any event, there is no question about the need for adequate documentation as a safeguard against changes in personnel and against that time in the future when even the originator of the program is hazy about its details. For this, AUTOFLOW has always provided much assistance, and the AUTOFLOW part of the current AUTOFLOW II system is accordingly discussed language by language.

COBOL

At first sight, it is not obvious that COBOL needs documentation. After all, COBOL programs are written in an English-like form that is fairly easy to read, has a standard way of describing data, etc., and so appears not to need much additional documentation.

The key point, however, is the fact that human beings tend to think in terms of pictures rather than sequences. The COBOL program itself is simply a sequence of instructions which, if a reader had absolute recall, would be equivalent to a full flowchart. In fact, programmers do not have memories nearly that good, so it can be very useful to them to have the computer create clear, pictorial representations of program logic.

Originally, there was only one form of COBOL/AUTOFLOW flowchart. This included the complete detail of the program as taken from the instructions, interspersed with such notes as the programmer included with the program. This is still the most popular form, but two additional capabilities allow the program to be shown with either more or less detail.

Where less detail is required, Compress/COBOL removes from the charts various types of operations–such as the actual processing. This version might be used to bring out the network of the program without jamming in all the details of the processing which does not change the connections of the network.

The other form, which comes in the same package as the Compress form, is the Chart form. This expands the

systems, and special FORTRAN support exists for Honeywell DDP 24 Series, Xerox 900 Series, CDC 3000 and 6000 Series, and UNIVAC 1107 and 1108 (FORTRAN IV and V) systems. AUTOFLOW accepts most versions of FORTRAN IV and many versions of DOD or ANS-compatible COBOL. Language upgrades can be assisted by, for instance, submitting a COBOL F program identified as an ANS COBOL program and acting upon the diagnostics produced by AUTOFLOW.

PROCESSING: An AUTOFLOW run involves three separate modules: the general AUTOFLOW Input Controller, a language processor, and the master AUTOFLOW Processor. For a given computer system, the first and last of these modules are identical for all source languages. The master AUTOFLOW Processor accepts the output from the language processor and outputs the flowcharts and auxiliary listings.

A programmer can take three principal directions with AUTOFLOW to produce program documentation: full flowcharts and listings (either with ADR's patented two-dimensional approach or straight linear charts), compressed or higher-level charts, or CHART/AUTOFLOW.

Higher-level charts, which are obtained either through control cards inserted in the input run or through a separate language processor, emphasize the control and logic flow of the program by reducing the procedure-oriented code. For example, higher-level charts could more clearly point out the logic of which employees are selected for payment in particular runs of a payroll system, while the full chart would be required to illustrate the mechanics of how each pay would be calculated.

CHART/AUTOFLOW depends entirely on directive statements inserted by the programmer, using the comments facility of the language. Beautifully detailed and exquisitely worded charts can be prepared using CHART, but the programmer must accept the responsibility of making sure that the program agrees with the comments.

Input to the AUTOFLOW system consists of a source program and any control cards required to identify processing options in effect. Output consists of a title page, a straight listing of the input source program, a procedural statement label index, a transfer-of-control cross-reference table, diagnostics, flowcharts and any special listings unique to particular source language. Only the title page, transfer-of-control cross-reference table, and diagnostics are always produced; all other charts and listings are optional. Many of the fixed AUTOFLOW parameters can be changed by the user.

CHART FORMATS: In a full AUTOFLOW flowchart, a four-column format is used. The boxes are numbered, from top left to bottom right, by the program, so that any particular box can be identified throughout the flowchart, and referenced to the COBOL source program by line number, printed on the left.

There are four basic types of box used: 45-degree parallelograms for input/output, diamonds for single decision points, rectangles for general processing, and "circles" for jumps. The formation of these boxes is affected by the suitability of the printing characters used, with the asterisk-based diamond and circle being the weakest forms. The maximum number of characters on each line within a box ranges from 15 to 21.

A number of versions of the rectangular process box are used, depending on the source language. For example, in COBOL the addresses for PERFORM . . . THROUGH . . . statements give the AUTOFLOW coordinates of the starting and ending boxes vertically on the left and right, respectively; while the table boxes, such as GO TO . . . DEPENDING ON, simply list the possible cases under a general definition of the program statement.

AUTOFLOW II
Applied Data Research, Inc.

amount of detail in the charts by allowing the programmer to write notes about the program and have them inserted into the flowchart in the special-shaped boxes instead of having them always separate from the operations they are describing.

Automatic flowcharting does not save any computer time directly. However, the evidence over the past three years is that it does save programming time and reduce programming errors, so it can save computer time indirectly.

ASSEMBLY LANGUAGE

The Assembly/AUTOFLOW program is the oldest, and perhaps the most needed, of all the AUTOFLOW automated flowcharting programs. In most installations, assembly language is now being used only for programs or subprograms which for one reason or another cannot be handled in any of the standard languages.

The reasons why assembly-language programming has otherwise gone out of style are simply that it takes too long to write and that the resulting programs are too error-prone. Assembly/AUTOFLOW is one possible means to help redress the balance.

The error-proneness of assembly programs comes from their extreme intricacy, which can become downright unmanageable in the case of large, complex instruction sets such as that of the System/370. It has often been said that computer programs in fact are never completely debugged and that the system will always have errors in it which are slowly flushed out as seldom-used paths are one by one brought into operation. In fact, this is probably more true of assembly-language programs than of any other type.

Automated flowcharting so far is about the only technique that can help management feel reasonably confident that its programmers do have tools to safely write assembly programs. Flowcharters can detect and display many error conditions which otherwise either have to be paid for by keeping program-writing down to a crawl (like two effective instructions per hour!) or by putting up with problems during the operation of the program. These costs used to be reasonable when there was no way around them—but now that there is, the situation has changed.

Automated flowcharting can also be justified by the saving in documentation costs. Assembly documentation is both particularly important (for how many people can read someone else's assembly program?) and particularly time-consuming. There are more instructions to describe in an assembly program than in other programs, and their significance is far less likely to be immediately apparent.

FORTRAN

The success of the COBOL/AUTOFLOW program in becoming the leading proprietary program currently

The arrangement of the boxes is one of the key features of the AUTOFLOW charts. In the two-dimensional format advertised by Autoflow, one column is strictly in the order of source input. However, the other three columns consist of those parts of the program with which the jumps shown in the main column connect—no matter where in the program they appear.

As the spacing of the chart must be related to the longest column, this technique leads to longer (in terms of pages) flowcharts then straight-forward sequential flowcharts (which are also optionally available), but it can show the inter-relationships between various parts of the program in a way no other arrangement manages to do.

Only the flowchart symbols themselves are actually moved by the AUTOFLOW reorganization. Note statements are not moved; they are left in their original positions in the charts, and so they may not refer to the adjacent boxes.

COBOL/AUTOFLOW Format

The output of the program is primarily the flowchart. This can be complete with all the statements (as produced by the standard COBOL Processor); or it can have additional commentary added (by the use of the Chart Processor); or it can have much of the detail left out (by the use of the Compress Processor).

Particular "compression" possibilities include the elimination of arithmetic or move statements to bring out the logic and interconnections, elimination of decisions which do not transfer to paragraph names to show only major decision points, or elimination of specific paragraphs to reduce the volume (and perhaps the duplication) of the records.

In addition to the flowchart, a Procedure Division Summary is provided. This lists the non-computational or data transfer activities succinctly, allowing quick programmer checks and showing where in the main flowchart specific data has been printed.

Data names used in the program are cross-referenced in an alphabetical listing, showing their occurrence in the program and whether they are being modified, tested, or used in each particular statement. A Data Record Map with a tabular layout of all records is also provided.

Paragraph names are listed in alphanumeric order, showing the first box of each paragraph in the flowchart.

Diagnostics, including some syntax errors (missing periods, etc.) and any impossible breaks in the network (statements without entrances, altering lines that are not GO TOs, etc.), are listed if they occur.

Optionally, a listing of the complete input and of text material supplied by the user to amplify the flowchart is also provided.

Assembly/AUTOFLOW Format

While the boxes used for assembly-language flowcharting are standard AUTOFLOW, there are a number of differences about their referencing and contents which are specific to assembly programs. In fact, an assembly program is radically different from a high-level language program so far as flowcharting is concerned. In an assembly program the machine addresses are—for the purpose of the chart—practically in English. They can therefore easily be used in the text of the boxes, but it is not easy to provide any definition of their content. The machine operation, on the other hand, being in its coded form, is quite easily understandable, and its nature can be noted either by the shape of the box or in other ways.

AUTOFLOW II
Applied Data Research, Inc.

▷ available has in many ways overshadowed the position of FORTRAN/AUTOFLOW. The FORTRAN package was actually the second AUTOFLOW to become available, following the original work with assembly languages.

Perhaps because of its smaller number of users, FORTRAN/AUTOFLOW does not appear to have received the same amount of attention from ADR as COBOL. The FORTRAN listings currently provided do not have the cross-referencing strength of the Procedure Division Summary and other listings that are provided with COBOL, though in many ways the need for such cross-referencing is just as great in the FORTRAN language.

At the same time, there are some major virtues in the FORTRAN system, both directly and indirectly. The ability to add to the flowcharts the details of the implied instructions involved with the DO loops will help in diagnostic operations. The use of the same symbols as those used by the AUTOFLOW processors for the other languages allows a standardization of approach throughout an installation without regard to the languages used.

PL/I

PL/I AUTOFLOW has not achieved nearly the degree of usage as other AUTOFLOW systems because the language itself has not achieved widespread acceptance. Commensurate with PL/I's complexity of syntax, a number of auxiliary listing and flowchart compression options are provided to clarify source program structure and logic. Among these is a separate Introductory Comments section if four or more comments are encountered at the beginning of a program; a Duplicate Declaration Map; and lists of Signaled On-Unit Action Blocks, Get/Put Format Statements, etc.

CHART/AUTOFLOW

The Chart/AUTOFLOW programs provide, at the same time, a puzzle and a promise to management. Their theoretical basis is clear, and their implementations seems at least reasonably adequate—yet their practical importance is doubtful.

Ever since programming started, a basic problem has been keeping the program and the program flowcharts which describe it parallel and up to date. This has been handled quite successfully by AUTOFLOW and its horde of imitators. It can be genuinely said that automatic flowcharting of *programs* is becoming the standard way of life in well-run installations. But, while the lack of agreement between flowchart and program was certainly the first major problem, an almost equally serious problem was the lack of agreement between the higher-level block diagrams and the programs. This is the problem that Chart/AUTOFLOW approaches.

The basic trouble, according to the Chart/AUTOFLOW philosophy, lies in the fact that programmers are simply ▷

➤ By contrast, in higher-level languages the operation code equivalent is already in non-coded form and usable in English, while the addresses are normally hidden behind the data-names.

The assembly text, therefore, is not in as readable a form as the high-level languages, and it normally includes the entire instruction–operation code, addresses, indexing, and all–just like the program itself.

In addition to an English part of the flowchart showing the programmer's comments printed in the boxes with the instructions to which they were appended, there will also be a limited English-language translation facility. This is a sort of "verbalized decision table" that helps make the essential logic of the program clear.

Where a comment has been spread over a number of lines that also include instruction codes–which is quite readable in the program listing itself–the comment will be split up on the flowchart and can become somewhat misleading.

Each box is labeled with the box number together with any tag applied to the statement by the programmer. References in the boxes are specified in terms of the box and page number where the referenced instruction can be found. Treatment of macros requires the programmer to define the style of box to be used and the name of the macro.

The Chart mode of operation works in conjunction with the assembly flowcharting. In this mode a programmer can insert additional material into the flowchart in accordance with coded instructions inserted in the program.

In addition to the flowchart itself, the output includes a diagnostic listing and a cross-reference table. For assembly programs the diagnostic listing incorporates records of general errors in logic flow: incomplete branches, syntax errors, missing references, etc. It also includes references to basically uncomputable items, such as branching that is indexed by a register. The cross-reference table shows all references to modified addresses, all macros used, and a summary of literals. The diagnostic listing and cross-reference table can be used as supplements to the assembler for program debugging.

The particular diagnostics that are special to Assembly/AUTOFLOW are excessive use of assembler continuation cards, missed branches, and overflow of the user macro table.

For AUTOFLOW II, the Assembly processor has been completely rewritten. Although the Chart language remains the same, specific enhancements include: a COMPRESS facility that allows flowcharting only of specified statements, complete label cross-referencing–including both referenced and nonsymbolically referenced registers, new multiple symbol definitions for macros, special new handling for relative addressing, handling of conditional assembly instructions, handling of multiple CSECTS, and English-like text conversions (for example, MVC, A, B becomes Move B to A). Also, support now exists for RCA (or Siemens) disk DOS work files. Xerox Sigma Series assembly language support for output on a System/360 or 370 is also new.

FORTRAN/AUTOFLOW Format

The rectangular box has a number of versions in addition to the plain rectangle which is used for ordinary computations. Subroutine calls place the address of the subroutine (in AUTOFLOW chart references) vertically down the left of the box.

Treatment of DO loops is oriented to improve the use of the flowchart when checking out programs. The DO ➤

too busy (some say constitutionally unable) to document things more than once—or to realize all the necessary stages involved before they start coding. The solution, therefore, could be found by allowing them to document programs while they are coding them—or, better still, to use their coding to produce adequate documentation.

When assembly languages, with their fixed-format cards, were the most commonly used languages, providing such documentation was comparatively easy—it was simply written alongside the coding and listed out. ADR at that time saw that its flowcharting programs could simply pick up these comments, interpret them in the light of the various assembly instructions, and use them instead of a flowchart taken from the less readable instructions.

Later, when the various compiler languages came into use, it was still possible to use the Comment sections which all of them provide (and which the compilers ignore). However, by now the programmer had to add a few details to his notes, which gave him more work to do.

Yet, the basic idea was still there. Flowcharts understandable to management could be created automatically, and kept up to date. A created flowchart could be quickly checked for completeness, accuracy, and fulfillment of specifications—and without the programmer having to write everything out twice. Indeed, such a flowchart also showed whether or not the programmer's notes on the program provided a really adequate description of it.

And the need is certainly there. However, Chart/AUTOFLOW differs from most of the AUTOFLOW programs in that the thrust must come from management—not just the initial thrust to obtain the program, but also the thrust to keep it in use so that it becomes routine. Chart/AUTOFLOW is a management tool rather than a programmer's tool. It is one of the first attacks on an area which is still hurting, and as such it deserves serious consideration in a management context.

SYSTEM SUPPORT

The wide acceptance of AUTOFLOW in the marketplace has created a need for ADR to give special attention to the maintenance and support of its installed AUTOFLOW systems. Toward this end, an automated distribution system and a Software Error Information System have been developed. The latter is a means for ADR to track all known errors and identify the limitations of AUTOFLOW both to its own field personnel and to its customers. In fact, the AUTOFLOW users contacted by Datapro were very enthusiastic about the support ADR provides. Apparently it is not at all unusual for a program fix to be available within 24 hours after a trouble report has been delivered. Furthermore, ADR reports that support system development is being given great emphasis, thereby assuring that this important aspect of non-mainframe program product usage will continue to be one of the big plus factors for AUTOFLOW users.

statement itself is regarded as being a note rather than a procedural statement; it is therefore boxed ahead of the actual operations. Because the DO statement is really a very abbreviated shorthand representation of its true meaning, the note gives a much broader definition of the actual meaning than would otherwise be the case. The statement "DO 1000 1=1, K1" for instance, is shown as "BEGIN DO LOOP/1000 1=1, K1/SET 1=1" (slashes indicate new lines), which gives a clearer picture of what is actually happening.

At the end of the DO loop, a similar extension of the program as written is inserted, with the implied decision being flowcharted explicitly. However, the details of the specific test involved or the incrementation of the index variable are not given, and the box simply says END OF DO LOOP, with paths leading back to the beginning of the loop and on to the next statement.

ASSIGN and GO TO statements are also put into a much clearer form than in the program itself. Boxes are used with the statement placed at the top and a list of the various values involved placed below. Next to each value are shown the page and box number of the appearance of the branch concerned. Assign statements are placed in a special type of box that shows that the flow of the program has been altered.

The latest versions of the program include provision for asynchronous operations, using the FIND instruction. A special symbol, common throughout the AUTOFLOW charts, is used for this.

The input for AUTOFLOW/FORTRAN must be segregated into three areas before being read into the program. The first area consists of comments which are about the whole program rather than about specific statements. These head the programmer's deck, and are followed by all the function definition statements. It is necessary to enter all function definitions before any procedural statements, as AUTOFLOW/FORTRAN will chart any later definitions as algebraic assignments.

The output from AUTOFLOW/FORTRAN includes three major listings. The flowchart itself is the major one and is followed by a list of the various non-procedural statements: DIMENSION, EQUIVALENCE, COMMON, etc. This list simply reproduces the statements from the input program in the order in which they appear.

A Reference Listing is provided which relates the statement numbers with the AUTOFLOW boxes and notes the references to them in the program—also by AUTOFLOW box number. A separate listing itemizes the use of all subroutines within the program.

After these listings a series of diagnostics appears. The special FORTRAN diagnostics deal partly with the programmer's program and partly with any overloading of the FORTRAN/AUTOFLOW program. An error in a DO loop is noted where the loop is either incomplete or else is not inside any outer DO loop. Where the number of DO loops nested together exceeds ten—which should not often happen—the FORTRAN/AUTOFLOW program cannot handle the situation, and notes the fact in the table.

Similarly, where there is inadequate core storage available to handle dimensioned variables, the program aborts and should be re-run using a larger core partition.

In addition to these FORTRAN diagnostics, other general ones appear, such as cases where some internal connector appears to be undefined, names have been duplicated, etc. In addition to these listings, a copy of the source program can optionally be provided.

PL/I/AUTOFLOW Format

Flowcharting for PL/I programs is similar to that for the other languages. Comments, statement prefixes, and each

AUTOFLOW II
Applied Data Research, Inc.

USER REACTION

AUTOFLOW users contacted by Datapro generally share in ADR's opinion that the package is the complete program documentation and pre- and post-production tool. While no user has as yet had a chance to see AUTOFLOW II with CPA and ETC in action, the users contacted were well experienced (6 or 7 years of use was not uncommon) and highly opinionated. Their opinions can be summarized as follows:

- Many users feel that flowcharting facilities are of doubtful value today except as a tool for documenting the logic of older programs or of programs being converted. Less than 20% of the users currently enforce AUTOFLOW use in documentation standardizing, and one who does reports that he's encountered stiff resistance to using the system as a debugging tool. Another says that his experienced programmers would rather deal with the source code of an existing program than with a flowchart of it when modifying it. Some feel either that the flowcharts are too detailed or that flowcharts by themselves aren't enough. One simply stated that flowcharts are passe'.

- Some users feel that the cross-referencing facilities of ANS COBOL supersede those in AUTOFLOW.

- Many users expressed an interest in CPA, especially those going to data base systems (or already using one) and one who has many programs of 20 to 30 steps. One user wants CPA as a standalone program, and another says he'll only renew AUTOFLOW in order to get CPA.

- Most users can't see the point of ETC, and one who claims that he does, flatly states that in his opinion the option can't be cost-justified.

Datapro contacted users who represented both the earliest and the most recent ADR software customers. Without exception, all had praise for the firm's professionalism, for its technical support, and for AUTOFLOW's ability to perform as advertised. Many of the users, however, tend to see the traditional usage of AUTOFLOW as being no longer applicable to their needs, and many tend to disagree with ADR's new marketing philosophy for AUTOFLOW. Thus, it appears that there are some leaks in the AUTOFLOW dikes to be plugged, but ADR itself is strong and commands respect. Many of the AUTOFLOW users are also using the LIBRARIAN, METACOBOL, ROSCOE, and/or PI SORT, and they are generally happy with the firm and the products in use.

There are indications that if you run a tight EDP ship, documentation standards of the usual type will suffice. But AUTOFLOW II and its options can definitely ease the captain's job and will undoubtedly provide documentation of top-notch grade when the system is properly applied. It's up to each potential user to determine for himself just how much all that good documentation is worth to him.

program statement generate appropriate flowchart symbols. Perhaps more so than in any other high-level language handled by AUTOFLOW, auxiliary listings play an important part in producing understandable documentation. Many options in the PL/I language processor are directed toward producing higher-level flowcharts than clarify program logic by reducing the detail in one or more types of coding. Assignment statements and IF statements can be compressed. Comments can be excluded, and Declaration Statements can be printed out separately. In addition, the programmer can use an Ignore facility to exclude any group of coding.

Supplementary charts and special listings are included to further clarify the source program. A total of eight additional outputs are provided, including On-Unit Action Blocks, Called Procedures Cross-Reference, Signaled On-Action Blocks, Label Assignment Cross-Reference, Get/Put and Format Statements, Declaration Statements, Condition Prefix Map, and Duplicate Declaration Map.

If these listings are insufficient to describe your program to your satisfaction, or if you want flowcharts before coding has been generated, you can use the PL/I CHART module.

Chart/AUTOFLOW Format

Normally, the actual coding is ignored when the Chart mode is used. The input consists of a source program deck, which must include some specially written instructions for the Chart program in the comments sections which are ignored by the compiler. The source program itself is only of minor interest to the Chart program. The interest lies in nothing the comments made by the programmer to illustrate various parts of the program, formatting them suitably into decisions, processes, branches, etc., and producing a flowchart which contains them.

The various levels of flowchart which can be produced in this manner are limited only by the programmer's imagination, and the readability of the resulting charts is generally much higher than that of automatically created flowcharts. In fact, it can approximate that of the best manually produced charts.

The technique used is to provide a special Chart language, which the programmer uses when he is writing his notes alongside the program instructions. Like any other computer language, this consists of an instruction, a name, and an operand–generally an item of text for insertion on the flow-chart, such as "THIS BRANCH SHOULD ONLY OCCUR ON WEDNESDAYS."

The function of the instruction codes is to show how the text is to be formatted on the chart; i.e., what type of box is to be used. There are 17 standard AUTOFLOW boxes (for input/output, subroutines, decision points, etc.), and each of them can be generated by the use of one of the Chart instruction codes. The text "NOW WE LOOK FOR PART NUMBERS" could be wanted in an input/output box (if it is a new file of part numbers), or in a decision box (if it is a check that the records already matched referred to the same part number), or in a subroutine box (if handling the part number search is part of a subroutine), etc.

The flowcharter does not know how the programmer wants his chart prepared–so it has to be told. If, in the comment card, the first word is just the letter "P", for instance, then a processing-type rectangular box will be used; while if it is an "S", then the distinctive subroutine box, with its double line down the left side, will be used.

However, more items than simply the text and the box shapes must be provided for a reasonable flowchart to be created. Often the box itself will need to be named so that it can be quickly understood. Immediately after the

AUTOFLOW II
Applied Data Research, Inc.

▷ AUTOFLOW boasts an extremely strong customer base. This base will remain strong as long as users remain convinced of the value of flowcharts. The base will grow if users can be convinced that AUTOFLOW II with its options is a valuable and effective documentation tool. To that end, the Cross-Program Auditor (CPA) seems to hold the most promise, especially now that the days of data base systems are upon the user community. □

➤ instruction code therefore, and before the text, Chart/AUTOFLOW expects the programmer to write the name that he wants used. This is restricted to 10 characters (the box itself is normally only 21 characters wide).

Use of the name is optional, and each of the various languages has its own way of labeling boxes where the programmer has not provided a name. FORTRAN, for instance, uses instead the last statement number which preceded the comment cards being analyzed. This relates the box to the actual program, but not to the problem.

Even after the text supplied by the programmer for the flowchart has been appropriately boxed and labeled with its own name, it still has to be connected into the network of boxes that make up the flowchart. For this purpose a group of specifiers are used to indicate destinations, path conditions, etc. Normally, of course, the destination is simply the next box, but branches, subroutines, etc. are somewhat more complicated.

The output of Chart/AUTOFLOW is simply flowcharts, without any cross-reference lists of other items. Operationally, Chart/AUTOFLOW is a complete AUTOFLOW processing module, just as the various COBOL, FORTRAN, and other specific-language flowcharacters are. It simply works from the programmer's notes, rather than from the program text. Like the language flowcharters, it uses the standard AUTOFLOW front-end processor and output modules.

Cross Program Auditor (CPA)

AUTOFLOW II's CPA option is a system-oriented rather than a program-oriented tool designed to extract information common to any number of input programs. CPA can perform strong COBOL-oriented and more limited general source module analysis. It operates in a report mode and a query mode so that users can directly isolate areas of specific concern in a program maintenance effort. CPA is mainly a post-production tool, and can be most helpful in converting from a general file processing system to a data base system.

A CPA user, can, for example: find all occurrences of and references to a specific data-name, locate CALLs and ENTRYs across program boundaries, determine if and how a particular data item is altered to ascertain the impact of modifying that data item, and identify data sets used by individual programs. Of concern to a data base administrator, CPA can provide extensive structured cross-references of data among programs, simplify adding to or changing the format of data base information by informing the data base manager of the impact of changes, and supply comprehensive information on file structure and usage that can be critical when adding applications that will use existing files or when restructuring files. CPA can report on the total number of programs involved, the programmers responsible in each case, the extent to which functions will be affected, and the files to be used, and can also provide a glossary of all data names used within a system.

In converting to a data base system such as IBM's IMS or Cincom's (or Honeywell's) TOTAL, CPA can perform a data mapping function to determine where data is being duplicated and where naming conventions conflict. The information thus gained can be helpful in properly structuring the data base.

CPA can aid in enforcing conformance to standards and can locate even a series of words that may contain variables. It can pinpoint consistent inclusion of REMARKS, heavy use of system-dependent verbs (e.g., STOP, DISPLAY), poor subroutine usage, and use of inefficient COBOL verbs (e.g., EXAMINE...REPLACING ALL) in one report for management's review. It can verify that data definitions are consistent among programs.

COBOL-CPA includes the following features:

- Name Searches–printing and/or counting references to specified names within records.
- Area Searches–printing and/or counting references to specific data areas defined by location within modules.
- General Syntax Searches–printing and/or counting references to a given sequence of COBOL words which can include variables.
- File Activity Searches–extracting information relative to files, such as OPENs, CLOSEs, READs, and WRITEs, and sorting according to user-defined criteria (e.g., by FD name, external name, or alias).
- File Organization Searches–extracting detailed information on data items, such as name, level, and usage, and displaying the information to illustrate interprogram file usage.
- Special Searches–to locate common COBOL language statement usages, such as the STOP literal, that play a critical role in program conversion and maintenance.
- COPY Searches–locating COPY statements referencing a specified library name (or names) in selected program modules.

General-CPA has three features:

- General SCAN Search–listing and/or counting of a string of characters in any specified modules.
- INCLUDE Statement Search–listing of all INC control statements in modules stored in The LIBRARIAN, ADR's source program maintenance, retrieval, and security system; all included statements can be searched by CPA.
- Module Profile Search–sorting and listing information on modules stored in The LIBRARIAN by programmer name, account number, source language, date, and size. A special directive is provided that allows non-users of The LIBRARIAN to include such information within each module.

CPA will accept as input modules stored in The LIBRARIAN (OS or DOS version), in OS partitioned data sets, or in DOS source libraries. Or, the input can come directly from a DOS or OS job stream. A user must know the basic CPA commands, and, more importantly, discipline himself to query his systems via CPA as appropriate.

Extended Text Compositor (ETC)

ETC is an AUTOFLOW II optional feature designed to facilitate computerized production of textual material, especially in situations where the narrative documentation is constantly changing. Input and editing controls are analogous to the basic controls on a typewriter, and output is on a high-speed line printer. The most powerful and important of ETC's capabilities are the automatic ➤

AUTOFLOW II
Applied Data Research, Inc.

➤ table of contents facility and the ability to generate a comprehensive index of a document in accordance with user desires. ETC also has a macroizing facility for predefined word sequences, even ETC commands. The facility can be useful in producing tables and graphs. ETC has a full complement of the expected editing facilities.

PERFORMANCE: The following tables show representative AUTOFLOW speeds on IBM System/360 computers.

SYSTEM/360 AUTOFLOW SPEEDS
IN STATEMENTS PER MINUTE

Language:	Assembly or FORTRAN	PL/I or COBOL
OS-60K with Disk I/O or DOS-40K with Tape or Disk I/O (per CPU minute) for:		
Model 30	200	150
Model 40	450	350
Model 50	750	600
Model 65 & above	1100	900
OS-60K or DOS-40K with Card-In and Printer-Out (per wall clock minute) for:		
Model 30	125	100
Model 40	160	140
Model 50	190	170
Model 65 & above	215	200

SYSTEM/360 CHART/AUTOFLOW SPEEDS
IN COMMENTS PER MINUTE

I/O Mode:	Disk-In, Disk-Out	Card-In, Printer-Out
OS-60K (per CPU minute) for:		
Model 30	NA	NA
Model 40	500	190
Model 50	800	210
Model 65 & above	1300	230
DOS-30K (per wall clock minute) for:		
Model 30	120	90
Model 40	180	120
Model 50	240	140
Model 65 & above	350	180

The above figures, supplied by ADR, assume 2314 Disk Drives are used as work storage; with 3330 Disks, there is about a 10 to 15% speed increase for the tape-in and tape-out mode and about a 5 to 10% increase for card-in and printer-out.

The figures do not include DOS overhead, which is estimated at 40 sec execution (elapsed wall clock time), or OS overhead, which is estimated at 47 sec/execution (elapsed CPU time).

HARDWARE/SOFTWARE REQUIREMENTS: Essentially, AUTOFLOW will operate on any system on which the source language being charted can be compiled. In general, three to five work units (separate tapes or a disk file) are required.

PRICING: The AUTOFLOW II System is available on a monthly license, permanent license, or 3-year lease basis. The latter two plans can be paid for in a single payment, 3 annual payments, or 36 monthly payments. Maintenance is included in the cost with a monthly license. Under the permanent license, an annual maintenance fee of 10% of the single-payment price is charged after the initial year. The 3-year lease includes 2 years of maintenance, and is renewable at about 13% of the single-payment cost per year from the fourth year on. The effective base price for AUTOFLOW II under the single-payment plan for a permanent license is $3,630 plus the price of one of the first three options (processing modules) listed below.

To the basic charge, add the indicated amounts for:

Assembly and Chart/Assembly:	$ 990
COBOL (including Data-Name Cross-Reference):	2,750
Compress/Chart:	990
FORTRAN:	2,310
Chart/FORTRAN:	990
PL/I:	3,300
Chart/PL/I:	990
CPA (Cross-Program Auditor):	2,750
ETC (Extended Text Compositor):	1,980

Prices for the UNIVAC Series 70 are identical (except that there is no PL/I processor). Pricing for other processors and computer systems follows a similar pattern.

The lease charges can also be paid in yearly or monthly installments. The annual-payment plan calls for three payments of about 35 to 37 percent of the single-payment charge. The monthly-payment plan calls for 36 payments of about 3.25 percent of the single-payment plan. Maintenance is included for 2 years at no additional cost; thereafter, annual maintenance is billed at 10% of the single-payment price.

After the expiration of the initial three-year lease, renewal is on a year-to-year basis for about one-seventh of the single-payment charge each year. (There is no charge for the COBOL and FORTRAN Chart or other auxiliary modules after the first three-year period.)

The Metered System Usage plan provides for a minimum charge of $100 per month and a charge of 3¢ per card for the first 13,000 card images processed, and 2¢ per card for additional card images processed. A "purchase option" for conversion from metered usage to an unlimited lease is available that ranges from 100% of the monies paid if converted during the first month to 50% of the monies paid during the first six months for conversion at any time thereafter.

INITIAL DELIVERY: COBOL, Assembly, and FORTRAN/AUTOFLOW were all delivered at different times within 1966. CPA and ETC were first delivered in the second quarter of 1973.

CURRENT USERS: There were about 1800 AUTOFLOW installations of all types as of September 1972. Of these, there were about 1000 Assembly/AUTOFLOW users, about 1200 COBOL/AUTOFLOW users, about 250 FORTRAN/AUTOFLOW users, and about 1000 Chart/AUTOFLOW users. As of May 1973, ADR said these figures were essentially unchanged. ■

ROSCOE
Applied Data Research, Inc.

MANAGEMENT SUMMARY

Applied Data Research designed ROSCOE (Remote OS Conversational Operating Environment) to aid large computer installations in accurately measuring and reducing their program development costs. The system was developed and is used at ADR to facilitate program development and to distribute associated resource utilization charges to the proper accounts. ROSCOE extends the programmer access facilities normally provided in a closed-shop batch environment to any number of programmers using local and/or remote terminals. An elaborate, powerful programmer-oriented command structure eliminates handling of program decks and allows programmers to edit their work on-line under control of ROSCOE.

Using ROSCOE, programmers thus experience a "hands-on" environment without tying up a computer room. ADR has found that this simulated hands-on environment, right in the programmer's office area, both improves programmer morale and reduces the elapsed time for program checkout.

ROSCOE was announced in January 1970 and immediately began competing with IBM's Conversational Remote Job Entry (CRJE) system. This no-charge IBM system was not deliverable at that time, but was widely promoted, thus contributing significantly to the cool initial reception received by ROSCOE. ADR stopped actively selling ROSCOE in June 1970 and filed suit against IBM to stop the sale of a "ghost," resulting in an out-of-court settlement to ADR in August involving about $2 million. CRJE was finally delivered at about that time, and ADR evaluated the IBM product for several months and developed comparative analyses prior to resuming active marketing of ROSCOE in the first quarter of 1971.

But ROSCOE/CRJE competition (although ROSCOE had the advantage) is now a moot point, because IBM support of CRJE terminated with the availability of OS/VS1. ROSCOE now competes against IBM's TSO – but in a very special way.

Although ROSCOE and TSO (Time Sharing Operating system) have similar facilities, they are conceptually quite different. TSO is a generalized time-sharing system, while ROSCOE is a low-overhead remote batch system. It would be difficult for an installation to cost-justify TSO unless it has the need for on-line mathematical problem solving and on-line operating system maintenance and control. But if the only need is for remote preparation, testing, and maintenance of source programs, ROSCOE can provide the necessary support at a far lower cost than TSO.

ROSCOE creates a "hands on" environment at multiple remote terminals for program development, testing, and maintenance on IBM System/360 and 370 OS and OS/VS systems. It permits syntax checking of COBOL, PL/1, and FORTRAN programs and of JCL. Its monitor subsystems provide OS data set management facilities.

CHARACTERISTICS

SUPPLIER: Applied Data Research, Incorporated, Route 206 Center, Princeton, New Jersey 08540. Telephone (609) 921-8550.

FUNCTION: ROSCOE is a remote job entry system, written in IBM System/360/370 Assembly language, that provides remote terminal users with full access to an IBM computer under OS or OS/VS for program development, testing and maintenance. As a programmer's tool, ROSCOE permits users to automatically generate JCL statements and job streams with syntax checking of COBOL, FORTRAN and PL/1 statements; scheduling of compilation and testing from a remote terminal; and recovery of compilation and test results at the terminal. ROSCOE keeps statistics on programmer activity that include, at management option, system resource utilization as well as time and frequency of system access by programmer and account number. Two monitor routines, COPY and UTILITY, provide comprehensive data set retrieval and management.

OPERATION: ROSCOE is loaded/terminated through the computer operator console as a "never-ending" job in a partition or region under IBM's OS. Numerous programs can be developed concurrently from multiple remote terminals under ROSCOE, with an Active Work Space (AWS) and a user library associated with each terminal. A conversational command system allows the user to manipulate his AWS area and library files, with job submission, initiation, and output recovery all controlled at the remote location. At the user's option, an eight-character password can be used to protect his AWS and library from unauthorized access.

Users' Active Work Space (AWS) and library storage areas are assigned as they log into the system. ROSCOE drops trailing spaces or blanks, thus reducing library data set storage space in main memory and on direct-access devices. This data compression is not optional, and ROSCOE does not take advantage of extra disk or main memory availability in larger installations. However, the compaction saves about 50 to 60 percent of the disk storage requirement and, more importantly, permits a corresponding reduction of about 50 percent in disk accesses for reads and writes.

The AWS handles card-image data and is organized into 80- or 84-character lines at user option, with a 4-character line ID number. Manipulation or editing of the AWS is done implicitly by line number, or through explicit conversational ROSCOE commands. These commands permit assignment or renumbering of line numbers; display, modification, insertion, or deletion of all or part of the AWS by line

ROSCOE
Applied Data Research, Inc.

▷ USER REACTION

ROSCOE users contacted by Datapro were very well pleased with the performance of the system and reported that all vendor claims were true. One particularly large (2-million-byte) installation expressed regret that the ROSCOE system was not set up to take advantage of larger main memory and disk availability, finding that the memory-saving storage techniques used by ROSCOE were not necessary for them and that ROSCOE service would be faster if they were not employed.

The important thing, though, according to other users, is ROSCOE's speed when compared to the slow response times that would result from the strain that TSO would place on the system. Users contacted recently comment that ROSCOE is stable, without problems, and has often completely eliminated JCL syntax errors. They also favor its powerful edit facilities that make mass changes to programs much less of a problem, the limited use of resources by ROSCOE, and the ease with which programmers learn to use it. In summary, installations that have about 20 or more programmers and are running large volumes of batch processing should take a serious look at ROSCOE. □

➤ number; character string manipulation for single or multiple lines; and tab settings so that remote keyboard data entry is automatically inserted at user-specified positions in the designated AWS line. ROSCOE reserves a basic set of 1000 AWS lines (80K bytes) on direct-access storage.

The User Library is organized into named data sets and provides permanent storage facilities for user data. Individual data sets are protected by a prefix associated with the user's ROSCOE authorization code. Each user data set is manipulated as a unit, and the SAVE and FETCH user library commands allow the transfer of complete data sets from the AWS to the library and vice versa. Other commands that operate on the User Library provide for listing and updating the data sets, as well as combining them in various ways.

Frequently-used sequences of ROSCOE commands can be stored in the User Library under a data set name. Such a sequence of commands is known as a ROSCOE conversational procedure or ROSPROC, and can contain a certain amount of logic or decision-making capability. Of particular interest is the CALL command that permits one ROSCOE procedure to call another, with linkage beginning and ending at user-specified statements in the called ROSPROC. This CALL command greatly enhances the logical capabilities imbedded in each ROSPROC.

Actual control over the running of users jobs is handled through the SUBMIT, LIST, and DISPLAY commands. The SUBMIT command enters a single job or a job stream to OS for batch processing, with appropriate JCL entered either through the remote terminal keyboard or fetched from a user library data set under remote terminal control. ROSCOE edits each input job stream and supplies a /* card according to standard JCL conventions. Input without a valid JOB card is automatically rejected.

The LIST command is used to recover output from a job at a remote terminal. Compiler results can be extracted for terminal display through the use of a COBOL, FORTRAN or PL/1 post-processor that is stored in the system library as a cataloged procedure. An EXEC statement invokes the appropriate procedure and extracts the source statements that are in error, with corresponding diagnostic messages, and prints them at the terminal.

The DISPLAY command allows OS system messages pertinent to a given job to be displayed at the appropriate terminal.

ROSCOE includes syntax checkers for COBOL, FORTRAN, PL/1 and JCL statements. These syntax checkers can operate upon single statements in an incremental mode, or entire programs or program divisions can be checked.

ROSCOE's copy routine permits on-line retrieval of OS data sets. The UTILITY monitor routine performs ALLOCATE's, SCRATCHes, and CATALOG's on OS data sets; BUILD's and DELETE's entries in the OS catalog; RENAME's OS data sets; FIND's a catalog entry; LIST's the VTOC entry for a data set; and WRITE's an OS data set on a direct access device. The functions can be password-protected.

HARDWARE/SOFTWARE REQUIREMENTS: An IBM System/360 or 370 with at least 256K bytes of main memory, operating either under OS/MVT or MFT with HASP, or under OS/VS1 or OS/VS2, operating in virtual storage under control of the paging supervisor. Because ROSCOE is I/O-bound, the program can be executed in either main memory or LCS (at the same speed); but it cannot be rolled in/out or time-sliced with other regions or partitions. A typical 6-terminal IBM 2741 ROSCOE system occupies 70K bytes in a main memory partition or region including OS control requirements, but exclusive of syntax checkers. The FORTRAN Syntax Checker requires 6K bytes, COBOL requires 12K, PL/1 requires 18K, and the JCL Syntax Checker requires 8K bytes of main memory. The UTILITY subsystem requires about 20K bytes, which can be shared with the syntax checkers. Each additional ~~terminal requires about 2K bytes of main memory for local~~ areas and buffers.

ROSCOE libraries can be set up on any direct-access device (with about 4000 records stored per 2314 cylinder), using data compression techniques. Typical medium-size ROSCOE installations (about 6 to 10 users with 50 to 75 programs and about 400,000 source statements) require about one-half of 2314 pack to be reserved for ROSCOE data sets. ROSCOE supports hard-copy devices such as the IBM 2741 or Teletype 33/35 and locally-attached CRT display terminals such as the IBM 2260 and 3270. ROSCOE is optional for local 3270 support; remote 3275's are not supported.

PRICING: ROSCOE is available on a permanent license for $18,000, including one year of maintenance. Subsequent maintenance costs $1,800 per year. A monthly lease fee that includes maintenance for the life of the installation is available for $1,000 per month. ADR supplies source code, documentation, training, installation (of a tailored version, in about an hour), and special interfaces to ADR's LIBRARIAN and MetaCOBOL.

INITIAL DELIVERY: June 1970.

CURRENT USERS: 25 as of December 1973. ■

MetaCOBOL
Applied Data Research, Inc.

MANAGEMENT SUMMARY

MetaCOBOL accepts COBOL source programs interspersed with macro calls designed to speed up the writing, testing, and debugging of programs; it produces an expanded COBOL source program. Other effective uses include conversion between COBOL dialects and addition of new functions, perhaps tailored to a specific industry.

COBOL has indeed achieved the impressive intent expressed in its name. Common Business Oriented Language. There are specific frills to learn when you cross from one machine to another, but for practical purposes, COBOL truly exists as a standard language. It's the language you expect a programmer to know when you hire him for business-oriented applications programming.

But COBOL is not without shortcomings and, consequently, critics. Perhaps the loudest of all criticisms leveled against COBOL is its sheer verbosity. Sometimes it seems that you'll never finish the confounded Data Division, much less progress into the Procedure Division. When you make it that far, you long for the bygone days of assembly-language coding when, with just a few characters per command, you could really crank out the coding. (Programming was actually slower, it just seemed as if you progressed faster.)

This shortcoming of COBOL has given rise to two distinct types of program packages. One of these types is a programming system making use of highly formatted "input sheets" to capture the characteristics of the files and their handling. An actual COBOL source program is then generated from these specifications. The second type makes extensive use of abbreviations and other conventions to reduce coding effort but retain the flavor of COBOL programming. Again, a valid COBOL source program is the result.

MetaCOBOL is a sort of synthesis between these two approaches. Through a comprehensive macro capability, you can create abbreviations to your heart's content. The same macro capability also allows you to create programming shortcuts, routines, or even languages at will, thereby duplicating the intent, if not the format, of the parameter-driven systems.

Input is COBOL with imbedded macro calls; output is a COBOL source program ready for compilation. Extra-cost items facilitate generation of test data files, debugging, and monitoring of program characteristics.

Since the first installation of MetaCOBOL in September 1970, the package has rapidly gained a wide following and enjoyed a good reception in the marketplace. Apparently this acceptance is well earned; all of the users contacted by Datapro reported that they are happy with the performance of MetaCOBOL and with the prompt maintenance and program fixes that accompany it. Among the uses to which MetaCOBOL is being put are:

- Enforcement of programming standards and acceleration of program development by taking advantage of the MetaCOBOL abbreviation features.

▷

CHARACTERISTICS

SUPPLIER: Applied Data Research, Inc. Route 206 Center, Princeton, New Jersey 08540. Telephone (609) 921-8550.

BASIC FUNCTION: To simplify coding of COBOL source programs for IBM System/360 and System/370 computer systems. This aim is accomplished through a macro facility which can be used to reduce the effort of programming sections of coding often repeated, eliminate much of the wordiness of COBOL, and conveniently expand the programming capabilities of COBOL programmers.

OPERATION: MetaCOBOL accepts a COBOL source program that has macro definitions and usages interspersed with conventional COBOL coding and produces a listing of the input, a listing of the resulting COBOL program, and a COBOL program ready for compilation. Extra-cost modules can be added to assist in testing and debugging the resulting COBOL program, as well as in evaluating the performance of the program in a production environment.

Distinction needs to be made between subroutines and macros and between the use of macros and the generation of macros. Disregarding classical and formal semantics, we can distinguish between macros and subroutines primarily by the way in which a programmer codes their usage. A macro is generally used in place of many lines of coding, as a sort of abbreviation. It generally looks like the rest of the lines of code. The actual lines of code are substituted for the macro at some later time. A macro generally cannot be distinguished from surrounding native code unless you are familiar with the source language.

A subroutine, on the other hand, usually uses a rather formal series of commands and parameter lists to cause a branch in the program to another set of coding and a return when completed; all the coding generally appears in the source program when written.

Speaking in generalities is necessary because of the many exceptions, such as the functions in FORTRAN, which are subroutines but look like macros, or the macro procedures in some assemblers that are so formal they look like subroutine calls.

As ADR uses macros in MetaCOBOL, they represent direct substitutions of coding, either entered ahead of the application coding or stored in a library, for special groups of words used in the application coding.

In the definitions of the macros, the programmer can make use of conditional statements to selectively control the coding that is generated for each occurrence of the macro in the application coding.

➤

- Easing conversion problems involved in transferring COBOL dialects by using word or syntax substitution and flagging potential problem areas.
- Development of new programs by using macros to generate specific program segments.

You know, this sounds almost too good to be true. Apply a little intelligent effort toward developing the appropriate macros, and bingo, your COBOL compiler becomes an all-purpose tool for doing everything from producing report generators to creating a special language for eigenvalue problems.

To a large extent,these capabilities are made practical with MetaCOBOL. But there are several points that will require your evaluation before you can decide whether or not MetaCOBOL is for you. Let's discuss the topics of programming discipline, language capabilities, and machine time.

Programming discipline is a difficult area to discuss. Every installation thinks it enforces good discipline, but few actually do. The chief objective is to eliminate run time and modification problems as far as practicable while maintaining a high level of output (both quantity and quality). The way to this objective is seen by many as channeling the creative efforts of programmers into problem solutions rather than mundane, repetitive aspects of program generation. MetaCOBOL is admirably equipped to assist in this goal. However, its flexibility can cause problems. The same programmers who delighted in exploring the non-standard aspects of IBM 1401 instructions will have a fine time with MetaCOBOL. Proliferation of special-purpose macros can lead to confusion among the programmers. In all fairness ADR recognizes this problem and has adapted its training program to minimize the problem. Nevertheless, the added flexibility does add to problems of control in order for the package to be beneficial. After all, you can't give a man a 300-horsepower car and expect him never to exceed 50 mph.

In addition, two source programs may need to be maintained: the MetaCOBOL program and the COBOL program. Each individual installation will need to set up its own standards for retaining or discarding the MetaCOBOL source program, as well as whether program modifications are to be made via MetaCOBOL or directly through the COBOL program. ADR encourages customers to test, debug, and maintain their programs only at the COBOL level.

If you plan extensive development of new languages, then MetaCOBOL should be scrutinized carefully. True, you can handle such a requirement with the powerful macros in the Translator—but, in general, don't plan to code anything you would not ordinarily code in COBOL. However, with MetaCOBOL, many common

MACRO DEFINITIONS: There are two parts to a MetaCOBOL macro. The prototype is, in essence, the name of the macro; specifically, it is the part that is used in the application coding to identify the macro. The model for the definition is the coding that will be substituted each time the MetaCOBOL processor finds the prototype in the application coding.

There are three types of macros defined for MetaCOBOL: word, prefix, and string.

The word macro allows a single word to stand for a fixed group of words. A typical usage would be to allow one word, such as CSECT, to represent the entire Configuration Section of the Environment Division.

The prefix macro is associated with a single word in the user coding; that word is dropped into the model of the prefix definition in the place specified by an ampersand (&). This permits the user to set up convenient coding abbreviations such as having BLOCK CONTAINS 10 RECORDS represented by BCR=10.

The string macro is associated with patterns of words and symbolic operands in the user's coding. Symbolic operands are dropped into a prototype model of the string macro in the place(s) specified by an ampersand or other special notation. This permits the user to set up convenient notations to handle complex source-language structures. The lines of code that form the model of the macro can utilize comparisons, conditional and unconditional branches, value assignments, subroutine calls, and other directives to create the coding that goes into the user program. These internal macro directives can be used to streamline standard COBOL statements, to generate additional functions for a specific application area, or to create entirely new language facilities. (Theoretically, an entirely different language could be created, with compilation being done by your conventional COBOL compiler; actually, this would be an arduous programming task, and the resulting object-code efficiency would probably be poor.)

Up to 15 symbolic operands can be used simultaneously. In addition, any number of internal variables can be defined within macros. The variables can be global (for referencing from another macro) or local (defined only for the macro in which they appear). Symbolic operands can be qualified as names or literals, with or without subscripts. Macros can be imbedded within other macros. Formal linkage can be established between macros, complete with equivalencing of variables; this facilitates use of library or other macros as building blocks.

In general, the coding for the model of a string macro has a definite COBOL flavor of its own, both in the manner of coding and the power of the directives.

Extensive facilities are provided for generating out-of-line coding, particularly for entries belonging to the Data Division. This permits data elements to be defined as they occur in the Procedure Division or in a macro model. The out-of-line coding is inserted in the proper place during the MetaCOBOL run. Also, 80-column output can be generated either ahead of or after other output and optionally placed on tape or disk as a data set.

Two annotation directives are included. The NOTE directive causes a diagnostic code to be printed along with a message included in the macro coding. The FLAG directive is similar; it causes the first word of the macro prototype to be printed whenever the macro is en-

MetaCOBOL
Applied Data Research, Inc.

```
MetaCOBOL                           COBOL

PRINT BODY BY 2.                    WRITE BODY AFTER ADVANCING 2 LINES.
                                    MOVE SPACES TO BODY.
                                    ADD 2 TO LINE-COUNT.
                                    IF LINE-COUNT IS GREATER THAN 56
                                      PERFORM PAGE-HEADING-ROUTINE
                                      MOVE ZEROS TO LINE-COUNT.

CLEAR WORK-RECORD.                  MOVE SPACES TO WORK-RECORD.
                                    MOVE ZEROS TO AMOUNT OF WORK-RECORD.
                                    MOVE ZEROS TO HOURS OF WORK-RECORD.
                                    MOVE ZEROS TO RATE OF WORK-RECORD.

CANJOB 3.                           ENTER LINKAGE.
                                    CALL 'CANJOB' USING 3.
                                    ENTER COBOL.
                                    STOP RUN.

INPUT-FROM CARDS-IN                 OPEN INPUT CARDS-IN.
  INTO CARD-IN.                     PERFORM ZZ-INIT-ZZ.
                                    GO TO ZZ-OPEN-ZZ.
                                ZZ-READ-ZZ.
                                    IF ZZ-LR-ZZ = 'N' CLOSE CARDS-IN STOP RUN.
                                    READ CARDS-IN AT END MOVE 'N' TO ZZ-LR-ZZ.
                                    GO TO ZZ-BREAK-ON-ZZ.

LISEZ LA-CARTE,                     READ LA-CARTE,
  A LA FIN ALLEZ A SORTIE.            AT END GO TO SORTIE.
MULTIPLIEZ LES-HEURES PAR LE-TAUX   MULTIPLY LES-HEURES BY LE-TAUX
  POUR DONNER LE-SALAIRE.             GIVING LE-SALAIRE.
DEPLACEZ PIERRE A COLETTE.          MOVE PIERRE TO COLETTE.

01  TRAN-SORT-KEY.  02 SEQ-KEY.     01  TRAN-SORT-KEY.
03  ACCT.  04 ACCT-KEY.                 02  SEQ-KEY.
05  ACCT-NO P=X(10).                      03  ACCT.
                                            04  ACCT-KEY.
                                              05  ACCT-NO   PICTURE IS X(10).
```

The power and ease of use of MetaCOBOL can be seen in these comparisons between MetaCOBOL statements (on the left) and the resulting COBOL source code that is generated (on the right). A stored library of macro definitions can also be used to handle "home-made" extensions to COBOL or convert source-language statements between different levels of the language. The French-language statement conversion to COBOL shown above is an extreme example of this macro facility.

▷ tasks that you might not ordinarily want to program as subroutines can be handled conveniently.

Expect your machine time for initial compilations to double when you go to MetaCOBOL. The installations that perform extensive compilations because of a large number of one-shot program requirements will need to consider this carefully. It may be that special programs can be conveniently programmed through MetaCOBOL, so that the need for extensive compilations can be reduced.

If you are facing a tight budget (and who isn't these days?) then the versatility of MetaCOBOL may allow it to take the place of several specialized packages, even though the result may not have the speed or slickness of the specialized packages. Just how much MetaCOBOL can do for you depends primarily on your own ingenuity. □

➤ countered. A typical use of the NOTE directive is to warn the programmer of certain conditions. A typical usage of the FLAG directive is to flag all occurrences of a particular word or phrase. These directives enable user-coded diagnostics to be included in a program.

When written, a macro is assigned to one or more of the four COBOL divisions. Only that division is searched for occurrences of macro prototypes (names) that have been identified either through direct coding or calls from a library. All coding not associated with a macro is treated as pure COBOL and is transmitted unchanged to the output. Optionally, MetaCOBOL can check the syntax of any non-macro statements.

OUTPUT: The output listing contains macro definitions, the user's MetaCOBOL source program, and the equivalent generated COBOL program. If desired, any of the listings can be suppressed; this is desirable to speed processing when a library of macros is used.

Each line of the MetaCOBOL program (including macro definitions) is automatically assigned a sequence number. These numbers are used to create sequence numbers within the COBOL program. All lines inserted because of a macro carry the same number as the line in the MetaCOBOL program that used the macro. Out-of-line entries carry the number of the user-coded line that used the macro in which they were defined. Thus, correlation of the COBOL output coding and the MetaCOBOL input coding can be established. Optionally, the COBOL statements can be resequenced (i.e., assigned consecutive sequence numbers), but correlation between input and output is lost. ➤

MetaCOBOL
Applied Data Research, Inc.

➤ The format of the output listing of the COBOL program is standardized, with up to six levels of indentation and other rules to promote easy reading. The user can vary his output formats according to his own standards at either installation or run time by a format parameter.

At the end of the MetaCOBOL coding portion of the listing, statistics are listed showing the number of errors, warnings, and user-code notes and flags; the size of the generated symbol, macro, and data tables is also shown.

INPUT: Input to MetaCOBOL is a series of card images containing macro definitions, library macro calls, and user coding in COBOL and macro language. Macros are identified by a W (work type), P (prefix type), or S (string type) in column seven; Macros are terminated by a new macro or a COBOL division header.

The dialect of the COBOL language accepted by MetaCOBOL is optional through the use of a directive and a set of macros. Standard provisions are included for IBM System/360 ANS, DOS Level D, and OS Levels E and F. Several users are successfully entering Honeywell and other lower-level implementations of COBOL for conversion to ANS.

TEST DATA GENERATOR: This extra-cost option permits generation of test files from the Environment and Data Division entries with simple statements. Any number of record formats can be included in a file, and any number of files can be generated. File sequence can be specified, and any valid COBOL file organization can be generated. Data values can be varied or held constant from record to record. They can be varied randomly within a range, clustered about a midpoint, or computed from other fields in the same or other files.

A special variation of the random generation includes invalid entries to exercise the program logic. Applied to files, this facility causes some out-of-sequence records to be generated. Applied to data fields, it causes invalid data types to be occasionally generated.

RUN-TIME DEBUGGING AID: This extra-cost option replaces the IBM debugging aids EXHIBIT and TRACE, and consists of two commands: SNAP and ABEND.

SNAP allows coding snapshots of selected data items; naming a group item automatically includes all related elementary items. Editing of output is performed to increase the intelligibility of printing. Numeric fields are displayed in both character (with asterisks replacing nonnumerics). All elementary fields are reported in both "display" and hexadecimal modes.

SNAP commands can be made conditionally related to other elements of the source program to more precisely control printout of data items.

The ABEND command allows output of a listing after each abnormal termination, containing a specified number of the most recently executed paragraphs, the program status word, and the termination code. If the cause was a data or arithmetic exception, the result of the exception instruction will be set to 1 and processing continued with the next instruction; a limit can be set to specify the maximum number of exceptions of this type that will occur before program abort. Other abnormal terminations cause a listing to be printed and an immediate abort.

The strength of these commands in comparison to the IBM ones is that they are effective when a bug occurs and thus can be used the first time through. The IBM commands output about the same things but are far less flexible; generally, they are used after a bug has been known to occur.

The Debugging aid consists of three assembly-language modules and a set of MetaCOBOL macros. Source code is generated that will cause the necessary object code to be generated when the program is compiled. Recompilation (through the COBOL compiler) is necessary to remove the debugging commands. If desired, the debugging commands can also be included as COMMENT cards and left in the program.

Of particular interest is the COBOL Performance Monitor (CPM). The CPM allows the user to sample the execution of each paragraph in a COBOL program at execution time. A single Procedure Division command is used to invoke the CPM, which then operates through the Translator to embed special statements in the resulting COBOL source program. Two reports are produced: a Complete Paragraph Summary that sequentially lists each paragraph with various usage statistics, and a Leading Paragraph Summary that identifies the most active and least active paragraphs.

PERFORMANCE: ADR states that the MetaCOBOL processor runs at about the same speed or slightly faster than the IBM ANS COBOL compilers, because it performs the same kind of input analysis to determine output. Extensive source input volumes or punching an output deck will slow it down due to the limitations of the peripheral devices. On the other hand, extensive macro expansion will make it faster than the compile run. Overall, the use of MetaCOBOL will roughly double your initial compile time (once through MetaCOBOL and then through the compiler).

HARDWARE/SOFTWARE REQUIREMENTS: MetaCOBOL Version 5, released in late March 1972, runs on an IBM System/360 or System/370, Model 30 or larger, with at least 65K bytes of main memory. It will run under either DOS or OS. A COBOL compiler is naturally required to make use of the output from MetaCOBOL. Any of the standard IBM COBOL compilers (ANS, DOS Level D, and OS Levels E and F) can be used. It should be a simple matter to accommodate new COBOL releases. Version 6 of MetaCOBOL is scheduled for release in January 1973.

PRICING: A license for the use of MetaCOBOL costs $7,500. The Test Data Generator, Run-Time Debugging Aid, and COBOL Performance Monitor each cost $2,000. The price includes installation, training, and one year of maintenance. Monthly plans, quantity discounts, and purchase options are also available; contact ADR for details. The training consists of a three-day class/workshop program. ADR advocates a two-level training approach. Senior personnel are taught the full facilities of MetaCOBOL. Junior personnel are taught only the basic portion, such as use of standard macros and abbreviations, to prevent them from getting in over their heads.

In addition to new releases, ADR circulates a newsletter that contains user-contributed macros; this is potentially a valuable way to conserve your programmers' time.

INITIAL DELIVERY: September 1970.

CURRENT USERS: 116. ■

Extracto
Aquila BST (1974) Ltd.

MANAGEMENT SUMMARY

Extracto, a Canadian product, is marketed throughout the U.S., Canada, and Europe. It was developed by the Canadian firm Aquila BST (1974) Ltd., now a subsidiary of System Development Corporation of Santa Monica, California.

The original Extracto package was developed in 1969 and marketed exclusively in Canada. In early 1971 marketing began in the U.S. The U.S. market has been hard to penetrate because of the many report generator packages already on the market. In spite of this, Aquila has been able to find dozens of U.S. companies willing to go to the trouble of evaluating the package in-house, and many of these have, in turn, become satisfied users.

At the present time, Extracto offers a dynamic DTF generator for the DOS version and a table look-up module in the print program, both free, and an extra-cost multi-file handling module (the SUPERIO option).

Extracto is primarily designed for retrieving information from existing files for many users rather than for building and maintaining a data base. In essence, Extracto is a ready-to-execute program written in assembly language. The user completes the program by writing a series of statements that supply the parameters needed to specify which records will be extracted, what will be printed, and where. A comprehensive set of relational operators, along with AND and OR conjunctions and arithmetic computations, enable the user to specify complex relationships for record selection.

Tabulated reports, which produce one printed line for each selected record, form the basic output. Three additional functions generate multi-line reports (up to 99 different output lines per selected record), create a new file from selected records, and perform limited updating of the input file.

Extensive titling, heading, and footing information (up to 16 lines each) can be included. In addition, any textual comments desired can be output in the validation listings of the Extracto statements, which accompanies each run.

The statements specifying the report generation are free-form and nonpositional (except for headings). Typically, the specification of a report line will spread over several cards, but it is highly readable. Compactness of form has been sacrificed for ease of use—a worthwhile trade-off for a system geared to nonprogrammers.

Probably the most impressive marketing claim for Extracto is the number of reports that can be prepared from a single pass of the data file. Up to 250 requests can be serviced at the same time, and each request can include any or all of the four report and file functions. However, it takes a large computer system to accommodate the full capacity. Extracto can be used on a basic 24K Honeywell Series 200, a 32K IBM System/360 or 370, a 32K UNIVAC Series 70 or 9400/9480, or a small UNIVAC 1100 Series computer system. Provided that enough tapes and/or disk drives were available to handle the work files

Extracto can generate multiple independent reports from one data file pass and can optionally create auxiliary files and perform some update functions. It runs on IBM System/360 and 370, Honeywell Series 200, and UNIVAC 70, 9000 and 1100 Series computers. Conversational Telextracto and Indexing (subscripting capabilities) are other options.

CHARACTERISTICS

SUPPLIER: Aquila BST (1974) Ltd., P.O. Box 10, Stock Exchange Tower, Montreal, Quebec. Telephone (514) 866-5841. Also, Insurance Systems of America, 3050 Johns Ferry Road, P.O. Box 47975, Atlanta, Georgia 30340. Telephone (404) 449-3950.

BASIC FUNCTION: To extract data from a file for report writing, file creation, or file updating. Extensive data editing and manipulation are permitted for all functions, though only constant information can be used to "update" a file. With the optional multi-file read module, however, input data from one file can be used to update another. Versions are available for Honeywell Series 200, IBM System/360 and 370, UNIVAC Series 70, UNIVAC 9400/9480 and 9700, and UNIVAC 1100 Series computers.

OPERATION: Extracto is a parameter-driven program rather than a program generator. Parameters entered through free-form statements constitute a Request. Multiple Requests can be processed against a data file in one pass of the file. Each Request can generate a Tabulated Report (one output line for each selected data record), a Multi-Line Report (multiple lines for each selected record), a new file, and/or an updated version of the old file.

Six groups of cards form the input for a request. The first card set contains comments which are subsequently printed in a listing, forming a convenient way to document a request. This group of statements also allows for definition of work fields and/or temporary file definitions by means of the DEFINE statement. The SET statement, when used in the first group, will modify the contents of the work field each time an input record is read. The SET statement can also be used at several other points in the request (e.g., after the CRITERIA statement).

The second set specifies the criteria for selecting records. Only one set of criteria can be specified for a request, so if multiple functions are coded, they must all operate on the same set of records. However, the individual functions are flexible enough to accommodate different types of records if the number of types and occurrences is low. Such devices as outputting a blank record when the selected record is not appropriate could be employed, but the programming would be complex and beyond the scope of nonprogrammers (and some programmers as well). For more extensive situations, multiple Requests can be used, which produce separate reports.

The other four card sets correspond to the four basic Extracto functions: Tabulated Reports, Multi-Line Reports, File Creation, and File Updating.

The standard input routine included with Extracto will accommodate a data file on cards, magnetic tape, or disk. If on disk, the file can be organized in sequential, indexed sequential, or direct (random) fashiön. Blocked files on tape or disk can be handled. For reading files with a different organization, or for reading multiple files, a user exit is furnished to interface the user's own routine, or the SUPERIO option is provided.

(one for each function) and sorting, such a basic system would probably handle about 6 to 12 requests simultaneously. To handle the maximum of 250, a main memory in the neighborhood of 80K to 120K would be needed, and that doesn't include the operating system.

Complex file organizations are not difficult for Extracto to handle as far as file definition and record selection are concerned. In the handling of report specifications, there is one major criticism. Multiply and divide operations are provided, but they do not include automatic decimal alignment (scaling). Computations involving addition and subtraction or multiplication and division are simple to scale. But combinations of the two classes present complicated scaling problems. Programmers have generally forgotten how to scale since the passing of assembly-language programming, and the concept will be totally foreign to most nonprogrammers.

In summary, the strength of Extracto lies in its ability to produce many reports, both in terms of quantity and types, from data files maintained by other means. File maintenance and more complex input operations involving several files will require some input-routine coding on your part, or the use of the SUPERIO option.

USER REACTION

Datapro conducted telephone interviews with five users of Extracto. These users included two large municipalities (one in Canada, the other in the U.S.) and a noted university. They rated the system as follows:

	Excellent	Good	Fair	Poor	WA*
Overall satisfaction	3	1	1	0	3.4
Throughput/efficiency	2	3	0	0	3.4
Ease of installation	2	3	0	0	3.4
Documentation	1	0	4	0	2.4
Vendor support	2	2	0	1	3.0
Training	1	2	2	0	2.8

*Weighted Average on a scale of 4.0 for Excellent.

The five users had the following computer configurations: IBM 370/145–3, IBM 370/158–1, and UNIVAC Series 70–1.

The system performed as advertised immediately for three of the users, and eventually for the other two. None of the five users made any major modifications to the system.

Generally, the users found Extracto a fine system for producing management reports quickly and with ease. However, they caution prospective users to initially screen the coding written by nonprogrammers. Some went as far as to retain control of the systems in their EDP departments. Aquila puts out periodic newsletters on Extracto, wherein can be found considerable user feedback and fixes for problems and complaints. □

An important product of an Extracto run is an output listing of the Requests. An extensive set of diagnostics identifies errors in format or syntax. If at least one Request passes the checks, the run will continue.

A separate work file is maintained for each function. After processing is completed, the work files are sorted to separate the results for each Request and to order the elements. The output can be printed or spooled to tape or disk for off-line or later printing. The exact facilities are a function of the operating system.

FILE DEFINITION: File characteristics are identified in the manner normal with the particular operating system involved. In addition, the fields of the record must be defined in parameter cards input to Extracto. To define a field, the user must identify a name to be used in all Requests using this file definition to access the file, the relative location in the record, the length, the data format, an edit mask, and the number of print positions required. The last two items are optional and, if supplied, can be overridden in the two report functions. Only the fields of interest need be defined. The areas within the record can be redefined as desired to simplify references or to permit handling of multiple record types.

Data formats supported include alphanumeric (250 bytes maximum), binary (4 bytes maximum), zoned decimal (15 bytes maximum), and packed decimal (8 bytes maximum).

A three-digit code is used to identify the editing action to take place when a numeric field is printed. The units position identifies the number of decimal places. The tens position identifies one of ten edit-mask classes, which are combinations of seven different actions, including zero suppression, comma insertion, floating dollar or minus sign, and trailing percent, minus sign, or credit (CR) sign. One edit mask specifies no editing at all. The first digit of the edit code identifies the action to be taken for a zero value or indicates that the edit mask is a non-standard one. Zero values can be blanked or can be printed as zero with appropriate characters inserted.

The preceding paragraph describes a total of 200 edit masks (numbers 000 through 199). Six nonstandard marks in the 200 series are reserved for printing dates, telephone numbers, and social security numbers (Canadian or U.S.). In addition, the user can easily create and assign numbers to any conveivable edit masks.

RECORD SELECTION: As the file is passed, each record is tested against the record selection criteria for each Request. If the record passes the test, the statements for that Request are processed. Any field identified in the file definition for the file can be used in the criteria. In general, a series of conditions can be linked by AND and OR conjunctions. Complex logical relationships can be constructed by the use of parantheses. Each condition can be a comparison between an alphanumeric variable and another alphanumeric variable, work field, or literal, or between a numeric variable or mathematical expression (which contains numeric variables and literals) and a numeric variable, mathematical expression, or literal. Comparisons can be for equal, not equal, less than, greater than, not less than, and not greater than. In mathematical expressions, addition, subtraction, multiplication, and division are permitted, but decimal alignment (scaling) must be done by the user.

TABULATED REPORTS: This type of report generates one printed line for each selected record. Optionally, a summary report with only subtotals can be generated. Seven different statements are available: REPORT, TITLE, HEADINGS, FOOTINGS, CONTROL, MATH-TOT, and TOTAL.

The REPORT statement completely defines the content of each column, which can be an alphanumeric or numeric variable, work field, or literal or the results of a mathematical expression. Horizontal spacing of the columns is achieved by SPACE qualifiers. There is no limit to the number of columns that can be printed as long as the line width does not exceed 130 characters. However, only 32 columns can be totaled.

If an invalid operation, such as division by zero, invalid data, or counter overflow, occurs, the result is replaced by a flag indicating the problem; any totals involving such elements are also flagged in the same manner unless specifically inhibited. Options, such as editing and specification of totals, as well as conditional expressions, are contained in parantheses following the element (variable name, literal, or math expression). Redefinition of a columnar element is allowable through an OR operator to give flexibility in report output.

The TITLE statement permits coding up to 16 lines of 65 characters each, which will be automatically centered at the top of each printed page. The HEADINGS and FOOTINGS statements allow up to 16 full-width lines to be printed at the top and/or bottom of each page, respectively.

The CONTROL statement is used to specify certain options such as the number of subtotal levels (up to 9 levels, which correspond to the 9 leftmost columns), lines per page, single or double spacing, and whether or not detail lines or only subtotals and totals shall be printed.

The MATH-TOT statement allows the coding of arithmetical relationships among the subtotals for various columns. This is a particularly useful facility for computing averages. The order in which the columns are laid out is of critical importance, because the detail lines are sorted before printing according to the absolute values of the columns, working from left to right in ascending order. Normally, this corresponds to the natural way of doing things, but in some cases, such as trying to group several sets of information into the same report, the finished report might look peculiar. This sorting order can be modified by altering the parameter cards for the sort routine, but this would probably necessitate a separate run for the Request because it would affect all Requests in the run, and that would probably be undesirable.

The TOTAL statement is used to print explanatory literals on subtotal and final total lines. There can be as many TOTAL statements as there are subtotals (the TOTAL statements are numbered). A table look-up facility is standard through the EXPERT print exit routine.

MULTI-LINE REPORTS: In this type of report, the output resulting from a single selected record can occupy up to 99 lines. Preprinted forms usually dictate the use of this function. Two types of statements are used to specify the body of a Multi-Line Report. The SEQUENCE statement specifies the order in which the report will be printed. LINE statements specify the contents of each line. Essentially the same facilities are available to the user for generating entries as in a Tabulated Report, but no totals can be accumulated. Titles and headings can be generated just as for a Tabulated Report. The CONTROL statement can be used to print elements side by side if desired, a very useful facility for small forms.

FILE CREATION: Based on the records selected, data can be extracted and manipulated to generate a new file. Data format and length can be changed and fields can be rearranged, but no editing is permitted. Arithmetic computation and literals can be used to generate fields for the new file. Output can be blocked or unblocked.

FILE UPDATING: Arithmetic expressions involving the existing data fields and literals can be used to modify the fields of the input file. Normally, in the case of a tape or card file, the whole file is rewritten. For files contained on direct-access devices, only the affected record can be rewritten. There is no standard provision for adding or deleting records, but an exit is provided so that the user can code his own routines for this. The principal use of the File Update function is to apply a uniform change to a selected set of records; e.g., all employees with a certain job title get a 10 percent increase in pay.

PERFORMANCE: This aspect of Extracto is difficult to put into perspective. Case histories show some enormous savings (up to 95 percent), but these savings were achieved principally because multiple reports can be prepared in one pass of the file.

The vendor makes a general statement that the Extracto program will run as fast as or faster than a specially tailored COBOL program.

The timing elements of an Extracto run can be thought of as three phases: validation of requests and record selection, sort (if required), and print. Timing is further complicated if several programs are running in multiprogramming fashion. In general, the more requests you can process in a single run, the better the performance will be.

HARDWARE/SOFTWARE REQUIREMENTS: Versions of Extracto are available that will run on a 24K Honeywell Series 200 (Model 115 or larger with Advanced Programming and Edit Instruction) under Mod 1, Mod 1 MSR, Mod 4, or OS 200; a 32K IBM System/360 or 370 (Model 115 and up) under DOS, OS, or their VS counterparts (and the VSAM access method); a 32K UNIVAC Series 70 (Model 70/35 or larger with decimal instructions) under DOS, TDOS, or TSOS; a UNIVAC 9400 or 9480 under OS/4; a UNIVAC 9700 under OS/4 or OS/7; a UNIVAC 1100 Series system under EXEC 8; a Siemens 4004 operating under DOS or BS 1000; and a Unidata 7.7 Series operating under BS 1000.

In addition to a card reader and printer, sufficient tape or disk drives are required to support sorting and provide one work area (tape drive or disk area) for each of the four Extracto functions implemented.

The basic main memory required by Extracto is 22K characters (Honeywell 200) or 24K bytes (IBM System/360 and UNIVAC Series 70). Within this partition, approximately 400 elements can be accommodated. An element is a variable definition, conditional, report column or field, or literal. Each additional 1000-byte area of memory available to Extracto accommodates about 80 additional elements.

PRICING: The current version of Extracto (Version 3), which includes table look-up (and dynamic DFT for DOS systems) to facilitate file independence by means of a parameter card, costs $15,000 for a permanent license. A subset, Version 2, is offered for $10,000. The purchase price of the system includes installation, training, and one year's technical support. After the initial year, technical support costs $500 per year for Version 3.

A lease/purchase plan is available with 26 monthly payments, each of 5 percent of the purchase price. Multiple installations are discounted as follows: each of the next five system installations within a company costs 40 percent of the first installation's price; additional installations after that cost only 25 percent of the initial installation's price.

The SUPERIO multi-file handling routine costs $2,500, and the indexing option costs $5,000.

Extracto is available in Western Europe through VOLMAC BV at its Rotterdam office. Complete documentation for Extracto is available in French (the program was developed in French, and Extracto coding sheets and parameter forms are bilingual in English/French), but the package is not being actively marketed in France. However, French-speaking Swiss and residents of the German region near Alsace may be interested in the French documentation.

INITIAL DELIVERY: The first commercial installation of Extracto took place in Canada in June 1969. U.S. marketing started early in 1971.

CURRENT USERS: As of February 1975 there were about 150 active Extracto users (updated user lists are always available in Extracto newsletters). About 60 percent of the current users are Canadian firms; all the rest, except for a few in Europe, are U.S. installations. ■

Payroll/Personnel System
Argonaut Information Systems, Inc.

MANAGEMENT SUMMARY

Any payroll/personnel package must solve these problems: the system must be generalized in its objectives yet capable of satisfying individual requirements; provision must be made for producing standardized output while accepting varying types of input; and the variables of tax calculations, work rates, and different union contracts must be current and accurate, yet responsive to modifications by a payroll clerk. While solving these problems, the system must not burden the user with clerical detail and must allow him to easily update and maintain the total system. It is one application that must be run on schedule, and it certainly doesn't lack for auditors.

The Argonaut Payroll/Personnel System is generalized and adaptable to specific user requirements. This is accomplished by the use of a Control File through which the user provides his variable information. Supplementing the Control File are system constraints that have been left open. For example, no limits have been placed on the sizes of the control fields. This means that employee number can be three positions or ten, that division code can be one position and department two, or department five and division three, etc.

Initially, only one input form is required (an employee status notice), which provides the data needed to create the payroll and personnel master records and to effect quarter and year-to-date adjustments. A second input form is used, when no time card or similar document is available, for entering hours or rate information. By these means, the varied user inputs are established on the system's files, and the system can then produce the "standard" payroll output.

The Payroll/Personnel System has several noteworthy features which address the constantly changing external environment surrounding payroll processing.

First, the system provides a tax calculation module (TAXBREAK) which contains routines to calculate federal, state, city, and/or county witholding taxes, FICA, and state disability insurance. All 50 states plus Washington, D.C. and Puerto Rico are included. All of the users Datapro contacted cited this as the finest feature of the system. The updating of this relocatable module is extremely simple. Maintenance service for TAXBREAK is available on a per-year basis, with all changes being immediately forwarded to the user.

Secondly, the problem of varying work rates and diverse union contracts is handled by the system's Union-Matrix Option. The Union Matrix handles not only trade unions, but also other complex payrolls, such as piecemeal rates or multilevel commissions. The matrix is actually a file of records residing on disk storage. One matrix corresponds to (usually) one union. Within each matrix, the rows generally correspond to positions or job classifications, and the columns generally correspond to pay categories. The naming of rows and columns is entirely arbitrary within a given union. There are three subfields within each column which specify first, second, and third shift rates.

This system handles standard payroll processing and personnel record-keeping. It can also process the complex types of information required by a number of diverse payrolls with a multiplicity of contractual pay and benefit requirements. It runs on IBM System/360 and 370 computers under OS or DOS, and on UNIVAC 70 and 90 Series systems under DOS.

CHARACTERISTICS

SUPPLIER: Argonaut Information Systems, Inc., 2140 Shattuck Avenue, Suite 203, Berkeley, California 94704. Telephone (415) 845-7991.

BASIC FUNCTION: The Payroll/Personnel System is designed to automate the clerical procedures necessary to calculate the salaries of hourly-nonexempt (overtime paid), salaried-exempt (no overtime paid), salaried-nonexempt (overtime paid), administrative, and executive employees. It handles the standard federal, state, and city withholding taxes. There are provisions for 20 standard (fixed), 20 variable (one-time), and 3 cumulative (e.g., bonds) deductions. Processing frequency may be weekly, bi-weekly, semi-monthly, and monthly. The Personnel Master File contains such information as: salary change date, next review date, merit ratings, and a user data field (30 bytes minimum). All programs are coded in ANSI COBOL or COBOL D.

OPERATION: Processing of the Payroll/Personnel System centers around master payroll and employee files. There is also a Control File through which a user provides certain variable information to define his particular requirements. This not only allows the average user to tailor the system to his needs without extensive reprogramming, but it also provides a means by which the user with multi-company requirements, such as a service bureau or corporate data center, can process diverse applications with the same system.

In a standard payroll run, the following tasks are performed: 1) load and validate transactions, 2) update Payroll Master File, 3) compute payroll, and 4) extract data for checks and reports and print.

The system requires one input form, which contains 12 lines corresponding to the data needed to make up the payroll and personnel master records and to effect quarter- and year-to-date adjustments. The bulk of this information is written only once to initially establish the master files. Changes to the master files are accomplished by filling out only that information which is to be changed.

A second input form is used, when no time card or similar document is available, to enter hour and rate information. The system includes a provision for standard hours and/or standard pay which gives the option of paying automatically; that is, without having to enter any hours and rate information. This feature also allows "exception" inputting, in which only non-standard information need be entered. Any combination of the above may be employed, from the automatic (no-input) processing of executive or administrative employees to the processing of multi-union hourly or commission/salary employees.

Once the master files have been created, the control file loaded, and the transactions validated, the backup Payroll/Personnel Master Files are copied. Audit and edit listings are produced prior to file copying for verification and re-entry of rejected transactions. Then the Payroll/Personnel Master Files are updated with the transactions

▷ These rates may be either the actual rates per hour, or they may represent "add-on" amounts; this option is controlled by codes within the matrix.

Finally, the Personnel Master File is primarily a data base used for personnel reporting, which can be tied into a user's existing personnel system.

USER REACTION

Datapro interviewed eight users of the Argonaut Payroll/ Personnel System. Their ratings were as follows:

	Excellent	Good	Fair	Poor	WA*
Overall satisfaction	4	3	0	1	3.3
Throughput/efficiency	0	5	1	2	2.4
Ease of installation	4	1	0	2	3.0
Documentation	3	4	1	0	3.3
Vendor support	5	3	0	0	3.6
Training	0	7	0	1	2.8

*Weighted Average on a scale of 4.0 for Excellent.

These users represented the following computer and operating system configurations: IBM 360/30–3, IBM 360/40–1, IBM 360/50–1, IBM 370/115–1, UNIVAC 70/45–1, and UNIVAC 90/60–1; operating under IBM DOS–4, IBM DOS/VS–1, IBM OS/MVT–1, and UNIVAC DOS–2.

The system performed as advertised immediately for six users, but never for one, while the eighth user inherited the system under a facilities management contract and did not respond to this question or rate the system's ease of installation. Modifications to the system were equally divided between the user and the vendor. The precoded modules and the quality of the code, commented two users, made their modifications extremely simple. However, one user commented that though he did not use the Personnel File he still had to run it, thus increasing his processing time. He also felt the maintenance of an operational control card file was cumbersome. Another user was experiencing difficulty in "getting needed information from the package".

Two users were using the Union-Matrix Option and reported no problems with it. A user processing the payroll for 1400 employees and 6 divisions weekly found "user support by vendor excellent, modification costs reasonable, and the Union-Matrix and tax modules superb." The Personnel segment of the system was praised by one user who wrote special print runs; one of these handled all of his Equal Opportunity Employment reporting.

In summary, the users found the Payroll/Personnel System to be responsive to their requirements, generalized enough to meet standard reporting needs, capable of handling changing information, and, above all, designed to be used effectively by a payroll department. □

➤ **that have been entered. The New Payroll/Personnel Master records are sorted sequentially and go to a Copy Updated Masters to Backup run. The Control and Edited Records Files are now used to update the Quarter and Year-To-Date adjustments. Employee Status Forms are then printed.**

The updating processing cycle is completed by the loading of a rate table, entry and validation of rate type cards, and the creation of a Records Edited File. The Records Edited File, Control File, and Rate Table File now enter a Rate Lookup routine. The balancing of these rate transactions is the final step before the calculation of gross to net pay.

The Control File, Payroll Master, and Edited Records File are the inputs to the Calculation Gross to Net run. This run calculates gross pay, taxes, and net pay, and updates the Quarter-to-Date and Year-to-Date amounts for each employee. Once this data has been recorded and extracted, a Report File is produced.

The Report and Control Files become the input to the last payroll processing task: Print Checks and Registers. The output reports are:

- **Payroll Checks.**
- **Detail Payroll Register–details total earnings of all types for each employee plus taxes withheld and total deductions taken.**
- **Payroll Deduction Register–details all deductions for each employee.**
- **Hours (History Report)–details each type of hours worked by each employee for the payroll period; shows year-to-date vacation and sick leave accrued and taken.**
- **Deduction Recap Listing–recaps deductions taken and shows total amounts deducted from all employees for the current period and year-to-date; may be run for all deductions or for selected deductions only.**
- **QTD/YTD Register–details all types of earnings and taxes withheld for each employee on a quarter-to-date and year-to-date basis.**
- **Taxable Wage Accrual Recap–recaps taxable wages for each employee up to variable limits selected by the user; reports totals of wages which are subject to taxes in various states as of the current payroll period.**

In addition, several "as needed" reports are produced: 1) a Labor Distribution Report, which details hours and earnings by labor account within department (or other control); 2) an Employee Information Survey, which details selected information from the payroll and personnel masters; 3) W-2 Forms (usually run annually); and 4) 941-A Form (usually run quarterly).

HARDWARE/SOFTWARE REQUIREMENTS: The system runs on IBM System/360 or 370 computers under DOS, OS, or their VS counterparts, and on UNIVAC 70 or 90 Series systems under DOS. It requires a minimum of an IBM System/360 Model 25 with 48K bytes of memory including a 12K supervisor, two 2311 disk drives (one for processing plus one for system residence), a tape drive (or another processing disk), card reader, 132-position printer, and console typewriter. Exact configurations required for given payroll applications may vary according to user specifications, volume, etc.

All the programs are written entirely in COBOL and are available in ANSI COBOL for OS systems and in either ANSI COBOL or COBOL D for DOS systems. All programs are modular in design, and full, descriptive data names are used throughout. Standard library routines are used in each program wherever possible, and the programs are easy to read and maintain.

PRICING: The base price of the system is $4,800. This price includes a complete set of the source code; the JCL; a documentation listing that includes program narratives, record layouts, clerical instructions, and definitions of terms; a copy of the tape file from which the documentation is listed; sample input forms; a user's manual for the tax routines; instructions for installation of the system; and maintenance service on the tax routines for one year. Installation services are available at an additional cost of $2,500.

INITIAL INSTALLATION: June 1969.

CURRENT USERS: 17 as of February 1975. ■

Project Control/70
Atlantic Software, Inc.

MANAGEMENT SUMMARY

The traditional use of computers has been largely in process-oriented environments. That is, the computer is employed to perform a repetitive task that has been described beforehand. Describing the process to perform the task is called programming, and it's a project (non-repetitive), rather than a process. Additionally, many non-EDP related fields have a great deal of project-oriented work that it would be nice to have performed with the aid of a computer.

Project Control/70 is a package designed to fulfill that wish. Growing out of the need corporate management often felt to take the mystery out of the EDP department's planning and treat the EDP staff as any other staff function, it can also be applied to project-oriented work in general. Project Control/70 is logically divided into four modules that correspond to the four requisities of effective project control: planning, monitoring, accounting, and evaluation.

Planning steps include defining a project, making estimates, setting target dates, assigning tasks, and breaking the project into task and cost units. To do this, one must know the makeup of the tasks, the skills or other resources necessary, the availability of skills and resources, impacts upon other projects when the skill, personnel, and resource pool is tapped, and so on. Often, management has a demand target date for project completion, and this necessarily raises the questions of "is it possible?" and "how will this impact other work in progress?" Project Control/70 has a Planning Module that lets the user make an estimate to "try out" a project. It will compute dates and costs, and show the effect on other work.

Project monitoring means comparing actual progress versus planned progress with respect to elapsed time, costs, and dates. The Monitoring Module of Project Control/70 presents status reports by project and personnel, and by skill and individual. It also generates exception reports to give late signals, variances from plan, and overbudget messages. These reports, which can provide a monitoring overview of a project, both by the project and by the resources employed, are prepared both at detail and summary levels.

Project accounting is the third necessary part of good project administration, and Project Control/70's Accounting Module can handle projects and activities within time periods and present statistics for evaluation and use in subsequent new planning.

Finally, Project Control/70 has an Evaluation Module to present reports that facilitate this vital step in project management. The reports are able to draw from information in Project Control/70's data base, maintained in simple fixed-length sequential files.

▷

Project Control/70 aids management in planning, monitoring, analyzing, and accounting functions on project-oriented activities. It can be used on any computer system that can support ANS COBOL.

CHARACTERISTICS

SUPPLIER: Atlantic Software, Inc., Lafayette Building, Suite 910, 5th & Chestnut Sts., Philadelphia, Pennsylvania 19106. Telephone (215) 922-7500.

BASIC FUNCTION: Automated system for the control of data processing and other types of projects. Project Control/70 helps management measure project costs, monitor project status, analyze staff capacity, and forecast delivery and costs. It is based on a timesheet and simple initilization forms, and can operate on virtually any ANS COBOL computer system.

OPERATION: Operation of a Project Control/70 system begins with a tailored installation that involves such basics as selection of a project numbering scheme, cost rating schemes, and scheduling algorithms. Atlantic Software provides assistance in these areas.

Project Control/70's highlights include:

- Flexibility – it can support virtually any project, resource, and task format, without limit as to the number or types of tasks or resources. The system has a complete priority facility, supports custom-tailored descriptions, cost rating schemes, and scheduling algorithms, and can process on any required time cycle, both for input and reports. PL/1 programming projects can also be handled.

- Minimal user input – task records can be initiated from a timesheet; only one card is required to reassign resources, delay or improve a project and its parts, or delay or improve a resource's schedule. Users can input priorities at task, program, and system levels, and the system has a complete comments facility.

- Data base structure – only one master file is used, and its format is fixed-length sequential. Consequently, the system is typically output-bound in performance.

- Outputs – a large number of reports can be produced, suiting user needs at various levels. These are, for example: turnaround time documents, individual performance records, analysis of completed programs, completed projects analysis, a project scheduling barchart, project planning report, manpower availability report, resource planning report, resource status report, active projects due report, over budget report, project status report, and an executive summary report. Not all of these reports must be produced. Any can be produced on a demand or periodic basis.

- Cost – while close to competing systems in cost, Project Control/70 contains no hidden charges. The license fee is a one-time charge, and maintenance is

Project Control/70
Atlantic Software, Inc.

▷ Project Control/70 is one of the more successful packages of its type, with over 100 current users. It was initially released (Version 1/Release 1) in the autumn of 1971. At this writing, Project Control/70 is in Version 2/Release 8, and Version 3/Release 1 is projected to arrive in mid-1974. Atlantic Software sends out new releases quarterly, and creates a new version about every 18 months. Existing customers receive new releases free and can obtain new versions at a discount.

Atlantic Software is a reputable vendor, known to many for its association with Score. Score is an information retrieval package that is currently marketed mainly by Programming Methods, Inc., a GTE Information Systems subsidiary. The package was named to Datapro's 1973 Software Honor Roll, and is covered in Report 70E-457-02. Many Score customers obtained the package from Atlantic Software, whose offices overlook Independence Hall.

USER REACTION

Atlantic Software has a reputation for top-notch service and competence. Project Control/70 users report that Atlantic provides education at installation time that is good enough to charge for, but it's free. Atlantic also offers a color videotape cassette educational package for users with monitors.

In operation, Project Control/70 is reputed to be nearly error-free. Users report that it is generally installed without problems and that questions are handled correctly and promptly by the vendor.

Users also report that it is easy to pyramid reports from only a project number and time estimates; that the reports are useful, seeming almost to be designed by users and making sense to programmers; that only one card can change assignments, make adjustments, etc.; that the reports provide objectivity to project control; and that the system is flexible and can have wide applications. A DOS user states that Project Control/70 has satisfied his charge-back requirements completely, and that its run times are "realistic."

An OS user, who reports run times per phase of 30 to 40 minutes on a 370/145, says that the system was installed in about 18 hours of wall-clock time and is used as the complete basis for a charge-back, accounting, and reporting system. Another user, this one under DOS, runs Project Control/70 in only a 42K-byte partition and has interfaced the package to GRASP's Job Accounting module.

Users note that Project Control/70 is individually tailored, and that makes the few criticisms about it extremely subjective. Naturally, large projects (involving many people or resources and long time-frames) can become unwieldy in page volume, but Project Control/70 lets the user select as few or as many reports as he wants, as often as he wants.

In summary, Datapro feels that organizations that need more effective project control for small and medium-sized efforts should seriously consider Project Control/70. □

➤ **free during the initial year of use, and optional at reasonable costs thereafter.**

- **Applications – Project Control/70 has been successfully implemented to control various types of EDP and non-EDP projects.**

Project Control/70 has the important ability of being able to handle any number of types of tasks and/or resources. Its planning facilities can report or project on the basis of tasks or resources, which lets the user look at things pretty much the way he wishes to.

PERFORMANCE: Benchmarked on a System/370 Model 155 under OS/MVT, with 150 resources reporting, the weekly CPU time was 10 minutes and the elapsed time was 30 minutes. Naturally, the performance of a project control system is most meaningfully discussed in terms of the facilities provided, ease of installation, and whether the operation is easy and error-free. Project Control/70 rates highly with users in these respects.

HARDWARE/SOFTWARE REQUIREMENTS: Project Control/70 is written in ANS COBOL, and can run under most operating systems and on most computers that will support ANS COBOL. Atlantic Software provides the ANS COBOL source code. The system has been used on IBM System/360 and 370 DOS, OS, and virtual storage computers; Honeywell 3200 computers; Burroughs B 3500 and B 6700 computers; and UNIVAC 1106, 1108, and Series 70 computers.

PRICING: Project Control/70 is licensed for a 1-time charge of $9,500. This sum includes installation services, basic education, maintenance, and new releases (quarterly). New versions are available at discount rates. Extensive training and documentation are provided by Atlantic Software.

INITIAL DELIVERY: 3rd quarter 1971 (Version 1/Release 1). Project Control/70 is presently in Version 2/Release 8, and Version 3/Release 1 will appear in mid-1974.

CURRENT USERS: Over 100 as of December 1, 1973. ■

Customer Service Payroll System
Automated Financial Systems, Inc.

MANAGEMENT SUMMARY

The Payroll System is a general-purpose, COBOL-coded package for IBM System/360 or 370 computers operating under DOS. The system offers facilities for processing salaried, hourly, daily, and piecework payrolls. It handles varying pay-period cycles, produces a full array of required payroll reports, and includes a labor distribution reporting function.

The Customer Service Payroll System, which was initially developed for use in a bank environment, is marketed as a general-purpose payroll system. It is a basic payroll system which, on the surface, seems to be limited in its capabilities for handling complex payrolls. However, the package has a substantial number of users, not a few of which are reasonably large companies with requirements for more than basic payroll capability. AFS states that about the only type of payroll that they have not been able to accommodate with this package is one for a railroad. Expansion beyond the basic capabilities is achieved through additional programming and through use of the override features that permit submission of precalculated information to be included in the system just as if the system itself had calculated it.

The Payroll System is one of three Levin-Townsend packages acquired by Automated Financial Systems at its founding in May 1970.

The package consists of 26 programs, including a sort program, that operate under DOS on an IBM 360/25 or larger system with the specified minimum equipment configuration and a minimum of 32K bytes of core memory. Modification of the basic package to meet individual user requirements should be comparatively easy to accomplish at a fairly low cost because the package is written completely in COBOL. Consideration, however, must be given to the extent of the changes required to meet the special needs of a particular installation.

In some areas of payroll calculation, such as additional compensation and special overtime compensation, the system is restricted to utilizing a single calculation technique. However, the system provides an override feature that allows temporary changes to file record information and the entry of precalculated data. By overriding both established factors and normal calculation procedures during payroll processing, alternative methods of computation can be handled, but they must be performed outside of the Payroll System itself.

Designed to process up to five state and local taxes per employee, the Customer Service Payroll System can be expanded by user-coded subroutines or through an interface to Management Information Systems' ALLTAX package.

The Payroll System provides for production of up to 26 reports, including all required Federal reports, and utilizes standard stock paper for all printouts except the Payroll Register, Check/Deposit Slips, Accumulated History Register, and Deduction Register. The user is provided with samples of each of these required pre-printed forms. ▷

CHARACTERISTICS

SUPPLIER: Automated Financial Systems, Inc., One Decker Square, Bala Cynwyd, Pa. 19004. Telephone (215) 667-1000.

BASIC FUNCTION: The Customer Service Payroll System is a general-purpose payroll package that provides the capability to process a variety of different payroll bases and payroll-period cycles. It produces a full complement of payroll reports and incorporates a labor distribution capability.

OPERATION: The system uses the exception reporting mode of payroll production, providing the capability to override payroll factors which are pre-established in employee records. It also allows entry of precalculated data to replace regular system calculations before and during actual payroll production.

The system provides payroll control based upon a 13-character control number consisting of a 2-digit company number, 6-character department number, and 5-character employee number. Production of separate company or division payrolls can be accomplished through the use of this numbering system.

Utilizing control codes in the employee master record, payrolls can be produced for weekly, bi-weekly, semimonthly, and monthly pay cycles for salaried, hourly, daily, and piecework employees. Various specifications exist for acceptable methods of calculation of overtime earnings, rates of pay, special overtime, additional compensation payment, and sick, vacation, and holiday pay. Maximum values which can be accommodated for individual employees are usually set at $99,999.99. Hourly workers who get a guaranteed minimum pay regardless of the number of hours worked can also be accommodated.

Four standard methods of calculating Federal income tax and the standard method of calculating Social Security tax are provided. Subroutines to calculate other taxes for each employee can be included. However, only a single tax computation subroutine, utilizing a flat percentage of total earnings, is provided with the system.

A maximum of 12 salary deductions from total earnings can be accommodated. The user specifies the type of deduction represented by each code number as a standard code for all employees. Eight categories of deductions can be specified as representing either a flat amount or a percentage of total earnings. Once specified, the method ►

Customer Service Payroll System
Automated Financial Systems, Inc.

In the design of this package, emphasis has been placed on capability and flexibility in certain areas, with inevitable limitations resulting in other areas. The Labor Distribution feature, the number of deductions which can be processed, the minimum equipment configuration required, and the moderate price are important considerations for the potential user. □

used with that deduction applies to all employees of that company. Savings bond deductions utilize a standard subroutine.

Four deduction codes provide a floor and ceiling specification for setting minimum year-to-date earnings before deductions begin and maximum earnings after which deductions terminate.

All 12 deductions are processed in numerical sequence, with a report of deductions not taken produced by the system when employee earnings are insufficient to cover deductions.

The Labor Distribution feature allows total earnings to be distributed to 10 departments or divisions. Percentage of earnings chargeable to each department can be fixed in the employee record or changed each payroll period with the override facility. Multiple allocations of earnings by functions as well as departments can be accomplished, limited only by the combined maximum of 10 chargeable codes.

The override feature incorporated into the system allows the codes and factors for guaranteed wage, earnings rate, total earnings, regular deductions, and labor distribution to be altered for a particular payroll run without permanently changing the employee record.

INPUT: The system accepts, in addition to regular file inputs, a variety of input data depending upon the type of payroll being processed and the extent to which established payroll calculation factors are to be overriden and precalculated data is to be entered during a particular cycle of payroll processing.

OUTPUT: The output consists of updated files reflecting such items as new year-to-date and quarter-to-date accumulations, plus the following series of documents and reports:

Transaction Edit Report
Payroll Register
Checks or Deposit Slips
Check Register
Accumulated History
Deduction Register
Employee Labor Distribution Report
Departmental Labor Distribution Recap Report
941-A Forms
W-2 Forms
Dates of Service Report
Bond Purchase Report
Deductions Not Taken Report
File Maintenance Reports (one or two reports for each of three master files)
File Listings (one or two reports for each of three master files)

PERFORMANCE: Execution of the complete system to produce the specified output naturally depends upon the number of employees and the extent of the changes entered during processing. Approximately two hours of computer time are required for complete processing of a 5,000-man payroll with related reports. Reduction in the number of personnel on the payroll will not produce proportionate reductions in processing time because of the fixed elements of system overhead.

HARDWARE REQUIREMENTS: 32K IBM System/360 or 370, Model 25 or larger, equipped with two 2311 Disk Drives (includes operating system residency), 2540 Card Read/Punch, 1403 Printer, and 1052 Console Typewriter.

SOFTWARE REQUIREMENTS: The system operates under DOS, and is written entirely in COBOL.

DOCUMENTATION: User's manual with operating instructions and console messages, plus narrative descriptions, system flowcharts, program flowcharts (for the calculating programs only), source program listings, input record and report layouts, and sample reports.

PRICING: The system is available on a purchase basis only, at a cost of $5,000. Right of resale is prohibited under terms of the purchase contract.

SUPPORT: AFS will provide three man-days of on-site customer support for training, installation, conversion, and production operation.

Additional support for modifications to meet special user requirements or extended installation support is available on either a per-diem basis or negotiated fixed-price contract.

AFS warrants the performance of the system in accordance with specifications and will provide error-correction maintenance without charge for 90 days.

INITIAL DELIVERY: Summer 1966.

CURRENT USERS: More than 50 as of September 1972. ■

BOSS
Automated Financial Systems, Inc.

MANAGEMENT SUMMARY

BOSS provides the capability to operate multiple application programs in a multiprogramming environment, thereby permitting the user to expand beyond a single teleprocessing application.

The BOSS system (formerly called the Teleprocessing Master Control Program 'or TP/MCP) was one of three Levin-Townsend packages acquired by Automated Financial Systems at its founding in May 1970.

Originally developed for use in a banking system, BOSS has been rewritten to provide a general-purpose system that can be used with any mix of application programs. The system is written in IBM System/360 Assembly language (BAL) and is designed to operate under DOS or OS. The program will support any number of terminals of any type, limited only by the user's computer configuration capability and by terminal hardware compatibility with the computer.

BOSS is modular in design, allowing a user to select only those modules necessary to meet his system requirements. It appears, however, that users will typically require most, if not all, of the modules for a properly controlled system.

Security clearance for access to an application program and related files is controlled by a BOSS analysis of terminal identification in relation to the specific application concerned. However, no method is incorporated in the system for identifying and clearing the individual using the specific terminal.

BOSS has the ability to maintain system statistics on errors, activity volume, error messages, average turnaround time, and application execution time. These statistics could prove valuable to the user in analyzing total system performance to identify faulty terminals or lines and to better structure his telecommunications applications.

Additional features of BOSS include: code conversion to and from EBCDIC, input message analysis, and storage of and access to multi-page displays. The latter feature permits segmentation of responses sent to a terminal; this is useful for terminals such as CRT's that have a limitation on message size.

Any IBM System/360 or 370 user with an on-line application expanding from a dedicated system should carefully consider BOSS.

Careful attention should be paid to the adequacy of the security checking function. ▷

BOSS is a modular interface program that allows a System/360 or 370 DOS or OS user to process multiple telecommunications application programs in a multiprogramming environment, with access from any terminal in the system to any problem program for which security clearance has been established.

CHARACTERISTICS

SUPPLIER: Automated Financial Systems, Inc., One Decker Square, Bala Cynwyd, Pa. 19004. Telephone (215) 667-1000.

BASIC FUNCTION: BOSS interfaces IBM System/360 DOS or OS teleprocessing facilities and application programs.

OPERATION: All modules of the Teleprocessing Interface segment and the Master Control Supervisor segment are assembled and linked by the user to produce the BOSS program.

The Teleprocessing Interface segment provides system functions, including line control, polling, message receipt and queueing, code conversion, error control, statistics accumulation, and buffer control.

The Master Control Supervisor segment provides the following major functions: on-line files control, input message function analysis, security control, application program selection and interface, and system recovery.

Communication from a terminal, consisting of a two-character function code (application identifier) and formatted input message text, is received, time-stamped, converted from line-code to EBCDIC, and placed in the input queue. The message is removed from the queue, security is checked, appropriate timers are set for the application, and the application program is selected. The required message information is then transmitted to the specified application program.

Following processing by the application program, control is returned to BOSS and any output message is placed in an output queue. At the appropriate time, the message is removed from the queue by BOSS, converted to line code, and a transmission request issued.

INPUT: The input to BOSS consists of a terminal-originated message directed to any one of the active application programs. Input messages are preceded by a two-character application program identifier. The message must be in a format expected by the particular application program to which it is directed.

OUTPUT: Primary output from BOSS consists of messages directed from an application program to the originating terminal, any other terminal, or any other output device.

Additional output consists of printouts of statistical accumulation on system operation, error recovery, and ►

BOSS
Automated Financial Systems, Inc.

▷ Possible alternatives within IBM-supplied software to the use of BOSS include manual switching between telecommunications programs based on time-of-day scheduling and batch off-line reception and processing of messages. □

➤ dumps to tape. All file access is through a re-entrant read/write module.

PERFORMANCE: The performance of BOSS will depend largely on the operating environment in which it is used. Factors such as teleprocessing configuration, number of modules utilized, number of application programs with which it must interface, and computer system configuration, together with channel interference and available memory, will all affect performance at a particular installation. No specific performance figures are available to date.

HARDWARE REQUIREMENTS: 65K IBM System/360 or 370, Model 30 or larger, two 2311 or 2314 disk drives, and a 1052 Console Typewriter. BOSS requires a minimum of 27K bytes of core memory, plus additional core depending upon the modules utilized, and space for the application programs interfacing with BOSS.

SOFTWARE REQUIREMENTS: The system will operate under IBM DOS or OS.

DOCUMENTATION: The purchaser receives a user's manual with forms, descriptions, file setup descriptions, and recovery procedures; and a system documentation package with functional specifications, system flowcharts, program narrative, file descriptions and layouts, and code descriptions. Program documentation is also provided, consisting of BOSS in source form on cards or tape compilation listings, logic flowcharts, operating instruction, console messages, and setup card layouts.

PRICING: The price of BOSS is $18,500 as a stand-alone package. When used with the AFS Commercial Loan Information System (Report 70E-069-03), BOSS is available for $10,000.

Modifications to BOSS can be obtained on either a per-diem cost basis or a negotiated fixed-price contract.

SUPPORT: AFS will provide the services of a senior systems analyst on-site, for training, installation, conversion, and production operation. The amount of time provided depends on the system; additional services can be obtained on a per-diem basis.

AFS warrants the performance of BOSS in accordance with specifications and will provide error-correction maintenance without charge for 90 days.

INITIAL DELIVERY: Summer 1969.

CURRENT USERS: 10 installations as of September 1972. ■

Commercial Loan Information System
Automated Financial Systems, Inc.

MANAGEMENT SUMMARY

The Commercial Loan Information System is designed as a total processing and reporting system for all of the commercial loan activities of a bank. Structured on a modular subsystem basis, the system permits the user to select those functions, activities, and reporting capabilities to be automated. Only the Basic Accounting Module is required; it can be used as a limited system by itself or as the nucleus of an expanded system.

This system was one of three Levin-Townsend packages acquired by Automated Financial Systems at its founding in May 1970.

The system modules operate under a Control File which serves as a functional system supervisor. All necessary information relating to available modules in the system, schedules, and regular information requests for a particular configuration of the system are stored by the Control File.

Expansion of the Loan System can be accomplished with minimal impact on an existing system by altering the contents of the Control File to reflect the availability of additional modules and changed processing requirements. These design characteristics of the Loan System allow time-phased installation or expansion of the system to meet changing requirements in commercial loan activity.

The system develops and maintains a detailed commercial loan data base with comprehensive cross-referencing of information entries. This data base serves as the source of information for the production of numerous regular and special reports. All output reports are automatically "spooled" to disk or tape for subsequent printing.

The system can operate in a batch mode on an IBM 360/30 with 65K bytes of memory. Implementation of an on-line system can be accomplished through the use of the On-Line Module, which requires an expanded equipment configuration consisting of an IBM 360/40 with 128K memory.

At present, a mortgage loan and an installment loan module are under development for use with the Commercial Loan Information System's Basic Accounting Module. These systems are planned for release by early 1973.

The modules are written almost completely in COBOL, with a few routines written in BAL.

▷

The Commercial Loan Information System, designed for operation under DOS or OS on an IBM System/360 or 370, is a package of interlocking subsystem modules that provide processing and reporting capability for the commercial loan activity of a bank. The modular nature of the package facilitates system expansion.

CHARACTERISTICS

SUPPLIER: Automated Financial Systems, Inc., One Decker Square, Bala Cynwyd, Pa. 19004. Telephone (215) 667-1000.

BASIC FUNCTION: To provide processing, management reporting, management information and portfolio control for the commercial loan activity of a bank. Major features of the system include a modular system structure and a choice of batch or on-line mode of operation.

OPERATION: The Loan System is an integrated package of subsystem modules. Each module, consisting of a number of programs, is designed to provide processing, data manipulation, file updating, data base updating, control reporting, and management reporting for its own functional area of the total commercial loan application.

STRUCTURE: Configured of specific modules to meet a user's requirements, the resultant system operates under the functional direction of its own Control File, which interfaces with the operating system. The Control File, which resides on disk, provides for inclusion and execution of programs within the system. All requests for information to be produced by the system on a regular basis are contained within the Control File and analyzed daily to schedule and execute the required combination of programs to produce the required information. In addition, the Control File analyzes all transactions entering the system and provides a schedule of programs required for processing.

The Control File maintains stored directories which are used to control and direct system execution and ensure security. Included among these directories is a listing and description of all data within the system. Utility routines, provided with the system, can be used in conjunction with the data directory to produce one-time reports.

The Control File also serves such functions as the maintenance of statistics on terminal usage and allocation of commercial loan activity by bank branch or subdivision.

The Basic Accounting Module, consisting of 54 programs, is the only module required in all configurations of the system. It can be used by itself in a basic system or as the nucleus in conjunction with any combination of other modules. It maintains and controls all financial and obligor information in the system. It consists of four major functional sections: Current Obligation Accounting which processes all loans outstanding and maintains liability ledgers and customer billing; Indirect Liability

►

Commercial Loan Information System
Automated Financial Systems, Inc.

▷ The comprehensive design of the Loan System, with respect to both processing functions and management reporting capabilities, warrants careful consideration of this package by most commercial banks. Its exceptional capabilities may well justify its rather stiff price. □

➤ Accounting, which determines total obligor exposure; Future Obligation Accounting, which controls and reports upon all future obligations, such as credit lines, commitments, and letters of credit; and Participations, which provides complete recording and reporting of all participation activities.

The Collateral Module, consisting of 16 programs, supports accounting, control, and reporting for loan collateral. It also provides collateral review reports, automatic security pricing, collateral substitutions, and field audit reports.

The Statistical History Module augments the detailed transaction information maintained by the Basic Accounting Module, with statistical accumulations of data by month for the most recent 24 months and by year for the previous six years for past outstanding loans. In addition, data is accumulated for cost analysis purposed, such as servicing statistics, allocation of expense, and profitability reporting. The reporting facility of this module allows data to be summarized at various levels and permits preparation of historical reports in graph as well as tabular format.

The Obligation Analysis module, consisting of 12 programs, provides the facilities for analysis of current loans and related lending patterns. Elements of this module provide for cash flow analysis with anticipated interest and principal as well as anticipated income based on maturity dates with various repayment plans. Anticipated revenues, determined by accruals, can also be forecast.

The On-Line Module is required for the on-line mode of operation. It provides the interface and control necessary for support of terminal input and inquiry and can be used in conjunction with other on-line bank applications. Routine transaction input, in approximately 50 formats, for transaction entries on new loans, interest and principal payments, etc., is supported, as well as transaction proof control and required loan calculations.

The Portfolio Analysis module is effectively a report generator. It uses the Control File directory of fields contained in all of the system files to produce specified reports. The module allows the selection of variable data fields with the report item sequences and degree of detail desired. Parameters can be stored in the Control File for regular reports as well as special reports.

INPUT: The selection of modules, the choice of off-line or on-line mode of operation, and the nature of the special reports and analyses determine the required inputs to the system. Basically, inputs consist of transaction information on the various phases of commercial loan activity, inquiries, information requests, and report generation parameters.

OUTPUT: The system outputs are contingent upon the configuration implemented by the user and the nature of the inquiries and management information reporting requirements. Approximately 22 standard reports are produced by the Basic Accounting Module, with up to 28 additional reports produced by the other available modules. Additionally, more than 50 terminal inquiry request formats are available for entry and retrieval of data.

PERFORMANCE: System performance naturally varies with equipment configuration, complement of system modules used, etc. AFS states that a complete system supporting 15,000 obligors with 25,000 obligations requires about two hours of processing time when operating on a daily accrual basis.

HARDWARE REQUIREMENTS: The system can be operated in batch mode on a 65K IBM 360/30 with three 2311 Disk Drives, three 2400 Tape Drives, card reader, and printer. Additional disk capacity is required for a system with more than 6,000 obligors on file. The on-line mode requires a 128K IBM 360/40 and can support any terminal device compatible with System/360 hardware. A version for use on the Burroughs B 3500 is also operational.

SOFTWARE REQUIREMENTS: Operates under System/360 or 370 DOS or OS.

DOCUMENTATION: A user's manual specifies input descriptions, input controls, and audit procedures. Five volumes of system documentation include specifications, narrative, flowcharts, report layouts, file descriptions, and input forms. Also provided are logic flowcharts, operating instructions, and the program decks or tapes.

PRICING: The system is available on a purchase basis only, without right of resale. Prices for the individual modules are as follows:

Basic Accounting Module	$30,000
Collateral Module	7,500
Statistical History Module	3,000
Portfolio Analysis Module	5,000
Obligation Analysis Module	3,000
On-Line Modules:	
Inquiry	4,000
Update	6,000

A version of the AFS BOSS system adapted for the On-Line Module is available for an additional $10,000.

An additional installation charge of $5,000 is also made for OS users, which includes 5 man-days of support.

SUPPORT: AFS will provide, free of charge, a senior systems analyst with extensive commercial loan experience on-site for 10 days. Services of the analyst may be apportioned to training, installation, conversion, and production operation as desired by the user. Additional support services are provided on a per-diem or contract basis. The system is warranted for 90 days following the delivery of the last module.

INITIAL DELIVERY: Initial versions of the system with certain modules were installed during 1966.

CURRENT USERS: There were 21 installations running at various stages of completeness as of September 1972. All of these have contracted for all modules except On-Line. Several installations have committed for the On-Line Module, but all except one have deferred its installation until all other modules are up and running. ■

QUERY3
Azrex, Inc.

MANAGEMENT SUMMARY

QUERY3 is an inquiry system designed to extract, summarize, reorganize, print, and/or copy information from tape or disk files. The package employs an English-like inquiry language that is both flexible and powerful in addition to its inherent simplicity.

QUERY3 can be used to search a file for information, to summarize data in a file, or to create and maintain files. In its basic version, it is restricted to batch-mode inquiries against a single sequential file. Optional features permit inquiries against ISAM (indexed sequential access method) files, on-line inquiry, and, recently, access from multiple files. A current restriction is that the records must be of fixed length. The basic package comes complete with a SORT verb and can sort a file conditionally. It also has a full-power COMPUTE verb.

QUERY3 is also notable for its relatively low cost and for the fact that it is highly computer-independent. To date, it has been installed on IBM System/360 and 370 computers, Model 30 and up, under DOS, OS, and VS; on CDC 3000 and 6000 Series computers under SCOPE, MASTER, and KRONOS; on Burroughs B3500 computers under MCP; on Honeywell 2000 Series computers under OS200; on UNIVAC (ex-RCA) Series 70 computers under DOS and TDOS; on NCR Century 200 computers under 7-B; and on DEC PDP-10 computers, where the basic system includes an interactive query facility. This computer-independence is due largely to the fact that QUERY3 is written in ANS COBOL. The package also contains user exits that permit it to be called from a COBOL program's linkage section and have even allowed sophisticated users to interface QUERY3 to IBM's IMS data base management system. (See Report 70E-491-01 for a description of IMS.)

After a file has been described once (as a dictionary) to QUERY3, the non-EDP user can make inquiries against the file without concern about, or knowledge of, the file's structure. He need not concern himself with the report format, either; QUERY3 has sensible defaults that control spacing, margins, and page breaking and numbering in the absence of detailed user specifications. Once the user has mastered the QUERY3 language (a simple process), he can specify production of customized reports of almost any type and format. Also, due to its simple query language syntax and COBOL source form, it can be adapted to resemble other languages. In fact, a Spanish version is currently available.

The genealogy of QUERY3 can be traced back to military command and control systems developed during ▷

This simple but powerful file inquiry system can be used on virtually any computer system with an ANS COBOL compiler. The basic system processes fixed-length records in sequential files. Options enable QUERY3 to handle ISAM files, multiple files, and on-line inquiries.

CHARACTERISTICS

SUPPLIER: Azrex, Inc., 215 Middlesex Turnpike Burlington, Massachusetts 01803. Telephone (617) 272-8750.

BASIC FUNCTION: QUERY3 is a user-oriented inquiry system featuring machine-independence. Its design includes a simple English-like inquiry language. It operates against a fixed-length sequential file (multiple-file input and disk ISAM file input are optional) in a batch or optional on-line mode. The files it operates against are defined only once to the QUERY3 dictionary, so that the user need not be familiar with file formats. Also predefined by extensive defaults are report formats, so that novice users need not concern themselves about report appearance, but only with their queries.

OPERATION: Before QUERY3 can be used to generate inquiries to any file, that file must be described to the QUERY3 dictionary. This is a simple process that is described in Azrex documentation and can be performed by the vendor for files to be used initially at installation. Thereafter, this one-time task is performed by the user for new files.

In processing a query against a file, QUERY3 operates in three distinct phases: sort the input file, analyze the inquiry, and generate the response report. These are indistinguishable to the user, who can merely query the system and receive his answer.

Once the user has learned the QUERY3 language well enough, he can employ it with remarkable versatility. QUERY3 can sort conditionally, produce up to 99 reports in one pass of the input file, compute, total, count, average, etc. It presents users with complete control facilities, as well as standard defaults, for page breaking, horizontal and vertical spacing, margins, titling, headings, and logical controls and selections. It can be used to generate mailing labels selectively, produce one-shot reports, or create special or updated files. It has a standard exit for attachment of the user's own coding to accomplish additional functions.

Perhaps the best concise description of the QUERY3 inquiry language is that it is quite similar to the Procedure Division syntax of COBOL. It also offers similar power and flexibility, complete with branches, arithmetic operations, logical operations, etc. A big difference is that QUERY3 inquiries can use straightforward PRINT statements, literals, and constants, and can create and use working storage.

PERFORMANCE: QUERY3 performance is highly dependent on the efficiency of the object system's ►

QUERY3
Azrex, Inc.

the early 1960's. Some of its forebearers were languages such as ADAM, AESOP, and COLINGO. Later, Azrex's president, David McIlhenny, perceived that commercial firms should have similar systems, and QUERY1 was developed in 1967 to interrogate personnel files for one company. QUERY2 was next; it enabled queries to any sequential fixed-length file but lacked the powerful verbs of QUERY3. The current product has moved through versions QUERY3.0 to the present QUERY3.3. QUERY3.4 is scheduled for November 1974 release and distribution to all customers with a maintenance agreement in force at the time. The primary author and developer of QUERY is Peter T. Ladd.

Azrex was founded in the fall of 1970, but its principal technical group has continuity in its field dating back some 14 years. The company currently employs 25 professionals. In addition to marketing QUERY3, Azrex is currently active in the fields of computer facilities management, scientific counsulting, and commercial data base system design and implementation.

QUERY3 is also marketed by Benefacts, Inc., 300 Joppa Road, Baltimore, Maryland 21204 (302-296-5500), but not on a stand-alone basis. Instead, it is sold as a query facility feature to that film's Career Inventory System package. Benefacts customers with QUERY3 deal only with Benefacts, however.

USER REACTION

Datapro interviewed QUERY3 users with various IBM hardware and some Honeywell equipment; we also spoke both to EDP- and non-EDP-oriented users. Users generally expressed high regard for the vendor's technical competence and service. The ways in which QUERY 3 was used differed considerably, showing just how the package can be applied in different situations. The package is in use to query personnel files, career inventories, payroll records, and donor records, for example.

A word of caution should be entered here: Due to its use of COBOL for machine-independence, the throughput of QUERY3 in response to requests can be no more efficient than the COBOL compiler of the object system. This can be poor at times; for example, a user has complained about poor throughput due to the inefficient SORT verb on the CDC 3300. The COMPUTE verb can also present problems if your critical criterion is timing or machine efficiency. One user expressed a wish that QUERY3 could work with variable-length records and states that he believes the vendor is working on the problem. Another user, slightly more resourceful, has solved the problem for himself using the standard user exits in QUERY3.

Datapro feels that QUERY3 can be a valuable addition to an environment where queries to a data base must be made by personnel not necessarily well acquainted with data processing methods and/or where machine independence in a query system or one-shot report generator is required.□

COBOL compiler. With this in mind, users praise QUERY3's performance. One cautions, however, against failure to maintain operations control that could allow repeated, or production use, of raw query code in cases where modification by a programmer could significantly increase its efficiency.

HARDWARE/SOFTWARE REQUIREMENTS: QUERY3 will run on any computer system that can support ANS COBOL and sequential, fixed-length records. It is currently operational on IBM System/360 and 370 DOS, OS, and VS equipment from Model 30 up; on the CDC 3000 Series and 6000 Series under SCOPE, KRONOS, and MASTER; and on the Burroughs B 3500, UNIVAC Series 70, Honeywell Series 2000, NCR Century 200, and DEC PDP-10. Written in ANS COBOL, QUERY3 can easily be transferred to additional machines.

PRICING: QUERY3 is available under a permanent license or 3-year lease. Maintenance is free under the lease and free for the initial year after purchase; thereafter, maintenance is available at ½ of 1 percent of the purchase price per month. Azrex provides COBOL source code, an object deck, several copies of documentary manuals at different levels, installation, set-up tests and system tuning, and a 2-hour instruction seminar for users, all under a 2-week free trial period. Improvements to QUERY3 are distributed free of charge to all customers under maintenance agreements.

Module	Purchase	3-year Lease
Basic tape/disk QUERY3 with SORT verb	$8,000	$288/month
Disk ISAM module	1,500	55/month
On-Line Inquiry Module	1,500	55/month
Multi-File access module	2,500	90/month

INITIAL DELIVERY: QUERY3.0 – March 1971. Current version, QUERY3.3 – February 1973. ISAM module – March 1972. On-Line module – November 1971. Multi-File module – June 1973.

CURRENT USERS: Basic QUERY3 – 15. ISAM module – 3. On-Line module – 2. Multi-File module – none to date. ■

EZ-Task
Beverly Bancorporation, Inc.

EZ-Task is a "partition operating system" for IBM System/360 and 370 DOS computers. This monitor uses DOS multi-tasking macros to control up to 9 subtasks and 512K bytes of main memory. Thus, operators can easily run several jobs within one DOS partition.

MANAGEMENT SUMMARY

EZ-Task is designed to help some IBM System/360 and 370 DOS installations increase their system throughput by creating the effect of additional processing partitions, which may in turn eliminate the need to convert to DOS/VS. What's more, the EZ-Task system is modestly priced at a one-time charge of only $1,000. All that is required is a DOS supervisor capable of supporting subtasking, plus operator training in the few additional EZ-Task console commands and responses to the new messages. Also, programs should be made self-relocating, an effect that would be beneficial under any circumstances.

EZ-Task uses DOS multi-tasking macros to control up to 9 subtasks in up to 512K bytes of main memory, all in one DOS partition. This is done with no specific tailoring of the applications programs, which would be required if conventional multi-tasking techniques were used. Moreover, the jobs involved can be run as subtasks under the EZ-Task monitor or as stand-alone jobs in their own partitions under the DOS operating system. Installation and use of EZ-Task do not require extensive modification of supervisor routines or changes to application programs, except for certain specialized routines, which require only a straightforward one-for-one macro replacement.

EZ-Task is a monitor, which means its function is to control a stream of jobs, allowing each to have proper access to the resources of the computer system. The monitor itself uses no peripheral devices, leaving these available for the programs, and does not use DOS job control except at its own loading and termination. EZ-Task uses only 6K bytes of main memory for itself.

EZ-Task is an operator-oriented monitor. As it controls the execution of subtasks, handles terminations, and manages memory, it communicates with the operator via the system console. Within the limits of the resources, the operator can control the number of subtasks and the program executing within each subtask as he wishes. He can also make inquiries as to the status of the EZ-Task partition. EZ-Task memory management is dynamic; i.e., as soon as a subtask releases a 2K block of memory, it is made available for another that may need it.

Finally, EZ-Task provides support for DOS job accounting routines. Thus, it is entirely feasible to use the subtask monitor even where accounting is required or chargebacks must be generated.

USER REACTION

At this writing, there are three EZ-Task users besides the vendor, and one of these is a new installation. However, Datapro contacted the two experienced EZ-Task users and gained a wealth of information – all favorable.

The first user has been running EZ-Task on his 240K System/370 Model 135 since February 1973. The EZ-Task monitor resides in 6K bytes in the F1 (Foreground

CHARACTERISTICS

SUPPLIER: Beverly Bancorporation, Inc., 1306 West 105th St. Chicago, Illinois 60643. Telephone (312) 445-2200.

BASIC FUNCTION: Monitoring of subtasks in a System/360 or 370 DOS computer so as to virtually act as an operating system in a single partition. EZ-Task controls a partition's resources (dynamically assigning main memory in 2K-byte blocks), provides operator console communications, and interfaces with DOS job accounting. Up to 9 subtasks and 512K bytes of memory can be controlled.

OPERATION: EZ-Task is installed by cataloging the relocatable modules and link-editing all phases. After those steps, it can be tested using the supplied EZ-Test program.

When the system is installed, the operator allocates main memory, batches the partition, and enters a // JOB EZ-TASK statement, using DOS facilities. He then executes EZ-Task, which builds a table of memory available, relocates itself, and marks a "core" table to reflect the main task boundaries. The core table is a string of bits, with each bit representing a 2K-byte blocks of memory.

As the operator initiates jobs, the table is marked to indicate which blocks of memory are in use, and the blocks are dynamically released upon termination of a job. Upon execution, EZ-Task attaches a single subtask which is immediately placed into the wait state. The main task issues a ready message and is put into a wait status to await an operator request. Now the operator can start a job using the waiting task, and can also add more subtasks and run other jobs. As jobs terminate, their associated subtasks are placed in the wait state, thereby maintaining constant availability to the partition.

EZ-Task uses no resources except for 6K bytes of memory, and uses DOS job control only for its loading and termination. It manages memory, monitors subtasks, and facilitates smooth operations via console commands. There are, however, restrictions in the use of EZ-Task, due to the fact that the jobs run under the monitor are subtasks and the fact that EZ-Task is not a supervisor. These restrictions are not overpowering, and quality operators can work around many of them. They are as follows:

- Job control is not invoked when subtasks are started, and statements such as // ASSIGN, // DATE, // UPSI, etc., cannot be a required part of program operations. However, EZ-Task console commands can effectively replace most of these statements.
- STXIT IT may not be issued within a subtask.
- STXIT OC may not be issued within a subtask. This means that the subtask program is not externally interruptable. However, a substitute (T$STXOC) macro permits the operator to interrupt a subtask via the $MSG command.
- Since there is only one communications region for the batch partition, and its contents can be affected by any subtask, its integrity cannot be guaranteed; so a short "dummy" communications region is maintained for each subtask.

EZ-Task
Beverly Bancorporation, Inc.

1) partition of 144K bytes and controls subtasks of CICS, using 82K bytes, POWER, using 28K bytes, and CAROL (Citizen's Audio Response On-Line, another Beverly Bancorporation product, for control of IBM 7770 Audio Response Units) using 28K bytes. This user reports that the system has consistently performed as well as or better than represented. Its ability to perform, in his opinion, is limited only by DOS. The user also states that he has noticed no degradation in program performance from stand-alone timings, even when all three subtasks are running concurrently. Installation required only one change in CICS (due to use of timer facilities) and changes to about 12 instructions in DOS (due also to special operations).

The second user has been running EZ-Task since May 1973 on an IBM System/370 Model 145 system with 256K bytes. In a 100K-byte F1 partition, he runs EZ-Task in 6K bytes, some on-line audio response units and terminals using another Beverly Bancorporation product in 46K bytes, a second audio response application in 20K bytes, and keeps 28K bytes available for additional subtasks. He has run up to two concurrent subtasks in this area, for a grand total of four subtasks running concurrently in F1. In fact, this user really gets mileage out of his DOS system, because he also uses GRASP (Report 70E-760-11) in its own FO "dummy partition," and has used GRASP's Job Accounting (Report 70E-760-12) with EZ-Task. The user sees no other way, with DOS and this hardware, that he could have two audio response jobs running together all the time and still process other jobs constantly in the F2 and BG (Background) partitions. The user says that EZ-Task was installed in about an hour.

Both users voiced the highest possible satisfaction with Beverly Bancorporation as a vendor, and both also use other Beverly software products. Both also said that their operators had no trouble learning to use EZ-Task.

In summary, EZ-Task appears to be a fine product from a fine vendor that should be examined by any DOS user who is beginning to feel pressed into DOS/VS or an upgrade to OS. Many of these users may find it possible to use EZ-Task to stretch their hardware dollars by increasing the cost-effectiveness and capabilities of their present systems. □

- A MAP command will show only the main task being executed in an AP (asynchronous processor) partition, so EZ-Task provides a $MAP to show the subtasks in a partition.
- The ATTENTION ROUTINE CANCEL command cancels the main task and all subtasks in the partition. But EZ-Task lets an operator cancel just a subtask, etc., as detailed later.
- If programs are not self-relocating, then the operators must be proficient enough to predict the memory execution address that EZ-Task will assign to a program.
- Programs must not alter the contents of memory beyond the original load boundaries of the initially loaded phase.

The T$TASKS macro maintains a list of the programs permitted to run under EZ-Task control for the systems programmer. The macro's parameters describe the program names and memory sizes. Changes in these must be reflected in the T$TASKS macro. Incorrect size specifications could cause one subtasked program to alter the memory allocated to another.

EZ-Task decides how much main memory to allocate for each subtasked program based on one of four criteria: (1) the program's size as expressed in the T$TASKS table; (2) the size entered via the console, if the operation is a test; (3) the size expressed by the JASIZE= parameter of T$TASKS, if it is coded; or (4) the size required to load the initial phase from the core image library, but only if the VERIFY= parameter is coded with the T$TASKS macro.

EZ-Task lets the operator simulate many job control functions and also inquire as to the status of the EZ-Task partition. When the operator wishes to communicate with EZ-Task, he interrupts the partition by standard DOS methods (e.g., Interrupt Key for BG or MSG command for FG), and EZ-Task responds with the message "TM104A $FUNCTION." The operator then enters one of 20 EZ--Task commands (all of which are preceded by a $) to perform an operation such as these: (1) start a job, if a subtask is available and sufficient memory can be located; (2) cancel a currently running job; (3) interrupt a job via linkage to the operator communications routine; (4) print a snapshot dump of the subtask on SYSLST; (5) assign a programmer logical unit to a hardware device; (6) cycle down the partition by issuing EOJ in maintask after all subtask jobs are inactive; (7) cancel the entire partition at once; (8) notify the operator of the number of tasks assigned and the number of jobs currently active; etc.

The $AJA (active job accounting) command instructs the EZ-Task system to load the standard phase $JOBACCT upon termination of each subtask. This phase is fetched into the memory just vacated by the subtask, and the size of $JOBACCT must be entered as the JASIZE= parameter of the T$TASKS macro in order to support job accounting. The job accounting table filed is relatively standard, with only a few entries padded.

EZ-Task has 55 formatted messages for diagnostic and control purposes. For example, $ NUMBER means that the operator has requested a function relative to a specific subtask, but has failed to enter the $ ID number of the subtask. All of the messages are quite clear, such as "INSUFFICIENT CONTIGUOUS CORE."

PERFORMANCE: Please refer to the User Reaction section. Briefly, users note smooth performance in all aspects, no job degradation, easy installation, minimal operator reorientation, and excellent vendor support. Present user experience is limited to 3 or 4 subtasks in up to 144K bytes, except for the vendor, who has run up to 8 subtasks in 106K bytes. All experience is with the F1 partition.

HARDWARE/SOFTWARE REQUIREMENTS: An IBM System/360 or 370 DOS system with a multiprogramming (MPS=BJF), asynchronous processing (AP=YES) supervisor. A few other rather standard supervisor options and the time-of-day clock (System/370) or timer (System/360) are required. For most efficient operation, programs should be made self-relocatable. Private or IBM catalogs and/or relocatable libraries are supported. EZ-Task itself uses 6K bytes of memory.

PRICING: EZ-Task is available for a 1-time charge of $1,000, which includes an object deck, documentation, installation, maintenance and updates, and training.

INITIAL DELIVERY: February 1973.

CURRENT USERS: 3 as of December 1, 1973. ■

RELOCATE
Boeing Computer Services, Inc.

MANAGEMENT SUMMARY

RELOCATE is one of the best of a considerable number of enhancements to IBM's DOS that were available prior to the announcement of DOS/VS by IBM in August 1972. With that announcement, much of the steam was taken out of the marketing efforts for DOS enhancements by independent software vendors.

DOS/VS (Release 28 of IBM's DOS) includes a self-relocation capability that supersedes the need for RELOCATE. Furthermore, the availability of DOS/VS at no additional charge on System/370 Models 135, 145, 155-II, and 158 tends to offer an almost irresistible incentive to current System/370 DOS users (who can make excellent use of RELOCATE at this time) to upgrade to DOS/VS. Coupled with the end of free support for DOS on System/360 or 370 computers by IBM as of March 31, 1973, a truly remarkable front of pressure is being applied to DOS users to give up DOS and/or the System/360 in favor of OS and/or DOS/VS on the System/370. (An analysis of IBM's upgrade strategies can be found in the System/370 report, 70C-491-04).

On the other hand, until mid-1973, DOS/VS is at best a "paper tiger", and DOS users of either a System/360 or 370 can well profit from a DOS self-relocatability enhancement until then. Subsequent to the free availability of self-relocatability in DOS/VS, the market for RELOCATE will be pretty much confined to a dwindling list of System/360 DOS users–but their ranks currently number some 10,000. For these existing DOS installations, RELOCATE can provide a useful additional facility.

Self-relocatability permits savings in the following areas:

- Machine time–RELOCATE eliminates the need to catalog the same program in the Core Image Library (CIL) more than once, for use in more than one partition, and similarly eliminates the need to recatalog the program when the supervisor or partition size changes.
- Programming time–self-relocatable programming is a complex and time-consuming effort if not done automatically.
- Operations–only one set of Job Control Statements per program is required, and job scheduling is greatly simplified since any job can go into any partition (space and peripherals permitting).
- Disk storage–RELOCATE saves Core Image Library (CIL) space since the program need be cataloged only once.

As a concept, relocatability (whether by "self" or by an operating system agency) offers a great potential for increased system resource utilization. Under unaided

RELOCATE adds self-relocation to IBM's DOS for COBOL, RPG, PL/1, FORTRAN, or BAL programs. Two versions are available–one that requires modification of the supervisor and one that does not. A dependable package, RELOCATE is well worth investigating by any DOS installation.

CHARACTERISTICS

SUPPLIER: Boeing Computer Services, Inc. (a wholly owned subsidiary of The Boeing Company), P.O. Box 708, Dover, New Jersey 07801. Telephone (201) 361-2121.

BASIC FUNCTION: RELOCATE is an IBM System/360 or 370 DOS enhancement that extends a self-relocation capability to nearly all monolithic or modular BAL, COBOL, RPG, PL/1, or FORTRAN application programs.

OPERATION: Two versions of RELOCATE are available, both of which provide essentially similar functional capabilities for users. RELOCATE I uses a transient routine to make user programs self-relocating and requires no modification to the DOS supervisor. RELOCATE II requires that the self-relocation function be assembled into the DOS supervisor as a macro. In either case, no special effort is generally required by the user at run time or during program preparation. Certain restrictions, however, are placed upon programs for which self-relocation is desired; these limitations are identified later.

For either version of RELOCATE, a control section (CSECT) of about 100 to 300 bytes is appended to the end of each phase or program at link edit time. At the time the program is subsequently loaded into core, the appropriate RELOCATE routine (a transient for Version I, or a macro incorporated into the supervisor for Version II) is executed. This routine calculates the actual relocation or displacement factors and modifies the address constants (ADCONS) in the user's program to reflect the partition into which the program is currently loaded. While most programs can be made self-relocating by RELOCATE, a number of application program conditions require special treatment or otherwise impose restrictions upon self-relocatability. These conditions are individually treated in the documentation provided by BCS, and techniques are described to minimize or overcome these difficulties:

- BAL programs with half-word address constants (ADCONS) may not be able to run in foreground partitions. Primarily because of this, the ANS COBOL compiler cannot be made self-relocating through the use of RELOCATE.
- The root phase or program in a multiphase or modular programming environment cannot be made self-relocating if the ROOT specification is used. A procedure is described in the RELOCATE documentation, however, to simulate the ROOT specification and allow the root phase to be made self-relocating.
- Registers 1 and 15 are used by the RELOCATE Control Section at the back of each program or phase, and thus the contents of these registers are not saved from one phase to another.
- The ANS COBOL compiler must be modified as described in the RELOCATE reference material before COBOL programs using the SORT verb can be made self-relocating.

RELOCATE
Boeing Computer Services, Inc.

▷ DOS, programs are non-relocatable in the core image library; i.e., they have been assigned to specific main memory locations. Thus, if one of the three partitions in a DOS multiprogramming environment is free, but not the one for which programs awaiting execution have been cataloged, that partition (and share of the system resources) must remain idle. Relocatability allows a program to run from any location in main memory, provided that enough space and peripheral devices are available. Larger systems with plenty of peripherals naturally have more to gain from relocatability, since there is more room to maneuver from a scheduling point of view, and also because the potential for wasted resources is greater.

RELOCATE is offered in two versions, both of which support the COBOL, RPG, PL/1, FORTRAN, and BAL languages and require that no changes be made by the user to his application programs. RELOCATE I does not require any changes to the DOS supervisor, whereas RELOCATE II assembles the self-relocation function into the supervisor as a macro. Under RELOCATE I, system design and subsequent operation are somewhat more complex for multiphase programs than under RELOCATE II.

Users of RELOCATE contacted by Datapro report that the package performs as advertised, but that certain limitations on internal sorts or multiphase (or modular) programs cause some inconvenience—particularly with respect to the COBOL compiler, which cannot be made self-relocatable by RELOCATE. (It should be noted that this problem with respect to the COBOL compiler is a direct result of IBM breaking its own rules through the use of half-words and violating guidelines set forth in the DOS System Programmers Guide for developing self-relocatable programs.) These limitations are discussed in the Characteristics section of this report. Otherwise, user feedback is quite positive. One user selected RELOCATE over competitive systems mainly because of the excellent documentation provided.

Although BCS sells and supports RELOCATE from 17 nation-wide sales offices, a "self-destructing" demo facility is provided with the package that allows the user to make his own tests without sales pressure for a predetermined length of time. Thereafter, the RELOCATE test package erases itself. Other independent sources for self-relocatability and other DOS enhancement facilities are available—both as standalone packages and as combinations of multiple DOS enhancements—and the DOS user will be well advised to investigate a variety of enhancements before making a decision regarding the future of his operating system. A number of these enhancements can effectively be used in conjunction with RELOCATE, while others overlap functional capabilities.

Of particular note in such an investigation are the projected plans of the vendor. Look for the possibility of price reductions and other usage concessions as a gambit for the suppliers of DOS enhancements to gain new business opportunities in your computer installation. This is a particularly good bet now, when it seems likely that comparatively few additional sales of such packages will be made to the general marketplace in the next year, and still fewer after that. (It's worth noting that BCS lowered RELOCATE prices by as much as 25 percent during the time this report was in preparation.) Careful shopping for DOS enhancements now could produce significant short-range payoffs. □

➤ The control section added by RELOCATE to each phase of the user's object program contains the locations of all address constants in the program. When the self-relocating object program is loaded for execution, the entry point for the program is in the appended control section. When control is first passed to the user's program, such address modification as is required to accommodate the program's specific location is made. RELOCATE itself is a self-relocating program and can be executed from any point in main memory.

PERFORMANCE: There are three major aspects of RELOCATE usage, each of which entails a different level of system resource utilization. When processing user object programs at catalog time to append the self-relocating control section, BAL programs consisting of about 100,000 source statements take less than 30 seconds of additional time to be cataloged. Most programs require only about 5 to 15 seconds of additional time. At execution time, the self-relocating control section generally takes less than 250 milliseconds to execute per program phase on a System 360/30 with IBM 2314 disk drives. The basic RELOCATE package can be installed and cataloged on the user's DOS in about 10 minutes.

HARDWARE/SOFTWARE REQUIREMENTS: RELOCATE can be installed on any System/360 or 370 under DOS Release 26 or lower that includes the Linkage Editor and loader facilities supplied by IBM. The minimum main memory requirement for RELOCATE is 16K bytes. This partition size is large enough to accommodate a monolithic or modular program with a total of 350 INCLUDE cards and entry points, 125 control sections per phase, and 115 address constants. The actual size of the control section appended by RELOCATE is 45 + 4C + 4R bytes, where C and R are the number of control sections and address constants contained in the user's object program, respectively. For RELOCATE II, about 650 bytes are added to the DOS supervisor by the RELOCATE macro.

PRICING: Either version of RELOCATE is available on a perpetual license basis and may be used on any number of computers at the same geographical location. Shorter license periods with reduced prices are also available. A charge of $1,800 is made for an initial six months of usage, with a charge of $1,200 for each additional six-month period. A liberal option to convert to perpetual license is provided; contact BCS for details. Prices for the perpetual license are as follows:

	Plan A*	Plan B**
First geographical location	$2,950	$3,950
Second geographical location	2,450	No charge
Third, fourth, or fifth geographical location	2,150	No charge
Sixth and each subsequent geographical location	No charge	No charge

* Plan A provides for individual maintenance at each geographical site.

** Plan B provides for unlimited usage worldwide, with maintenance by BCS to a single centralized customer site only.

INITIAL INSTALLATION: March 1970.

CURRENT USERS: More than 160 accounts with about 185 geographical locations. ■

RE-ACT
Boeing Computer Services, Inc.

MANAGEMENT SUMMARY

RE-ACT is a package that allows programmers or non-programmers to obtain management information they may require by simply filling out a single form, using English-like statements. Requests for reports are keypunched, using an overlay mask to prevent errors, and input to the system. Many format standards and report items (e.g., dates, page counts, tallies, control breaks, spacing, paging, and even listing of the request statements) can be specified with only a check mark on the form, and all are defaulted.

RE-ACT users can specify security information, report titles, headings, actions of data items upon others or upon constants, logical and/or arithmetical connectors, results, and sort and print sequences at various levels. Thus, the manager who learns to use RE-ACT can obtain information from data processing files in a short time, without the delay or expense of a programmer's efforts.

The complexity of a report can vary depending on the requestor's needs and familiarity with the request language syntax. A three-card request entailing only six words, a number, and a few check marks can constitute a valid request. Or, a report can be extensively titled, have imbedded computations and logical relations, be tallied and/or subtotaled at several levels, and be output-sorted to five levels. Output can be printed or sent to punched cards or magnetic tape.

RE-ACT comes in three versions for IBM System/360 and 370 computers: full OS (or OS/VS), full DOS (or DOS/VS), and mini-DOS (or DOS/VS). The differences are in main storage requirements, numbers of requests handled, and so on. Also available are a version for the Honeywell Series 200 and Series 2000 and another for the Honeywell Series 6000. Importantly, all versions are compatible across all operating systems and between IBM and Honeywell computer systems.

RE-ACT is the result of development that began in 1962 for the U.S. Air Force. The program was developed mainly by Mark Harren, now industry manager of Cybertech Data Systems, Inc. RE-ACT has as its ancestor the USAF Inquiry System completed in 1964 for use on a Burroughs B 5500. This was followed in 1965 by an IBM 1401/7010 version for Ling-Tempco-Vought, and in 1968 by a UNIVAC 1108 version for University Computing Company. Cybertech began to market IBM/360 and 370 and Honeywell 200 versions in 1970. The Honeywell 6000 version was developed in 1973.

Boeing began to market RE-ACT in September 1972. Cybertech markets only the Honeywell versions now, with Boeing marketing the IBM versions. Succeeding versions of RE-ACT were not simply conversions, but were complete rewrites with significant enhancements.

RE-ACT can be purchased as a complete package or as four separate subsystems on a stand-alone basis. It is in this latter way that one-third of the customers initially acquired RE-ACT, picking up the additional subsystems later. The RE-ACT subsystems are Retrieval, explained ▷

RE-ACT is a powerful information retrieval and reporting system for management that can be used by non-programmers on IBM System/360 or 370 DOS, OS, or VS systems and also on Honeywell Series 200, 2000, or 6000 systems. File maintenance, special forms reporting, and special utilities are optional features.

CHARACTERISTICS

SUPPLIER: For IBM systems—Boeing Computer Services, Inc., P.O. Box 23426, Seattle, Washington 98124. Telephone (206) 773-6161.

For Honeywell systems—Cybertech Data Systems, Inc., Suite 125, 2655 Villa Creek Drive, Dallas, Texas 75234. Telephone (214) 620-2551.

Cybertech also operates a service bureau with Honeywell equipment. Boeing has a service bureau with IBM equipment, and also has sales offices in Anchorage, AK; Atlanta, GA; Boston, MA; Cape Canaveral, FL; Chicago, IL; Detroit, MI; Dover, NJ; Ft. Lauderdale, FL; Greensboro, NC; Huntsville, AL; Los Angeles, CA; New Orleans, LA; New York, NY; Olympia, WA; Philadelphia, PA: Portland, OR; San Francisco, CA; Washington, DC; and Wichita, KS. Overseas marketing is conducted by Cybertech, through Dallas.

BASIC FUNCTION: File retrieval and report generation. RE-ACT is a package of four modules: Retrieval, File Maintenance, Matrix, and Special Utilities.

Retrieval is the basic subsystem and the one most users start with and use the most. It accesses files in a data base, processes the information, then formats and writes reports on the selected output medium. Up to 45 requests (20 in a mini-DOS version) can be handled in a single pass of the file and printed out in separate reports. Arithmetic calculations can be made on all data retrieved for processing. Grand totals and subtotals through six levels can be produced. Ouput data can be reported in any sequence and can be on punched cards, printed forms, or magnetic tape. Logical relations can be invoked, and the request form is extremely simple, using only check marks to specify some standard report items.

The File Maintenance subsystem helps in data base creation by using uncomplicated statements to define all requirements for changing, deleting, updating, or creating new master records from input transaction records or activity records. Numeric data is checked to see that it contains only numeric information; input transaction data is checked and validated against a criteria table for each field; and numeric and packed decimal fields of output master records are audited to see that no blank information is recorded therein. Processed transactions and deleted master records are recorded on a separate audit file.

Matrix is the subsystem that simplifies printing output on preprinted forms, such as W-2's, checks, invoices, and mailing labels.

The Special Utilities subsystem contains five programs that can be readily applied to rewrite, reorganize, and reformat files. These routines also facilitate matching and merging of records and handling of duplicate records.

OPERATION: After a straightforward one-time file definition step by the systems staff, users can employ the RE-ACT system with relative ease. Operation is about the ➤

RE-ACT
Boeing Computer Services, Inc.

▷ above; File Maintenance, whose function is self-explanatory; Matrix, a feature that simplifies printing on special forms, mailing labels, etc.; and Special Utilities, a set of 5 programs for matching, merging, duplicate record handling, and more.

USER REACTION

Datapro interviewed six RE-ACT users, five with IBM equipment (one on Boeing's service bureau) and one with a Honeywell 1250. All report extremely good vendor relations (they were Cybertech customers, except for the service bureau user, but Boeing has a well-earned service reputation of its own), extreme ease of installation (about one hour is typical), and great pleasure with the Retrieval module. The other modules are not used to the degree that the Retrieval module is, nor have as many customers had the additional modules for as long.

The power of the Retrieval module is indicated by the fact that three of the users use it in production reports, some run as frequently as daily. One user mentioned that RE-ACT handles OS partitioned data sets, and a DOS user reported that one weekly production job gives him seven different reports in one file pass. Another IBM user, who brought RE-ACT up from his 360/40 with 2314 disks to a 370/145 with 3330's, mentioned that he uses RE-ACT for 15 weekly reports.

The Honeywell user reports having used RE-ACT about 250 times since March 1972; he feels that it has paid for itself many times over. The service bureau user reports that RE-ACT Retrieval has permitted him to obtain needed data from proprietary software files at an average cost of about $70 per report, when otherwise it was unaffordable at an estimated $1,200 to $2,000 per report.

The vendors feel that RE-ACT competes against Informatics MARK IV (see Reports 70E-010-40 and 70E-500-01). RE-ACT certainly costs less, but is also somewhat less powerful. It probably is more in a class with Program Products' Data Analyzer, but RE-ACT differs in that, while it has fewer facilities of a special nature, its output does not require compilation. The Data Analyzer, comparable in cost, is described in Report 70E-691-03.

The vendors' forthrightness is exemplified by the fact that they readily admit to RE-ACT's chief present shortcoming: it cannot be used with multiple-file input or to accept input from a data structured base without a user-coded interface. Most customers use it only for single-input-file jobs. One user runs multiple-input-file jobs by a pre-merge, process, and un-merge method. Cybertech recommends using COBOL for multiple-input-file jobs, or employing the RE-ACT user exits and user-code for multiple-file and structured data base input.

Ease of use is also a consideration, and one user mentioned that his non-programmers were successfully coding requests the first time after a 1-day course.

In conclusion, Datapro feels that RE-ACT is a potentially valuable package for IBM 360/370 and Honeywell 200/2000/6000 users who require quick turnaround of reports requested by management.□

► same as for most packages of its genre: the requestor fills out a worksheet form, the form is keypunched, and the cards are fed to the system. RE-ACT generates the programs to be processed against the file(s). RE-ACT is a batch program.

PERFORMANCE: According to user reactions, RE-ACT is a good performer, generating report programs that suffice even for frequent production usage. Retrieval, however, should be regarded as a single-input-file function.

HARDWARE/SOFTWARE REQUIREMENTS: On an IBM System/360 or 370 under OS or OS/VS (any version) RE-ACT requires about 40K to 80K bytes. There is a 40K DOS or DOS/VS version, and a reduced-scale 22K DOS or DOS/VS version. The former two versions can support 600 data name table entries, while the latter can have only 200. There is no limit to the data base block size in the OS version, but it is limited to 7,300 bytes in the full DOS version and to 2,000 bytes in the mini-DOS version. The OS version also places no limit to the amount of memory available for the user's own "exit" coding, but this is held to 10,000 bytes in the DOS version, and none can be added to the mini-DOS version. The OS and DOS versions can have 80 work area fields and a 10,000-byte processing table, but the 22K DOS version is limited to 30 fields and 4,000 bytes, respectively.

A Honeywell 200 or 2000 system requires 32K characters for RE-ACT, which can run under the Mod 1 MSR, OS/200, or OS/2000 operating system. On a Honeywell 6000, 20K words are required for RE-ACT to run under any available operating system.

PRICING: RE-ACT is available for IBM systems under rental or perpetual license, both with included maintenance. For Honeywell systems, only a perpetual license is available. Multiple-copy discounts are available; contact Boeing or Cybertech, as appropriate, for details and also for training charges.

	Perpetual License	Rental (per month)	Installation Charge
For IBM systems:			
RE-ACT (all four subsystems)	$15,000	$500	$1,200
Retrieval (only)	9,000	300	850
File Maintenance	4,500	150	450
Matrix	2,250	75	NC
Special Utilities	2,500	75	NC
For Honeywell 200 and 2000 systems:			
RE-ACT (all four subsystems)	7,500	–	–
Retrieval (only)	5,000	–	–
For Honeywell 6000 systems:			
Retrieval (only)	12,000	–	–
Matrix (only)	3,000	–	–

Supplies are extra at nominal charges. IBM DOS to OS conversion assistance costs $250 plus reasonable expenses.

INITIAL DELIVERY: 1970 for the current IBM versions and for the Honeywell 200/2000 versions. August 1973 for Honeywell 6000.

CURRENT USERS: About 40, 32 of whom are IBM users and 7 of whom are Honeywell 200 or 2000 users. One system for a Honeywell 6000 user has been delivered to date.■

Problem Program Evaluator (PPE)
Boole and Babbage, Inc.

MANAGEMENT SUMMARY

Systems Measurement Software (SMS) is a generic name Boole and Babbage uses to identify a range of software products and services aimed at detecting what a computer is doing, thereby giving an insight into ways to improve its operational efficiency. The Problem Program Evaluator (PPE) portion of SMS is currently supported for IBM System/360 and System/370 computers under DOS or OS, and an OS/VS1 version is currently in beta test.

Boole and Babbage identifies three types of software measurement products: program evaluators, configuration evaluators, and accounting systems. PPE obviously falls into the first class; the Configuration Utilization Evaluator (CUE) is covered in Report 70E-098-02, and the Computer Accounting System (SMS/CAS) is covered in Report 70E-098-03.

PPE is designed to identify the time spent executing the instructions contained in segments of main memory identified by the user. The technique used is to read tables contained in the operating system at intervals specified by the user and timed by the computer's interval timer.

The output is in the form of tabulated percentages and histograms (bar charts showing the frequency of occurrence of particular events). From the output, a user can identify those sections of the problem program that require the most execution time (and are therefore the most promising for optimization efforts), as well as the wait times associated with either specific locations or data sets.

The kinds of inefficiencies most easily identified through PPE include poor iterative procedures, residency of supervisor modules, and excessive I/O times. Once the areas of concern are identified, it is up to the programmer to discover methods for improving the operations. Boole and Babbage, however, does not leave it entirely up to your programmers; helpful methods and tips are continuously being published in a newsletter *(SMS Product Status Report)* that goes to users of the package.

The promotional literature distributed for PPE and other SMS products and services includes a discussion in statistical terms of the factors affecting system throughput. The bottom line resulting from these discussions is no surprise. To improve system performance, you have to reduce the execution time of the problem program, reduce the time associated with supervisor calls, or eliminate I/O inefficiencies. Certainly a background in statistical methodology is not required to come up with these results.

Statistical theory is of importance to users of PPE, however, because the results are based on a fractional sample of the computer's operations rather than on a complete count. This is one of the most important theoretical differences between PPE and hardware monitors. By using sampling theory, an estimate of the confidence that can be assigned to a particular sampling rate can be made. ▷

PPE samples user programs to measure and report the activity of specified segments of problem program code. The output of PPE can serve as a guide to optimization efforts. Versions for IBM DOS and OS systems are available, and OS/VS versions are due in February 1974.

CHARACTERISTICS

SUPPLIER: Boole & Babbage, Inc., 850 Stewart Drive, Sunnyvale, California 94086. Telephone (408) 735-9550. See Report 70E-098-03 for other locations.

BASIC FUNCTION: PPE takes samples of the various operating system tables during the operation of a user (problem) program, while the program is actually running, and analyzes these samples in a separate job step to show a profile of the operations. Versions are available for IBM DOS, IBM OS, and IBM M65MP. OS/VS1 and OS/VS2 (Release 1) versions will be released around February 1974.

OPERATION: The system consists of two programs, an Extractor and an Analyzer, which are run as separate jobs or job steps. The Extractor samples the activity of the problem program and outputs this information as an Extractor data set. The Analyzer prepares Code Activity Reports from the contents of one or more Extractor data sets.

The program to be studied may reside either temporarily or permanently in a library, and is loaded and executed by the Extractor program. The required JCL changes involve DD statements for the Extractor data sets and possibly a JOBLIB card for the library (catalog procedures can be used). No change to the user's program is required.

The Extractor program samples the problem program under study at a rate specified by the user. Detailed data on the status of the problem program is collected within a range of core storage called the sample limits. These limits can be specified for each run, with a default to the region or partition bounds. Samples falling outside the limits are tabulated in less detailed terms, but sufficiently to determine the relative time spent inside and outside the sample limits for the problem program.

An optional feature, Expanded Module Analysis (EMA), lets the Extractor reside in a separate partition or region and run independently of the program being monitored. This is the method most users employ to measure long-running teleprocessing jobs, and it gives users the capability to measure IBM's CICS (Report 70E-491-02) and its related user modules.

An Analyzer run can be made immediately following the completion of an Extractor run, or at a later time if the Extractor output data set is saved. The Analyzer can accept one or more Extractor data sets for combination into a single report. The Extractor data sets are used as the basis for several reports.

The sampling is based on the use of interrupts from the real-time clock. The fastest rate is every 17 milliseconds, which involves taking a sample every time the clock "ticks," but slower rates are often used, such as every other tick, every tenth tick, etc.

Arguments about the correct sampling rate are based on two parameters: the amount of interference and the randomness of the sample. The interference is estimated to be about 1 millisecond per sample taken, and the number ➤

Problem Program Evaluator (PPE)
Boole and Babbage, Inc.

▷ Obviously, the more samples taken, the greater the confidence that can be placed in the result, right? Wrong. The reason is simple. The sampling process (time required to access and store the data) begins to interfere with program execution times as the number of samples grows. Fortunately, the number of executions in a computer program is usually a very large number, which means that a substantial number of samples, enough to generate a high confidence level, can be taken with minimum interference to the program.

USER REACTION

Users who reported their experience with PPE in the August 1973 Datapro software survey (Report 70E-010-040) and users contacted specifically for this report agree that PPE is a package that lives up to the claims of its vendor. Users rate their overall satisfaction with the package as good to excellent and assign similarly high ratings to the vendor's technical support.

One disappointing aspect has been the delay on the part of Boole & Babbage in developing new versions of PPE for IBM's virtual storage operating systems. Press releases heralding an OS/VS1 version of PPE have circulated since August 1973, and by January 1974 the company reported that six beta test versions of PPE for OS/VS1 had been installed. But as yet the vendor isn't mentioning a DOS/VS version at all.

Some impressive results have been achieved with PPE. Boole and Babbage, in fact, guarantees that it can improve the user's overall situation—and cites examples. A food processing company running a COBOL stock-control program for nearly an hour a day located an area of inefficient, highly-used code and eliminated 36 minutes per day from the run time. An insurance company found that a 400-byte section of coding in a large PL/I program was using 25% of the total run time, and cut the run time by 20% after a short upgrading of the code in the identified area. Other companies have reported similar savings.

However, a few users do not feel that they have benefitted to such a munificent degree. And their comments point up the chief difficulty in using PPE. The package is not a magical cure-all (and is not promoted as such). Multiple passes refining the areas of concentration and trying out new program patches are required to take advantage of the information gleaned from PPE.

To get your feet wet without committing yourself to the full price, you can contract with Boole and Babbage to conduct a three-day on-site study. This service costs $300 per day, plus travel and related living expenses. □

➤ of samples needed to provide an adequate pattern is estimated to be 8000. The clock starts at a random time with respect to the Extractor program and the program being examined; the samples obtained from several runs are therefore random and quite representative.

The Code Activity Report shows the actual counts indicating what portion of the program was being executed when the clock "ticked." These are presented relative to a segment or module identified by the user, who inputs the relative upper and lower bounds of memory locations for which he wishes a report. (The user normally defaults these boundaries to the entire module.) Thus, three counts are ➤ shown: number below lower boundary, number within bounds, and number above upper boundary. In addition, the number of interrupts is shown.

Other reports present the percentage of time spent in each identified segment or module. These are presented with I/O wait time included and excluded to concentrate on the area desired. In addition, the input control cards can specify intervals within segments for a closer analysis of individual portions; these results are presented as the percentage of the total study time spent in each selected interval. A list of intervals is also shown, ordered by activity.

Another of the reports presents a histogram which shows the percentage of time spent in each interval and allows quick visual identification of the time-consuming portions of the program.

All of these reports are geared to locating those areas of a program that use the most time and are therefore the most potentially rewarding areas for program improvement. In essence, the user identifies the specific core areas he wishes to investigate, the Extractor samples the whole program, and the Analyzer interprets the resulting samples relative to the indicated segments.

An alternate handling of I/O wait times is allowed. In the standard technique, I/O wait time is associated only with the section of program code responsible for issuing the commands causing the wait. The alternate method, called Data Set Oriented Wait (DSOW) for PPE/OS and File Oriented Wait for PPE/DOS, assigns wait times against the associated data sets (files). This permits identifying files that can potentially benefit from such changes as block size, access method, and/or number of buffers.

PERFORMANCE: The Extractor normally adds less than 1 percent to the running time of the program being analyzed: the exact added time is a function of the sampling interval. The speed of the Analyzer program is variable, but a typical complete analysis run on an IBM System/360 Model 30 requires less than 5 minutes.

HARDWARE/SOFTWARE REQUIREMENTS: Versions of PPE are available to run on an IBM System/360 or System/370 under DOS (Release 25 and later), OS (MFT Release 19 and later or MVT Release 19 and later), and M65MP (Release 17 and later).

Under DOS, the Extractor program requires 4K bytes (data set output on tape) or 6K (data set output on disk) and the Analyzer requires 24K bytes. Under OS, the Extractor program requires 6K bytes plus access methods and the Analyzer program requires 56K bytes plus access methods, I/O buffers, and overlay supervisor.

A few restrictions apply to the use of the Extractor program. Obviously, the configuration must include an interval timer. Under PPE/DOS (DPPE using the vendor's nomenclature), only the Extractor program can have access to the interval timer. Under PPE/OS (MPPE for M65MP, PPE-1 for MFT, and PPE-2 for MVT), the Extractor program may not be time-sliced or rolled in or out, and checkpoints in the user program are ignored.

PRICING: For PPE/OS, the one-time purchase price is $8,800, which includes the first year's maintenance; maintenance after the first year is available for $880 annually. For PPE/DOS, the one-time purchase price is $4,400; maintenance after the first year is available for $440 annually. The EMA option costs $1,000 plus $250 per year for maintenance after the first year.

Included in the maintenance service arrangement are correction of bugs, version updates, and subscriptions to the *SMS Product Status Report.*

INITIAL DELIVERY: February 1969.

CURRENT USERS: Approximately 420 as of October 1973. The majority of PPE users also use CUE. ■

Configuration Utilization Evaluator (CUE)
Boole and Babbage, Inc.

MANAGEMENT SUMMARY

The Configuration Utilization Evaluator (CUE) is a companion to Boole & Babbage's PPE package in its SMS (System Measurement Software) product line. PPE (see Report 70E-098-01) permits evaluation of user program execution. CUE permits evaluation of the peripheral component usage of a particular configuration.

The results of the two-program CUE package—one program for extracting data samples and the other for tabulating a whole set of samples—are directed toward answering three primary questions:

- Are any components (channels, control units, or peripheral devices) over- or under-used to the extent that changes in the configuration or program mix would be helpful?
- Can improvements be made by altering the distribution of data sets stored on disk?
- Can residency decisions be made about transient SVC modules that will improve performance or reduce core requirements?

No recommendations are gained directly from observing the output of CUE. Instead, the user receives factual information on which to base his own recommendations.

With CUE, the question of statistical validity assumes a different role than with PPE. When evaluating a program with PPE, you have assurance that the program (the subject of the measurement) does not change, and the concern is whether or not a valid sampling of the program has been made. With CUE, after you have satisfied yourself that the sampling of a particular period is an accurate representation of that period, the question remains as to whether the period sampled is representative of the overall operations of the configuration. The results can be different for each different mix of programs run. The answer, then, is to make several surveys and, most importantly, to remain aware of the period sampled and exactly what it represents.

Questions of head movement time and possible relocations of data sets within a particular disk pack (volume) are answered more directly by another package in the SMS product line. The Data Set Optimizer (DSO) is also structured around the concept of an Extractor and an Analyzer program. The output of the DSO Analyzer program includes a map of the disk pack analyzed along with reports on head movement time and recommendations for reorganizing the data set distribution. One important difference between CUE and DSO is that DSO reports data set names as identifiers rather than physical addresses, making DSO easier to apply. However, the output achieved with DSO can be obtained from CUE with a little work. DSO is available for $3,600 by itself or for an additional $2,700 if purchased in conjunction with CUE. Used by itself, DSO is quite limited. Used with CUE, it is something of a luxury.

CUE, the peripheral usage evaluation package in Boole & Babbage's SMS product line, extracts this data from tables maintained by IBM's System/360/370 OS or OS/VS. A separate CUE Analyzer program tabulates the usage of channels, peripheral units, and initiator/terminator modules and monitors head positioning in disk drives and loading of SVC modules.

CHARACTERISTICS

SUPPLIER: Boole & Babbage, Inc., 850 Stewart Drive, Sunnyvale, California 94086. Telephone (408) 735-9550. See Report 70E-098-03 for other locations.

BASIC FUNCTION: CUE collects and tabulates the activity of peripheral devices, SVC's, and I/O channels. This information serves to pinpoint prospective areas for system performance improvement. The optional Plotter feature can create a record of sampled results over a period of time. CUE can be run on IBM System/360 or 370 computers under OS (MFT or MVT), OS-M65MP (Model 65 multiprocessing), OS/VS1, and OS/VS2 Release 1. A version for OS/VS2 Release 2 will be released approximately 6 months after that operating system becomes available.

OPERATION: The CUE package consists of two programs, an Extractor and Analyzer, which are run as separate jobs or job steps. The Extractor samples the activity of the computer over a period of time and outputs this information on tape or disk as an Extractor data set. The Analyzer tabulates and summarizes the information from one or more Extractor data sets in six different reports.

The Extractor program must operate as the highest-priority job in the system; otherwise data collection will not be complete and would probably be misleading. Data is collected on hardware usage (CPU, channels, and peripherals), disk drive head movements, and loading of transient SVC (supervisor call) modules. Control cards are used to specify the hardware components to be monitored for utilization, disk drives to be monitored, sampling interval, and duration of data collection. The maximum sampling rate is once every 17 milliseconds, the smallest resolution of the System/360 interval timer; multiples of this basic interval are normally used. The operator can terminate data collection at any time.

The resulting Extractor data set for one sampling period covers all programs active during the period. The basic information collected for each sampling point includes: CPU busy (supervisor, system task, problem program) or in wait state; channels and peripheral units busy; disk units seeking (heads moving), along with cylinder addresses at beginning and end of seek; linkpack and jobpack module usage (option); initiator/terminator activity (option–MVT); partition activity (option–MVT); and SVC modules loaded.

The Analyzer program takes the raw data and outputs reports based on parameters in control cards. Any of the devices specified in the Extractor phase can be ignored in the Analyzer phase. The period covered in the reports can be a reduced segment of the period the Extractor data set covered.

REPORTS: CUE has been updated extensively since the previous DATAPRO 70 report of February 1972, when there were just 3 CUE reports. There are now a possible 14,

Configuration Utilization Evaluator (CUE)
Boole and Babbage, Inc.

USER REACTION

CUE users are satisfied with this package from Boole & Babbage, and they gave Datapro convincing testimony as to the reasons for that satisfaction. For example, one user with three years of CUE experience on OS/MVT systems (from a 360/65 under OS/MVT Release 18.0 to a 370/165 under OS/MVT Release 21.7 with HASP) has this to say: CUE is easy to use, and has been augmented with vendor technical support that is responsive and of excellent quality. Boole & Babbage has also lent assistance in its usage, and the package has been used to reconfigure both batch and on-line systems to improve their performance. CUE is also used to help make determinations regarding equipment acquisition.

Another user contacted for this report claims that CUE solved queuing problems on his 370/145 and 370/155, both under OS/MFT, and that he expects to get similar results from CUE on the 370/145 when he completes a conversion to OS/VS1. The two systems share several 3330 disks. This user also praises CUE's speed and efficiency.

CUE also earned very respectable marks from the eight CUE/PPE users who reported their experience with this pair of SMS products in Datapro's first annual Survey of Proprietary Software Users (Report 70E-010-40). □

plus the 3 additional Plotter option reports. The principal CUE reports include:

- CPU Utilization—a breakdown of CPU time into supervisor, system task, and problem program state, which also lists each of these as a percentage of system active time (i.e., the time any device is busy). This report also gives a measure of CPU queue behavior; when the system is busy, the number of dispatchable tasks are totalled and recorded as the maximum number of tasks dispatchable. Also given in the report is the distribution of CPU busy, by percentage over time. This is shown in a graph, which might illustrate, for example, that the CPU was 100% busy for 20% of the time, 50% busy for 10% of the time, and so on. The graph's increments are 5%.
- CPU/Physical Channel Activity—a chart graphically showing the overlap between CPU busy time, CPU wait time, and usage of each channel. This chart can isolate which channels are causing the CPU to be in a wait state.
- Combinations of Configuration Activity—a report giving all combinations between channels and CPU states, so as to further isolate causes of the CPU being in the wait state. With a control card, the user can specify additional combinations not in the default report.
- Logical Channel and Device Utilization—a report giving total busy time and queueing characteristics for each channel. For each device, error recovery, seek time, non-seek time, total busy time, and the queueing characteristics are given. The queueing characteristics show which device or channel is really causing tasks to be in a wait state. Freeing these tasks can improve CPU utilization and throughput.
- Direct Access Volume Activity—total head movements on each volume, as well as total movements to alternate cylinders, are given by this report, which isolates volumes having a high number of head movements.

Other reports produced by CUE are as follows:

- Direct Access Volume Head Movement Summary.
- Direct Access Volume Head Movement Activity.
- Physical Channel Activity by Logical Channel.
- SVC Loads.
- Linkpack Area Module Usage (an MVT option).
- Jobpack Area Re-Entrant Module Usage (an MVT option).
- Initiator/Terminator Task Activity (optional).
- CPU Utilization by Protect Key (optional).
- CPU Utilization by Initiator/Terminator (supplied to users who purchase the preceding two optional reports).

PLOTTER OPTION: The Plotter spots peaks and valleys of CUE measures over time. It can thus provide direct correlation with the values of other measures. It can also pinpoint critical time periods for which complete CUE reports might be desired. The Plotter: (1) provides graphic representation of CUE measures, pinpointing high and low activity trends for specified intervals of time; (2) permits correlation of up to four measures on one graph, to indicate times when a complete CUE report may be desirable; and (3) produces an exception report based on user-set tolerance levels to identify time periods when values are out of range. Up to three other measures can be given if the first one is out of range.

PERFORMANCE: The Extractor program imposes a small demand on computer resources. Boole & Babbage estimates that each sample point requires only about one millisecond of processor time, making the maximum demand a little over 5 percent. At more normal sampling rates, the interference would be only about 1 percent or less.

The Analyzer program is a straightforward tabulation and listing that requres at most a few minutes to run. However, many runs are typically required to properly measure and monitor the performance of a system, so it should not be considered a one-shot operation.

HARDWARE/SOFTWARE REQUIREMENTS: Versions of CUE are available for operation under IBM System/360 or System/370 OS MFT (Release 19 and later), OS MVT (Release 19 and later), OS M65MP (Release 17 and later), OS/VS1, and OS/VS2 Release 1. An OS/VS2 Release 2 version will be available 6 months after OS/VS2 Release 2 documentation is released. The Extractor program requires 22K bytes of main memory plus access method modules. The Analyzer program requires 94K bytes (MFT or MVT) or 140K bytes (M65MP) plus access method modules, I/O buffers, and overlay supervisor. The Extractor data set can be output on magnetic tape or disk. The computer configuration must include the interval timer and floating-point instruction set. The Extractor program cannot be time-sliced or rolled in and out.

PRICING: CUE is available mainly on a purchased license basis. Leases can be arranged. The first year's maintenance is included in the purchase prices.

Product	Purchase Price	Annual Maintenance
CUE	$8,800	$880
Module Usage	1,250	250
Initiator/Terminator	1,000	250
Partition Activity	1,000	250
Protect Key	750	200
Plotter	2,000	450

INITIAL DELIVERY: February 1968.

CURRENT USERS: Approximately 400 as of October 1973; the majority of CUE users also use PPE. ■

SMS/CAS
Boole & Babbage, Inc.

MANAGEMENT SUMMARY

SMS/CAS is a billing and accounting system for IBM System/360 and 370 OS, OS/VS, and TSO systems. A natural extension of the Boole & Babbage System Measurement Software, PPE and CUE, it is designed to make programmers aware of costs and help provide long-range system performance measures.

SMS/CAS (Computer Accounting System) is a Boole & Babbage System Measurement Software product with four major design objectives: (1) to minimize losses of billings; (2) to ensure consistent, accurate, and complete billing procedures; (3) to develop programmer awareness of costs for resources used; and (4) to provide long-range system performance measures. SMS/CAS is a logical extension of the company's earlier System Measurement Software (SMS) products, PPE and CUE, which are described in detail in Reports 70E-098-01 and 70E-098-02.

Losses in billings that SMS/CAS can recover can come from bad SMF data (SMF, or Systems Management Facility, is an optional accounting module in the IBM supervisory program) or from not charging for all the resources in the system.

Consistency in billing procedures means that runs of the same job in the multiprogramming environment should be billed at approximately the same amount at all times. Accuracy, as far as Boole & Babbage is concerned, means that high-speed device usage should be charged for at a higher rate than low-speed devices. Completeness means that all resources in the system should be charged for.

If a programmer is aware of the costs of the resources he calls upon, he may be more inclined to reduce those costs, Boole & Babbage feels. Also, if he finds that most of the costs are in CPU time, he may be inclined to call upon Boole & Babbage's PPE to reduce that time.

There are three advantages to the long-range performance measures provided by SMS/CAS: (1) bottleneck areas that should be measured with a software measurement tool, such as one of the other SMS products, can be pointed out; (2) the most accurate billing rates can be determined; and (3) it may become easier to plan for future machine acquisitions.

These objectives are accomplished by the three basic systems within SMS/CAS. The Billing System produces charges at step end and job end, so that when a programmer gets his job back, he knows exactly what its costs were. The same Billing System program also produces periodic billing reports. The STATS System produces performance measures, program usages, and resources usages. Finally, the Edit System checks for missing or incomplete data and also provides user exits, so that the user can change SMF records or add new ones if he wishes.

The Billing System and the STATS System produce a number of reports, which are described in greater detail in the Characteristics section of this report. These reports are ▷

CHARACTERISTICS

SUPPLIER: Boole & Babbage, Inc., 850 Stewart Drive, Sunnyvale, California 94086. Telephone (408) 735-9550.

Boole & Babbage sales and service offices in the U.S. are located in New York, Washington, D.C., Chicago, Detroit, Dallas, and San Francisco. In Canada, write to 701 Evans Ave., Etobicoke, Ontario. Boole & Babbage products and services are also available in Europe, Japan, and South America.

BASIC FUNCTION: Computerized accounting system for OS and OS/VS IBM System/360 and 370 computers. It provides billing, makes programmers aware of the costs of their programs, and yields long-range performance measures for the system. It also operates on TSO systems, and a HASP option permits charging for cards in, cards out, and lines printed by the spooler. Basic SMS/CAS also provides user exits to SMF.

OPERATION: In use, SMS/CAS fulfills the four major objectives outlined in the Management Summary. It does this using three programs, Billing, STATS and Edit, which are also outlined in the Management Summary.

SMS/CAS makes one minor modification to the Wait/Post routine of approximately 30 bytes, in order to add a measure called Voluntary Wait Time to SMF data. Voluntary Wait Time is the time that a program voluntarily waits for an I/O operation to be performed. Then, each time a job or job step terminates, SMS/CAS executes the IEFACTRIT exit to produce in-line charges. In a batch load, then, the Edit program reads the SMF data and produces STATS and Billing files which are read by those two programs to produce reports.

Total elapsed time of each job or step is broken up into CPU time, Voluntary Wait Time, and Suspend time. Suspend time is the time when a program "would like" to execute, but can't because a higher-priority program has control of the CPU. Since this time is not under the control of the user's program, he shouldn't be, and isn't, billed for it. Also, since Voluntary Wait Time can have variances due to contention problems, Boole & Babbage has given SMS/CAS the capability for the user to limit each wait to a maximum specified at installation (e.g., 50 milliseconds).

Under this system, net computer time is the sum of CPU time and Voluntary Wait Time. With SMS/CAS, a new definition, Opacity, arises as the CPU time divided by the next computer time. Expressed as a percentage, Opacity is the "CPU-boundedness" of the step or job.

Some of the information that can be printed out at the end of a step includes: step name, program name, condition code for the step, time the step was started, time ended, ➤

SMS/CAS
Boole & Babbage, Inc.

▷ Periodic Billing, System Statistics. Program Usage, Tape and Disk Drive Hours Usage, Core Usage, IPL (Initial Program Load) Dates and Times, IPL's and other operations that bring tapes on- and off-line, and Space Available on Volumes.

SMS/CAS has been available since June 1972 in its real memory version, and since August 1973 in virtual memory versions. At this writing, there are 25 users, most of whom are also users of other Boole & Babbage SMS packages. Importantly, the package is one of very few that can provide charge-back accounting under TSO (Time-Sharing Operating system) on IBM System/360 and 370 computers.

USER REACTION

Users interviewed by Datapro reported no problems with SMS/CAS, either in installation or operation. What's more, vendor support for the package by Boole & Babbage was rated good to excellent by all users.

One user mentioned that he liked the on-line report on occupancy and EXCP's, and the fact that the package tells the utilization dollar value. Another regards the package as an exploratory tool for an eventual billing system, but also uses the SMS/CAS package to point to areas of operations and programs that should be further analyzed by PPE and/or CUE.

Regarding installation, no user reported taking more than half a day to install the package, except for one unusual installation (OS/MFT run under VM/370) that required six hours. □

► elapsed time, net seconds, dispatched priority, and Opacity for the step. Also listed can be memory used and amount of allocated memory not used. Any large core storage (LCS) usage is shown. Actual charges for the step are given. CPU time, Voluntary Wait Time, main storage usage, a main storage surcharge (amount over an installation-specified limit which the programmer has used), LCS usage, an LCS surcharge, EXCP's, Occupancy (in terms of the time a step has tied up a particular type of device), and usage of each type of device can all be charged for at different rates. There can also be a special charge for each priority and/or shift in which that the step was run.

Job-end information is basically a sum of all step-end information, with mounting charges added.

REPORTS: Periodic Billing Reports can be produced at any time by the batch Billing program. Credits can be applied, and account numbers can fall under up to three subgroupings.

System Statistics Report #1 breaks down SMF data by shift and gives information on the total number of jobs, CPU usage, net hours used, and main storage hours used. It also gives information on jobstream net time, average job net time, average number of partitions available and used, and more.

The Program Usage Report shows the programs executed, the number of runs, and the total net hours the program accounted for. Programs isolated as the greatest time-consumers can be further evaluated under a software monitor, such as PPE.

The Tape and Disk Usage Report lists drive hours on-line, drive hours used, and drive net hours for each drive. It also gives EXCP counts for each drive.

A Core Use Report gives a region history of the region size requested, the number of steps, the average main storage used, total CPU time per step, and Opacity. Another report gives the time and date of each IPL, and another gives a report concerning when each tape drive was either brought back on-line via a vary on-line message, or when it was brought back on-line via allocation recovery.

A report on each volume serial number lists the amount of space available in each volume.

HASP INTERFACE: This is a modification made to HASP to produce the SMF Type 6 record. The record includes information on cards in, cards out, and lines printed, and these items can be charged for in the Edit program, with the total printed out in the Periodic Billing Reports.

PERFORMANCE: According to users, SMS/CAS performs as it was represented to them, and without problems. No adverse complaints about excessive overhead or batch processing time were heard. Highlighting the system's ease of operation, some users run the Edit, STATS, and billing modules as a single job-stream, in about 200K bytes of storage.

HARDWARE/SOFTWARE REQUIREMENT: An IBM System/360 or 370 computer under OS/MFT, OS/MVT, OS/VS1, OS/VS2, or TSO. HASP systems can also use SMS/CAS plus the HASP option. The Edit program uses the IBM Sort program to process SMF data sets, and runs in about a 128K region. Edit output is processed by STATS in about a 136K region. Accounting information produced by Edit is processed by Billing in about a 90K region. The SMF Exit code requires about 4K, and on-line monitoring may require about an 8K addition to the initiator.

PRICING: SMS/CAS is generally sold on a purchase basis, but leases can be arranged. Under purchase, first-year maintenance is provided gratis, as is installation, documentation, and some training. Purchase and maintenance prices are as follows:

Product	Purchase Price	Annual Maint.
SMS/CAS	$6,000	$600
Edit & STATS source code	300	none
TSO option	1,000	250
HASP option	1,000	250

INITIAL DELIVERY: OS versions, June 1972; VS versions, August 1973.

CURRENT USERS: 25 as of November 1973. ■

SPOOLER
Boothe Management Systems

SPOOLER, a 1973 Datapro Software Honor Roll package, is a low-cost output spooling supplement for System/360 and 370 DOS. It can replace IBM's POWER, using only one-third as much memory.

MANAGEMENT SUMMARY

SPOOLER is intended primarily for use on small to medium-size IBM System/360 Model 30 processors SPOOLER is not a radically innovative package – its capabilities are closely similar to those of its no-charge IBM POWER counterpart – but it operates in only 4K or 6K bytes of main memory, while the IBM equivalent requires 18K to 24K bytes. For smaller DOS installations, this size difference is quite important, and can easily justify the use of SPOOLER instead of the IBM package.

SPOOLER considerably extends the power of DOS and helps to increase system throughput. Boothe maintains that an overall throughput improvement of 20 percent is typical, with some users able to achieve up to a 40 percent throughput increase over a DOS installation that does not have spooling. This enhanced DOS operation may result in reduced expenditures for extra-shift rental and staffing costs, and may permit postponement of a transition to a larger system on the grounds that the present DOS system is not powerful enough to handle an increasing workload. Twenty-one operator control functions allow the user to override the automatic first-in/first-out processing logic in SPOOLER.

The concept of spooling (Simultaneous Peripheral Operations On-Line) is not new. Widely used on second-generation large-scale computers, input spooling allows streams of jobs and/or data to be transcribed from card readers or other low-speed input devices to higher-speed peripherals such as magnetic tapes, disk, etc. A similar process places output intended for low-speed devices onto high-speed magnetic peripherals, for later conversion to the final output medium. This input/output processing is done simultaneously with other computing tasks. (Note that SPOOLER provides only *output* spooling, from intermediate disk storage to a printer).

Boothe Management Systems is a subsidiary of Boothe Computer Corporation, which has experienced serious financial difficulties during the last few years. The parent firm's losses have been in the tens of millions of dollars, and there has been a change of top management. SPOOLER was once marketed by Boothe Resources International, since dropped for its unprofitable operations. In October 1973, an agreement in principle was reached for National Computer Rentals Ltd. to acquire Boothe Computer, subject to approval by the stockholders of both firms.

USER REACTION

In any case, the SPOOLER customers interviewed by Datapro aren't worried. They are using software that is bug-free in recent memory, uncomplicated in nature, and

▷

CHARACTERISTICS

SUPPLIER: Boothe Management Systems, 555 California St., San Francisco, California 94104. Telephone (415) 989-6580.

BASIC FUNCTION: SPOOLER provides printer output spooling for a background batch job stream on an IBM System/360 or 370 under DOS. As user programs prepare output, SPOOLER places this output on a direct-access storage device, using previously reserved disk space. As soon as 2 main memory buffers (3 lines of output in the 4K version or 7 or 8 lines of output in the 6K version) are filled, SPOOLER begins to transfer the output to the disk. Printing may then be initiated at any time. Subsequent jobs can be processed while output from the first job continues printing. Following initiation of SPOOLER in the foreground, an Automatic Mode of Operation continues without intervention except for forms changes, clearing of jams, and operator scheduling of print priorities. A warm restart capability is provided in SPOOLER by an automatic checkpoint facility and the ability to backspace printer output to a specified place.

SPOOLER is a DOS program. As yet, no steps have been taken to adapt SPOOLER for DOS/VS, nor does the vendor state any definite plans to do so.

OPERATION: SPOOLER is initiated as a user job in the input stream through job control statements, and is available either in a basic 4K-byte or full 6K-byte version. Both versions are for operation under DOS only. Either version of SPOOLER occupies one of the two foreground DOS memory partitions (F1 or F2), leaving two active partitions for other user programs.

The major difference between the 6K-byte and the 4K-byte versions of SPOOLER is in the speed of operation, with the 6K version operating substantially faster than the smaller version. Otherwise the two versions offer the same facilities. These features, present in both versions, include the following.

- Early printer start, enabling output to be printed while the job creating the output is still running.
- All printer output buffered directly to previously reserved disk storage only.
- No changes to DOS or previously set up JCL for input job streams that follow normal conventions where each job is preceded by a/JOB card.
- Single print file support where one printer is driven at a time.
- Support of IBM 1400 emulation.
- Automatic restart.

➤

unlikely to require vendor support. Datapro has been consistently informed by users that the package has not needed vendor support, and that revisions received in the mail (the last was about a year ago) work perfectly.

SPOOLER is often compared to the most widely used and successful DOS spooling supplement, GRASP (Report 70E-760-11). Under certain circumstances, where input spooling considerations are not a significant factor in the job stream, Boothe claims that SPOOLER can consistently outperform GRASP by providing 10 to 15% more throughput in the same amount of time. Such circumstances are much more likely to arise in smaller installations such as the 360/30, and will seldom be found with larger systems. Of course, the more powerful GRASP package offers considerably more features than does SPOOLER—but at a cost that is typically about twice as high for a comparably sized spooler.

Datapro investigated a number of SPOOLER installations, including some that had given SPOOLER an extensive tryout before taking the system. Uniformly, the users reported that all SPOOLER claims are valid, and that operations with the package have been smooth and trouble-free. In most cases, the primary considerations for selecting SPOOLER were the reduced main memory requirement for capabilities similar to those of IBM's POWER, and the ability to get along without input spooling in return for a comparatively low price tag. SPOOLER offers a potential increase in throughput for the smaller IBM DOS systems, and should be scrutinized carefully by such installations. □

- **Support of UCS for printer output.**
- **Multiple output copies.**
- **Ability to backspace the printer file for recovery.**

No job accounting routines are provided, although a record of SPOOLER activity is maintained on the disk queue file midway between the SPOOLER file extents. This location minimizes disk arm movement and makes the accounting data available to the user through his own program.

Provision is made in SPOOLER to change forms prior to printing a job through the Modify Job command. The operator enters this command through the console and specifies the forms type and specific job. The command may be entered at any time for a job in the queue that has not yet begun to print.

Because no input spooling is provided with SPOOLER, those functions associated with partition balancing, user program relocatability, execution priority, and the use of cataloged procedures are not present. Furthermore, all buffering uses disk storage only, and no remote terminal support is provided.

HARDWARE/SOFTWARE REQUIREMENTS: A DOS IBM System/360, Model 22 and up, or System/370, Model 125 and up, with a multiprogramming supervisor, a minimum of 24K bytes of main memory, storage protection, line printer, card reader/punch, operator console, and either a 2311 or 2314/2319-type disk device. Typically, about 50 cylinders of a 2314/2319-type device are required when operating under SPOOLER. The basic version of SPOOLER requires a 4K-byte Foreground 1 or 2 partition, and the larger version requires 6K bytes in F1 or F2.

PRICING: Update service is provided for all systems without charge, including documentation. Standard SPOOLER enhancements are provided at no additional charge for all leased systems during the lease period, and for 1 year for purchased systems. Several technical support personnel based in San Francisco provide maintenance for SPOOLER through Boothe Computer Corporation facilities located in most major metropolitan areas.

	Monthly Rental	**Purchase**
Basic SPOOLER (4K-byte version)	**$175**	**$3,500**
Standard SPOOLER (6K-byte version)	**200**	**4,000**

INITIAL DELIVERY: September 1971; 6K version in March 1972.

CURRENT USERS: More than 45 in the U.S. and Canada to date. ■

DOSRELO
Boothe Computer Corporation

MANAGEMENT SUMMARY

DOSRELO is among the best of a considerable number of enhancements to IBM's DOS that were available prior to the announcement of DOS/VS by IBM in August 1972. With that announcement, much of the steam was taken out of the marketing efforts for DOS enhancements by independent software vendors.

DOS/VS (Release 28 of IBM's DOS) includes a self-relocation capability that supersedes the need for DOSRELO. Furthermore, the availability of DOS/VS at no additional charge on System/370 Models 135, 145, 155-II, and 158 tends to offer an almost irresistible incentive to current System/370 DOS users (who can make excellent use of DOSRELO at this time) to upgrade to DOS/VS. Coupled with the end of free support for DOS on System/360 or 370 computers by IBM as of March 31, 1973, a truly remarkable front of pressure is being applied to DOS users to give up DOS and/or the System/360 in favor of OS and/or DOS/VS on the System/370. (A full analysis of IBM's upgrade strategies can be found in the System/370 report, 70C-491-04).

On the other hand, for existing DOS installations, DOSRELO can provide a useful additional facility. Self-relocatability permits savings in the following areas:

- Machine time–DOSRELO eliminates the need to catalog the same program in the Core Image Library (CIL) more than once, for use in more than one partition, and similarly eliminates the need to recatalog the program when the supervisor or partition size changes.
- Programming time–self-relocatable programming is a complex and time-consuming effort if not done automatically.
- Operations–only one set of Job Control Statements per program is required, and job scheduling is greatly simplified since any job can go into any partition (space and peripherals permitting).
- Disk storage–DOSRELO saves Core Image Library (CIL) space since the program need be cataloged only once.

As a concept, relocatability (whether by "self" or by an operating system agency) offers a great potential for increased system resource utilization. Under unaided DOS, programs are non-relocatable in the core image library; i.e., they have been assigned to specific main memory locations. Thus, if one of the three partitions in ▷

DOSRELO provides a self-relocation capability for IBM's DOS by adding control sections to the user's program in a manner transparent to the user. No changes are made to the supervisor by DOSRELO, which adds a valuable new function to DOS.

CHARACTERISTICS

SUPPLIER: Boothe Management Systems, a division of Boothe Computer Corporation, Bank of America Center, 555 California Street, San Francisco, California 94104. Telephone (415) 989-6580.

BASIC FUNCTION: DOSRELO operates on the input to the Linkage Editor and the SYSLNK file (which is built by Job Control or the compilers) to make IBM System/360 or 370 DOS object decks self-relocating. Following this treatment by the DOSRELO utility, COBOL, RPG, PL/1, FORTRAN, or BAL programs can be loaded into any partition of DOS, thus providing the operator (or automatic scheduler) with the flexibility to use any available DOS partition for a self-relocating object program at run time (provided the partition size and peripheral availability are adequate).

OPERATION: DOSRELO adds entry-point logic in front of the object code for a COBOL, RPG, PL/1, FORTRAN, or BAL program before the program is cataloged on the Core Image Library (CIL) by the Linkage Editor. It also adds a control section to the end of the program that includes data describing the position and size of the address constants that are to be relocated. The resulting self-relocatable program is independent of the physical address(es) in main memory in which it will function.

At execution time, the problem program is identified by a"// EXEC phase" statement in the job control stream, and control is transferred to the DOSRELO control section. The control section determines the offset amount by which the address constants in the program will have to be adjusted and resolves the actual program addresses to the current main memory location. At that point, control is transferred to the user's program logic and execution proceeds. Thus, the execution of user programs made self-relocatable by DOSRELO is transparent to the user except for the small amount of time and space consumed by the address resolution process.

Among the specific limitations placed upon the usage of DOSRELO are the following:

- The relocatable library modules may not contain common control sections, PHASE statements, or unnamed control sections if they contain ADCONS.
- All DASD files referenced by DOSRELO must be of the same type–either all 2311's or all 2314's.
- FETCH statements must be changed to LOAD statements.

➤

DOSRELO
Boothe Computer Corporation

▷ a DOS multiprogramming environment is free, but not the one for which programs awaiting execution have been cataloged, that partition (and share of the system resources) must remain idle. Relocatability allows a program to run from any location in main memory, provided that enough space and peripheral devices are available. Larger systems with plenty of peripherals naturally have more to gain from relocatability, since there is more room to maneuver from a scheduling point of view, and also because the potential for wasted resources is greater.

In addition to these general self-relocatability considerations, DOSRELO requires no changes to the user program or the supervisor and supports the COBOL, RPG, PL/1, FORTRAN, and BAL languages. Users of DOSRELO contacted by Datapro report that the package performs as advertised. Certain limitations on multiphase (or modular) programs cause some inconvenience, however, particularly with respect to the COBOL compiler, which cannot be made self-relocatable by DOSRELO. These limitations are discussed in the Characteristics section of this report. Otherwise, user feedback is quite positive.

Until mid-1973 DOS/VS will remain a "paper tiger," and DOS users of either a System/360 or 370 can well profit from a DOS self-relocatability enhancement until then. Subsequent to the free availability of self-relocatability in DOS/VS, the market for DOSRELO will be pretty much confined to a dwindling list of System/360 DOS users–but their ranks currently number some 10,000.

Other independent sources of self-relocatability are available–both as standalone packages and as combinations of multiple DOS enhancements–and the DOS user will be well advised to investigate a number of alternative vendors carefully before making a decision regarding the future of his operating system. Of particular note in such an investigation are the projected plans of the vendor. Look for the possibility of price reductions and other usage concessions as a gambit for the suppliers of DOS enhancements to gain new business opportunities in your computer installation. This is a particularly good bet now, when it seems likely that comparatively few additional sales of such packages will be made to the general marketplace in the next year, and still fewer after that. Careful shopping for DOS enhancements now could produce significant short-range payoffs. □

➤ The features of DOSRELO include:

- Normal Entry Point Override to change the normal phase entry point.
- Common control sections permitted in problem programs.
- Twenty-nine diagnostic messages to aid in running DOSRELO.
- Automatic handling of multiphase (or modular) programs, except that the load address for COBOL overlay routines must be determined by the user and specified to DOSRELO.
- Immediate aborts in response to fatal compile errors from a previous job step, rather than wasted effort by DOSRELO trying to massage a "dead" program.

DOSRELO itself is a self-relocating program.

HARDWARE/SOFTWARE REQUIREMENTS: DOSRELO can be installed on any System/360 or 370 under DOS Release 26 or lower that includes the Linkage Editor, loader facilities, and Multiprogramming (MPS) option supplied by IBM. The minimum main memory requirement for DOSRELO is 16K bytes. This partition size is large enough to accommodate a monolithic or modular program with a total of 48 INCLUDE cards and entry points, 16 control sections per phase, and 160 to 320 address constants. DOSRELO automatically expands it internal tables to take advantage of all available main memory, however, and each additional 2K bytes of storage allows the user program to be increased by 32 INCLUDES, 24 EXTRNS, 16 entry points, 8 CSECTS, and 80 to 160 ADCONS. The actual size of the control sections appended by DOSRELO is 248 + 8A + 4B bytes, where A and B are the number of control sections and address constants contained in the user's object program, respectively. (If sort exit linkage is required, the control section is increased by 160 bytes.)

PRICING: DOSRELO is available for purchase or on a monthly rental basis. Maintenance is included at no additional charge. Prices are as follows:

		Purchase	
	Monthly Rental	Class A*	Class B**
First location	$195	$2,500	$2,500
Second location	195	2,000	2,250
Third location	195	1,500	2,000
Fourth location	195	1,000	1,750
Fifth location	195	750	1,500
Sixth location	195	500	1,250
Seventh location	195	500	1,000
Each additional location	195	500	500

*Class A plan provides service by Boothe through a single central location.

**Class B plan provides individual service for each location.

INITIAL INSTALLATION: December 1969.

CURRENT INSTALLATIONS: About 40. ■

Fastball-76
Brown Brothers Enterprises

MANAGEMENT SUMMARY

Fastball-76 is a limited-usage COBOL preprocessing system whose low cost and potential for operational savings tend to outweigh its minor faults and limitations in the opinions of many users. It can be used to significantly reduce a programmer's efforts in producing report and summarization programs or file utilities. It is claimed that unmodified programs generated by this shorthand language can provide up to half of the COBOL programs required in typical installations.

Fastball-76 can save programmer time, keypunching time, and debugging time (and thus, computer time). It can also reduce turnaround time. The program generally punches 8 to 16 COBOL source cards for each Fastball card read. Assumptions made by the preprocessor eliminate almost all of the coding normally required for each of the four COBOL divisions.

Fastball is limited as to the number of files that can be processed by any one of its generated programs, but users commonly skirt this difficulty and produce multi-file programs by simply combining several of the programs output by Fastball-76. A corollary use involves employing Fastball-76 to generate the tedious portions of major programs, with the understanding that the source code produced by the package will be modified and combined.

All that is necessary to use Fastball is that the person desiring a job to be done by the computer fill out a self-explanatory Fastball transmittal sheet, have it keypunched, and execute Fastball. One-shot programs can often be turned around in less than an hour in this manner. A side benefit is that the entire COBOL program is written with uniform standards of alignment, punctuation, and terminology. Fastball can produce many such COBOL source programs in a batch because it is a stacked-job program that requires no re-link-editing or re-executing between runs.

The package is coded in COBOL and supplied in source-deck form. For installations as small as 32K, Fastball is supplied as two programs, one for 2-file and another for 3-file programs. Users with 65K or more bytes of main memory receive one program that handles both situations. Sales and service are handled through the mails.

Brown Brothers Enterprises is a diversified firm that was founded about five years ago.

USER REACTION

The 1973 Datapro survey of proprietary software users turned up two Fastball users, who rated the package ▷

Fastball is a low-priced shorthand language and preprocessor that can generate complete 2- and 3-file COBOL source programs with minimal programmer effort. It runs on IBM System/360 and 370 computers and has also been adapted to Burroughs B 5500 COBOL.

CHARACTERISTICS

SUPPLIER: Brown Brothers Enterprises, 4400 South Division Ave., Grand Rapids, Michigan 49508. Telephone: (616) 532-9079.

BASIC FUNCTION: Production from shorthand input of error-free 2-file and 3-file COBOL source programs on punched cards on a one-statement-per-card basis. Basic program types that can be generated using Fastball-76 are reports, summarizations, file creators, file copiers, indexed sequential file reorganizers, file updates, and record reorganizers.

OPERATION: The user fills out a Fastball transmittal sheet, has it keypunched, and executed Fastball.

The first card identifies the author, program, and report title. Additionally, four one-character entries specify generation of coding to support an interrupt routine, generation of appropriate Working Storage and Procedure Division statements to support an UPSI (User Program Switch Indicator) routine, generation of the linkage necessary to link to an assembly-language program and retrieve the data for use in a report from the communications region, and generation of a grand total routine.

The second card defines the input and output files.

The next four cards provide report headings. Sixty-six characters on each card are used, and each pair of cards is concatenated to create one 132-character line. Zero, one, or two-line headings can thus be generated.

Up to 29 following cards provide standard-format data name and picture statements, plus usage, total, and key information. Also often coded, but not necessarily keypunched, are indicators as to whether the data is part of a sending or receiving field item.

Finally, up to nine cards indicate record selection logic. This has the function of replacing the Procedure Division. AND, OR, NOT, less than, equal to, and greater than are selection criteria that can be used.

Fastball is executed as a stacked job. Multiple COBOL programs can be produced without re-link-edits or re-executes between runs.

PERFORMANCE: Simple programs can often be written within 15 minutes using Fastball. In programs of this type, most of the Identification Division and all of the Environment Division except for file names and indicators of file type need not be coded. Data Division coding is reduced by elimination of the need to code the words FILLER and PICTURE and periods, along with much ►

▷ good or excellent with respect to their overall satisfaction. Datapro contacted four other users and got the same type of reaction and comments. Basically, the users feel that the package is a bargain in terms of the reduction in number of errors and programmer time it yields—so much of a bargain, in fact, that these users stated openly that for the price they're paying, they don't expect better diagnostics, better service, or edited input.

Typical user comments indicate that the package is easy to learn and is generally accepted by all but the most pedantic programmers. Users stressed that the package makes it practical to code simple reports in COBOL rather than in RPG. One user maintains that, except for simple programs, one should not expect to directly apply the output of Fastball; instead, this user says that the package reduces or eliminates the "dog-work" from COBOL coding, and has enabled him to produce complex 6-file programs with a minimum of errors by combining modified COBOL source programs generated by Fastball.

Datapro feels that Fastball's low price and high utility combine in such a way as to make it appropriate for many users to investigate the package's use in their own installations. □

➤ other Data Division coding. The Working Storage section is completely eliminated for most standard purposes.

Up to 90% of the keypunch time associated with COBOL program production can also be eliminated for these simple jobs, with Fastball creating 8 to 16 source cards per input card. Standardization is an automatic by-product.

The limitation of two or three files per Fastball-generated program is not a severe one, since multi-file programs can be created by combining the outputs of several Fastball generations. Likewise, Fastball can be used to generate portions of large, complex programs. The chief drawback of the package is that its input is unedited. Indeed, input errors can cause generation of programs that won't compile or, worse, of programs that loop or have output errors. "Garbage-in, garbage-out," as the saying goes.

HARDWARE/SOFTWARE REQUIREMENTS: Fastball-76 requires a COBOL system with at least 32K bytes of memory, a disk drive, card reader, and card punch. The disk requirement is two cylinders of IBM 2311-type space or its equivalent, which is used as a permanent indexed sequential work file. For systems under 65K, two Fastball programs are supplied, one for 2-file programs and another for 3-file programs; either will run in 24K. Systems of 65K and up receive one Fastball program. Fastball is provided in COBOL source-deck form. It is currently in use on IBM System/360 and 370 and Burroughs B 5500 computer systems.

PRICING: Fastball-76 is provided on a 15-day trial, after which it can be acquired on a permanent license for $785. Supplied with the package are transmittal sheets and supporting documentation. Training is not provided, and sales and service are handled by mail.

INITIAL DELIVERY: January 1971.

CURRENT USERS: Approximately 75 as of August 1973. ■

CROSSTABS
Cambridge Computer Associates

MANAGEMENT SUMMARY

Information is relatively easy to collect in these days of punched cards, magnetic tape, and on-line disk storage. And it's not a major problem to get the computer to cough up great quantities of this information – but *interpreting* the information is another thing entirely.

Conventional EDP applications deal with problems where detail information is printed out; payroll checks, bills, input and output registers all require attention to each and every item in the file. But these applications generally make no provision for recognizing patterns and identifying data relationships or trends. For example, knowing the percentage of the U.S. adult population that drinks instant coffee may be far more valuable than knowing the individual names of each coffee drinker.

CROSSTABS (Cross Tabulation) is an important tool for establishing patterns and data relationships. It counts events within combinations of categories specified by the user. A cross-tabulation is essentially a table showing the number of occurrences of each combination of categories included in a report.

CROSSTABS results are not restricted to just two dimensions. Essentially any number of variables (groups of related categories) can be established. To present these when restricted to the two dimensions available on a sheet of paper, CROSSTABS simply produces many pages, each showing two variables and a particular combination of categories for the other (filter) variables. Multiple table requests (each with any number of variables) can be submitted for a single CROSSTAB production run.

The frequency distributions can be weighted by information internally contained in the source record, or externally supplied by the user. Arithmetical and logical combinations can be used to create variables not explicitly represented in the source data. The use of weighting factors can produce a more accurate representation or compare the results to related statistics. Up tc five different weighting factors can be specified per cell. Various statistical measures of independence and association (such as Chi square) are also available to the programmer. For still more sophisticated statistical procedures, linkages can be set up to user-written subroutines.

CROSSTABS's basic output format, although about as good as can be obtained with a computer line printer, requires that anyone reading it be generally cognizant of the CROSSTABS format. If fully applied, however, the labeling features of CROSSTABS can explain the table contents down to the finest detail.

CROSSTABS is a tabular statistical system that presents frequencies of events within combinations of categories in order to clarify relationships among these categories. A powerful Retrieval Option adds an RPG-like capability to the standard output format. CROSSTABS is available for IBM System/360 or 370 computers under DOS or OS.

CHARACTERISTICS

SUPPLIER: Cambridge Computer Associates, Inc., 222 Alewife Brook Parkway, Cambridge, Massachusetts 02138. Telephone (617) 868-1111.

BASIC FUNCTION: To display, by counting occurrences, relationships among categories of variables. The standard output consists of multivariate frequency distributions with weighting factors, all relevant percentages, and bivariate statistics organized into completely labeled tabular reports. Many weighting and statistical options aid in identifying and measuring relationships. Versions of the package are available for IBM System/360 and 370 computers operating under DOS or OS.

OPERATION: The basic mode of output presentation is a two-dimensioned table with rows and columns. Each individual row or column represents a category of a particular variable, and all the columns or all the rows in a table represent a particular variable. The intersection of a column and row is a "cell." Many different pieces of information can be accumulated for each cell, and each cell can thus consist of many lines.

Additional "filter" variables can be accommodated. A separate two-dimensioned table called a "plane" is produced for each filter variable. Each plane may extend to more than one page both horizontally and vertically. The collection of planes representing one output is called a "table." Titles with up to 360 characters explaining each plane or table can be printed on each page.

CROSSTABS input consists of records each containing information about one event. In the simplest case, the data value or code in each field (variable) represents one category of that variable. Other more complex variables can be constructed by arithmetical and/or logical combinations of the input variables. Records containing data items not in the range of interest can be excluded.

Weighting schemes are also available in CROSSTABS to correct distributions within the data set to those within a known population in order to give results more accurately reflecting relationships in the total population.

There is no limit to the number of tables that can be produced in a single run. A table must be held in memory until the data set is completely passed. If the memory size is insufficient to hold all tables, additional passes are automatically taken. The user can specify a precise order for the presentation of multiple tables; or, by default, CROSSTABS will optimize the order to reduce the number of passes required.

CROSSTABS
Cambridge Computer Associates, Inc.

▷ A Retrieval Option allows users to select and write out some of all of the raw data that was input to CROSSTABS. The user may reformat or "enrich" these inputs and then generate several RPG-type outputs in the same run – either as formatted list-type reports or as separate data sets for subsequent processing. This ability to capture, rewrite, and present detail information is a logical complement to the analytical and summary function of CROSSTABS, since the user is provided with both the aggregate statistics from his input file and the detailed data that generated those statistics.

Another CCA product, AUTOGRAF, is available as a compatible system to CROSSTABS and can be used independently or together with CROSSTABS to produce graphical representations of a CROSSTABS run.

CROSSTABS is employed at many levels by its users. Of those contacted by Datapro, some have never encountered requirements beyond the simple report stage and are quite content. (In its programming manual, Cambridge Computer Associates has defined a subset of the language designed to provide basic capabilities and to minimize the learning time.) Others use the full range of capabilities, in areas such as market research and operations analysis.

While few statistical systems are capable of accumulating any appreciable amount of information during the passing of data, CROSSTABS is geared toward counting the frequency of combinations. When properly used, CROSSTABS is a powerful tool for making sense out of seemingly unrelated data. It is a very useful statistical reporting system with an easy-to-learn command language, and is currently saving a lot of one-shot programming work for many of its users.

Two problem areas were reported among the CROSSTABS users surveyed. First, the summary nature of the output often is not immediately readable to the non-experienced user unless each print-out page has been framed with explanatory labels, Second, because the accumulating tables are resident in main memory, very large runs generally have to be broken into a number of smaller runs.

Nevertheless, CROSSTABS effectively performs functions that are otherwise very difficult to implement, and should be given serious consideration by users with a need for statistical analysis. □

➤ **Of particular interest is the CROSSTABS Retrieval Option. This option allows the user to complement the usual output of statistical tables with reports or files containing some of the detail information behind the statistics.**

Temporary internal tables can be created and their printing suppressed; this technique permits combining two tables to accumulate information not obtainable directly from a single table. The basic CROSSTABS calculations are the raw and weighted frequencies. Up to five weights can be specified in the OS version (but only one in the DOS version). The user can also call for mean, highest, lowest, range from highest to lowest, sum of the squares, and the standard deviation. Raw frequency can be expressed as a percentage of the row, column, plane, or table weighted totals. In all, a total of 96 different cell options are available; in each table, up to 62 items can be requested and printed for each cell.

Any of a dozen statistical measures (such as Chi square, Lambda Statistics, and Cramer's V) pertaining to the degree of independence and association of the row and column variables within one plane can be requested. The user can also use his own routines for calculating statistical measures through subroutine linkages.

The principal limitations of the DOS version of CROSSTABS are less flexibility of input and output. Input can be on tape or disk only, and fewer output options are available.

HARDWARE/SOFTWARE REQUIREMENTS: The DOS version will run on a 32K IBM System/360 or 370 configuration, and the OS version requires about a 100K partition or region under OS/MFT or MVT. Additional main memory permits more and larger tables to be accumulated with fewer passes of the source data set, thus speeding up the processing. The average CROSSTABS run takes about 100K to 150K bytes according to current user experience.

PRICING: Both versions are available for rental on a month-by-month basis, or for a term of one to three years – with prepayment of the term rent yielding discounts of 15 to 45 percent. For both versions, software maintenance, documentation (five sets), and updates are included in the rental price.

The monthly rental price of CROSSTABS I (DOS) is $350, and the additional cost of the Disk-Input Option is $50/month. The three-year lease price, by prepayment, is $6,930 for the basic package and $990 for the Disk-Input option.

Monthly rental for CROSSTABS II (OS) is $500; options such as the Multi-Punch Option ($75/month) or the Retrieval Option ($100/month) are added to the basic rental. The three-year lease price of the basic package is $9,900 by prepayment. Training and installation cost $250 per 1-day class plus travel expenses.

INITIAL DELIVERY: Early 1968 for both versions.

CURRENT USERS: About 80 OS users and 20 DOS users. ■

AUTOGRAF
Cambridge Computer Associates, Inc.

MANAGEMENT SUMMARY

Using AUTOGRAF, professionals without an EDP background can use a computer system to draw graphs on a standard line printer. The AUTOGRAF language lets the user do this by having him supply only the basic information as to where the data to be graphed can be found, and by using a flexible set of AUTOGRAF commands to specify what the graph should look like. Graphs can be an important addition to the usual type of computer-generated reports, since they present a picture of summarized data that can help the manager reading the reports to visualize relative data values. Additionally, one type of graph that AUTOGRAF can produce, the frequency distribution point plot, can help to summarize data.

AUTOGRAF can produce profile graphs (the ones that look like a city's skyline), bar graphs (in vertical or horizontal format), and point plotting graphs on a standard line printer. Cambridge Computer Associates, the vendor of AUTOGRAF, consulted professional designers when it created the package to ensure that its output would be of the highest quality, and that, for the broadest possible range of users, AUTOGRAF would be: (1) highly general with regard to input data requirements, (2) simple to use, even for the most sophisticated applications, and (3) capable of producing graphs that are attractive enough to use for publication, yet inexpensive enough to be used for rough drafts.

In addition, AUTOGRAF has language features that enable it to access and use the data set output of CROSSTABS, another Cambridge Computer Associates product that is described in Report 70E-123-01.

The graphs produced by AUTOGRAF can present data in up to four dimensions: the standard X and Y, Z (using different plotting symbols and a legend to explain them), and P (for planes of data, or overlayable graphs). An AUTOGRAF program of less than 20 cards can produce, for example, a titled, labeled, scaled bar graph with 3 strips of different symbols in each bar, and a legend explaining the symbols in the strips.

AUTOGRAF has special features that enable users to obtain the best type of graphic output for their particular applications without spending excessive time and money in trial-and-error sessions. The extensive set of AUTOGRAF default options lets the package virtually design the graph for the user. Comprehensive error diagnostics help pinpoint errors during a run. Graph specifications can be checked out by using a sampling procedure for, say, the first 1000 records or 1 minute of CPU time, whichever occurs first, or perhaps for every 100th record in a file. Formatting options let the user position several graphs on a page or extend a graph vertically over several pages.

The AUTOGRAF language is straightforward and easy to learn and use. (Of course, one mustn't tell non-EDP-oriented professionals that they're programming, lest ▷

Using an IBM System/360 or 370 under OS or OS/VS, a user with little or no computer background can use AUTOGRAF to produce fully labeled bar and profile graphs and correlation point plots on a standard line printer. With a little practice, complex multi-variable graphs can be produced.

CHARACTERISTICS

SUPPLIER: Cambridge Computer Associates, Inc., 222 Alewife Brook Parkway, Cambridge, Massachusetts 02138. Telephone (617) 868-1111. CCA also provides system analysis, documentation, and programming services, all on a world-wide basis, handled from the Cambridge office.

BASIC FUNCTION: Presentation of data summaries in graphic form. AUTOGRAF enables personnel without an EDP background to use a computer system to produce profile, bar (horizontal or vertical), or point plot graphs on a standard line printer. Complete formatting, scaling, and labeling facilities are provided, as is the ability to present Z-axis information (through the use of plotting symbols and an associated legend) and multi-plane graphs (P-axis presentation). Also inbedded in AUTOGRAF are certain computational capabilities, the ability to use CROSSTABS output data sets (see Report 70E-123-01), diagnostic and error messages, sampling and other try-out facilities, special formatting options for various effects, and default options that considerably aid inexperienced users.

OPERATION: AUTOGRAF must be provided with two things: basic file information, so that it can obtain the data to be presented in a graph, and at least a minimum amount of specifications as to how the graph should look. Both of these items are supplied by the user, in AUTOGRAF commands.

Data can be provided by an explicit user reference to a field in his records. Optionally, the user can perform record selection tests, specify values to be computed from certain data fields, or have AUTOGRAF invoke a user-supplied program to obtain those values. Additionally, CROSSTABS users can refer directly to predefined fields in CROSSTABS output data sets, such as WTD, RAW, and COL. (These are the other CROSSTABS-produced data items that AUTOGRAF can access: TABLE, PLANE, ROW, MAX, MIN, WSQ, and NZW. The "W" in a function's name refers to the CROSSTABS-assigned weight.)

AUTOGRAF can also perform computational functions upon fields supplied by CROSSTABS. These are: MEAN (arithmetic), RANGE (difference between largest and smallest operands in a cell), and SD (standard deviation, for values of indicated weight within the designated cell).

AUTOGRAF can apply a transform function against data in accordance with a stated rule. It can also compute the logarithm or antilogarithm (base *e* or any other specified base, or defaulted to base 10) of data, or express computed data as the percent one data item is of another. The mathematical value of the constant *e* is taken to 5 decimal places.

In terms of format, the user can specify the type of graph (point-plot, bar horizontal, bar vertical, or profile), the title, axis, plane, and legend descriptors and labels, footings, scale, and so on. These controls are quite straightforward, and are defaulted to commands that ease the task of specifying graphs for novice users. At the same time, the commands that control the appearance aspect of the graphs ➤

AUTOGRAF
Cambridge Computer Associates, Inc.

they become frightened—either of the machine or of losing status. And programmers must come to accept the idea that a computer *can* draw pictures, without a plotter or mind-boggling programming.) Once the graphing concepts are accepted by users, they might tend to recall their high-school algebra and wonder "what's so great about a computer plotting graphs?" and "is that all it can do?" Well, there's more: AUTOGRAF can use built-in or specified rules and transform functions, as well as performing the computational functions of logarithms and antilogarithms (to any base) and percentages in creating graphs.

Combined with the facilities for heading, footing, labeling, scaling, and otherwise formatting graphs, the total AUTOGRAF facilities present the user with an impressive capability for presenting data as information in graphic form.

USER REACTION

Overall reaction to AUTOGRAF is quite good, with its actual users more emphatic in praise of the product then EDP department personnel. This is to be expected, though, since it is the former who benefit the most and oftenest from AUTOGRAF, while the latter quite naturally guard every precious second of CPU time and byte of main memory.

A user with two large 360's (a Model 75 and a Model 91) praises AUTOGRAF's "graphic sense" and computational algorithms, and states that the graphs "read well." This user also feels that AUTOGRAF features extreme ease of use and simplicity in training personnel for its use. On the other hand, the same user's EDP department voices the feeling that the package uses too much main storage (170K bytes at their installation), and "is no speed demon." They also suggest that the interface to CROSSTABS could be improved so that AUTOGRAF could graph percentages from CROSSTABS output data sets. (But they concede that this could be a problem with CROSSTABS, not AUTOGRAF.)

Another AUTOGRAF user, this one with a 370/168, likes the CROSSTABS interface and also praises the package's ease of use. This user feels that the package runs efficiently (and is thus not prepared to comment on the amount of time a run may take). At this installation, AUTOGRAF uses a nominal 200K bytes of main storage, and the user notes that AUTOGRAF can avail itself of the PL/1 GETMAIN facility if more main storage is required.

One highly sophisticated user (with multiple System/370 computers and some 27 leading software packages, all mentioned in DATAPRO 70, installed) states that he likes AUTOGRAF because it generates graphs on line printers without a plotter or fancy programs, because of the CROSSTABS interface, because non-EDP-oriented professionals can use it easily (he has about 75 of them using the package), and because JCL can be made transparent to the users. This user provides an average of about 4 hours of training in the AUTOGRAF language to each user. He runs the package in 150K bytes and feels that its performance is "reasonably efficient."

Installation of AUTOGRAF was no problem for any user interviewed; the average user had it up and running in less than an hour, with no problems showing up in the initial test run. The EDP staff just link-edits the load modules supplied (the source language is PL/1). The users give Cambridge Computer Associates high praise for the service it supplies, the quality of its documentation, and the competence of its staff. All users also state that AUTOGRAF performs all the functions that it is advertised to perform. □

can be employed by more sophisticated users to create a wide range of effects. Graphs can be spread over several pages in a vertical direction, arranged in planes for overlaid graphs, etc. For example, special symbols can be created on a graph, such as an overprinted 0 and + to create ⊕. On point plots, these overprinted symbols can be used to indicate items in different categories occurring at the same coordinates (i.e., apples and cherries, both blooming in April in Washington, DC).

In addition to control over data and format, the user also has control over execution. To test a graphing program or its output, the user can sample every *n*th record, graph only the first *n* records, or run the graphing program for only a specified amount of CPU time. Error messages are clear and ample in number; and unless directed otherwise, AUTOGRAF always tries to produce a completed run.

OS JCL statements necessary to run AUTOGRAF are included in the user's manual, but can be incorporated into a cataloged procedure so that JCL becomes transparent to the user.

AUTOGRAF is written in PL/1 and supplied in load module form on magnetic tape.

PERFORMANCE: According to users, AUTOGRAF performs all of the functions it is advertised to perform, and is "clean" in its current version. While opinions as to its speed and usage of main storage differ, these are relative considerations. Users seem to agree that AUTOGRAF runs are efficient, with opinions on the degree of efficiency ranging from "reasonably" to "highly."

HARDWARE/SOFTWARE REQUIREMENTS: An OS or OS/VS IBM System/360 or 370. About 150K bytes of main memory can suffice, but larger graphs can require more. To get the memory it needs, AUTOGRAF can use the GETMAIN facility of PL/1.

PRICING: $350 per month on a month-to-month basis, or by rental prepayment of $3,570 for one year or $6,930 for three years. Maintenance and user support are included in the rental. AUTOGRAF cannot be purchased or perpetually leased.

INITIAL DELIVERY: January 1972.

CURRENT USERS: 10 as of December 1973. ■

UTILITY-CODER/360
Cambridge Computer Associates, Inc.

MANAGEMENT SUMMARY

UTILITY-CODER/360 facilitates data manipulation, reporting, listing, and data utility functions on DOS, DOS/VS, OS, and OS/VS IBM System/360 and 370 computers, and permits users to validate, edit, recode, and reformat data more quickly than with IBM's standard utility programs. It also allows character manipulation and string operations.

UTILITY-CODER/360 from Cambridge Computer Associates may well be the leader in its class as far as facilities provided to users and processing performance goes, but it also carries a substantially higher price tag than its less capable competitors. Consequently, in evaluating UTILITY-CODER/360, you must investigate its capabilities and how they can be applied in your installation, rather than simply questioning how the price of this utility language package compares to some others.

In terms of its processing speed and efficiency, UTILITY-CODER/360 is rated by experienced users to be as fast and powerful as assembly-language coding, except for computational functions. Moveover, UTILITY-CODER/360 is a language with a simple, natural syntax and appearance that can be used by non-programmers, thus realizing significant savings in "people overhead," or programmer costs, for its users.

UTILITY-CODER/360 has broad-ranging facilities not found in less expensive utility and "one-shot generator" packages. It can, for example, handle about a dozen I/O units in the same run, and can read and write on the same tape or disk unit in a single run. Illustrative of its capabilities, perhaps, is the fact that one user has used UTILITY-CODER/360 as a tool to create a compiler (in UC/360 language) that is an effective, in-house, tailored report program generator. The user plans to present his innovation at the next CCA User's Group meeting, so the concept may well soon be documented as a new (perhaps optional) part of the package.

Summarized, the following are the uses for UTILITY-CODER/360 that CCA touts in addition to the ability of the package to serve general data manipulation operations: file creation and maintenance, detailed editing or recording of data, retrieval, data evaluation and reduction, report generation and special-purpose printing, routine data transcription, and data conversion and translation.

In addition, UTILITY-CODER/360 can be called by other programs and can itself call other programs. Thus, its facilities can be added to those of the user's COBOL compiler, for example, and UTILITY-CODER/360 can call and use, say, the IBM sort program.

One of the reasons that UTILITY-CODER/360 is rated as a good performer is the efficiency with which it handles I/O. Scanning a file, for instance, involves only one I/O instruction for the entire operation when UTILITY-CODER/360 is used, whereas the IBM utility program would use one such instruction (EXCP) per record scanned. This is because UTILITY-CODER/360 can chain I/O ▷

CHARACTERISTICS

SUPPLIER: Cambridge Computer Associates, Inc., 222 Alewife Brook Parkway, Cambridge, Massachusetts 02138. Telephone (617) 868-1111. CCA consulting, documentation, and software are all supplied on a world-wide basis from this location.

BASIC FUNCTION: Data manipulation, reporting, and utility functions on IBM System/360 and 370 DOS, DOS/VS, OS, and OS/VS systems. UTILITY-CODER/360 is a simple and efficient language that can be used by either programmers or non-programmers. It allows users to validate, edit, reformat, and recode data more quickly than IBM's standard utility programs. It can also manipulate character and string data, and is particularly well suited to the preparation of raw data for processing. It is also useful in generating reports and listings. The package differs significantly from other independently vended utility packages in that it has a much more comprehensive and flexible syntax and can handle multiple simultaneous input and output files; it can also call other programs and be called by them. The OS version can be enhanced with options to permit processing of multipunch column binary data, and to allow invocation of the IBM OS Sort.

OPERATION: UTILITY-CODER/360 is supplied as a single object module and installed in the usual fashion. Its use can be self-taught by nearly any professional from the quality documentation provided.

The UC/360 processor organizes main storage into several areas, four of which are given over to the discretion of the user. These four areas are input, output, save (or work area), and counter. The user can refer to data in the first three of these areas either by the position of a field within the area or by a variable number assigned to the field. The data is stored internally as EBCDIC characters to facilitate rapid manipulation of character data for basic utility tasks, and is thus not in a format best suited for arithmetic operations; since UC/360 programs can be linked with external routines written in any System/360 language, extensive computations can be accomplished in other languages without losses in efficiency.

The flow of processing in a UC/360 program is normally sequential, with overriding control given to a truth indicator that registers the results of every user's test in the program. At the end of a UC/360 program, the END statement loops back to the beginning. Thus, the process is reiterated until, normally, the input is exhausted or a user-specified limit is reached. ►

commands together, whereas the IBM utility program does not.

USER REACTION

UTILITY-CODER/360 is a favorite with all of the users interviewed by Datapro. All report that it is very easy to learn and use (many have clerical personnel using it regularly), and that it greatly reduces the "people overhead" in creating "one-shot" programs to get at data. One user reports that UTILITY-CODER/360 is almost too good; it keeps creeping into production use at his installation. He doesn't mind this too much, because the programs are quite efficient and were written by his programmers without wasting time, but programs that programmers write as one-shots tend not to be documented.

After all, the programmer must be thinking that the program will be used only once, so why bother to document it? This is no reflection on UTILITY-CODER/360 itself, because it has the facility to include line-by-line self-documentation in its listing. What happens, though, is this: A programmer once created a UTILITY-CODER/360 program to clean up a set of files in order that program A would run. Somewhere along the line, another person discovered that it's always a good idea to run the files through that UTILITY-CODER/360 program before running production program A. But the UTILITY-CODER/360 program to clean the files was never documented.

The moral here is "UTILITY-CODER/360 is fine for production use, but be sure to document the programs." One satisfied user routinely uses UTILITY-CODER/360 to modify keys prior to special sorts, for example.

Some UTILITY-CODER/360 users have found that the package links nicely to CCA's CROSSTABS and AUTOGRAF (Reports 70E-123-01 and 70E-123-02). All report that UTILITY-CODER/360 has never failed to perform the functions claimed for it in the user's manuals.

UTILITY-CODER/360 is supplied in the form of a single object module, and users say it can be installed with no trouble whatsoever. The JCL necessary to run a UTILITY-CODER/360 program can be taught to users, left to system programmers, or cataloged.

In the opinions of all UTILITY-CODER/360 users interviewed, Cambridge Computer Associates provides software, service, and documentation of the highest calibre. □

The UC/360 program as written by the user consists of a Declarative Section and a Narrative Section. In the former, the user: (1) assigns logical unit designations to input and output data sets, (2) reserves any required save area, and (3) specifies execution time parameters, such as a limit on processing time, printing line output, input records read, etc. In the Narrative Section, five kinds of processing – input/output, test, data manipulation, logic flow, and diagnostic – are performed upon the data that has been specified either as literals, as the contents of fields in the user main memory areas, or as special-purpose variables.

The UC/360 language consists of English words, abbreviations, and some symbols. The syntax is simple, and can be diagrammed as on a COBOL reference card. The elements of the language are virtually self-explanatory, and words that are ignored by the UC/360 processor can be added into the coding for additional clarity. Some of the language elements have synonymous forms that programmers can use as they see fit. The two parts of the UC/360 program are separated by a dollar sign.

Statements available to the Declarative Section are used to define processing requirements, formatting, and execution time parameters. The UNIT statement specifies a reader (multifile is possible), printer, punch, and up to 9 additional units (input, output, or both, with multiple files possible) for a total of 12 I/O units maximum. The SAVE statement reserves up to 32,767 bytes of user working storage. The MAX statement ends program execution after transmission of a specified number of records from or to one input or output unit. The TIME statement places a specified limit on the minutes of processing time used. The NLINES statement limits the page of printed output to a given number of lines, but does not control the format of the program listing. The CARD statement frees a given portion of each source program card for comments. Fixed-length output padding, with alphanumeric or hexadecimal characters, can also be specified.

Note that the user need only specify his SAVE area. The input, output, and counter areas are created automatically, and all areas are automatically initialized with blanks. The input area is allocated dynamically as input records are read. Both the Declarative and Narrative sections can contain any number of cards that begin with an asterisk, as these contain comments. The end of the Declarative Section is denoted by the $ symbol.

Operands are described and/or defined to UC/360 using the following notation: Literals are data encoded in apostrophes and can be up to 133 bytes long. Literals are normally alphanumeric, unless the initial apostrophe is preceded by an X, in which case the literal is hexadecimal (up to 266 "hex" characters fit into the 133 bytes). INP, OUT, or SAVE, followed by a single-byte address or left-to-right location, specify data fields in the input, output, or SAVE areas of storage. CNTR specifies a 7-byte counter, and is indentified by a number from 0 through 99 that the user must supply. INPUT and OUTPUT are 7-byte fields that respectively contain the length in bytes of the records in the input and output areas; the latter is initialized as 50 bytes. PAGE is a 7-byte page counter field, initialized as 1. BLANK denotes a 1- to 133-byte field of blank characters that is automatically set to the number of bytes called for by context. DATE and JDATE are 5-byte operands automatically set to the current date (format 24 MAY 41) or Julian date (format *yyddd*). TIME is an 8-byte field automatically set to the current time in hours, minutes, seconds, and hundredths of seconds. The V-operand statement assigns a symbolic variable number to any operand except an indexed operand. The number can be any integer in the range 0 through 32,767.

The form *operand$operand* specifies indexing, in which the data is located at the location of the first operand adjusted

UTILITY-CODER/360
Cambridge Computer Associates, Inc.

➤ by the value of the second operand. In the hands of a capable programmer, this indexing capability can become a powerful tool.

UC/360 has the following I/O commands: READ and WRITE, to/from a logical unit, an operand – I/O units being designated as CARD (reader), LINE (printer), or PUNCH (card punch). DISPLAY acts as WRITE but retains the record in the output area. These commands transfer data; it takes a PRINT or PUNCH command to print a line or punch a card. SPACE, followed by a number, skips down a line printer page the specified number of lines or to the top of the next page, whichever is less. RESTORE positions printer forms to top-of-page. MARK pads or truncates a designated data set and writes a file mark and standard OS trailer label as specified. REWIND repositions a unit at the initial record in the first data set; it has input and output functions. CLOSE and OPEN logically disconnect or connect a unit. TRUNCATE causes an input skip of a designated number of records in a data set block or treats the last output record written as the end of a block. SKIPFILE causes a skip to the next data set on a designated unit. QUERY prints the contents of the output area on the console typewriter and halts program execution until a reply is received.

Logic-flow commands are very simple: RESET and NOT, respectively, set the truth indicator to true or reverse its state. STMNT labels a statement to which processing can branch, and GO TO directs an unconditional branch to a labeled statement or to the statement whose label is the value of a specified operand. CALL can be likewise directed, and RETURN directs processing to begin at the statement after the last executed CALL. There are also LOAD and CALL statements for loading and calling external subroutines. END branches back to the first narrative statement. STOP terminates execution when the truth indicator is set to true, and can have an associated return code in the range 0-255.

Full testing facilities are provided for operands, and some tests for I/O conditions are also provided. The user can request IF tests for EQ (equal), NE (not equal), GT (greater than), GE (greater than or equal to), LT (less than), and LE (less than or equal to), and also for whether an operand is in a list (INLIST). Conditionals can be joined with AND, OR, in SAME as, DIFFERS from, and IMPLIES; the last three operators combine elementary logic operators into composite tests.

UC/360 contains an impressive list of data manipulation commands. Data can be MOVED FROM one location TO another of the came core size, or the amount of data so moved can be equal in length to the value specified in an operand in a FOR statement. There is a COMPUTE verb with algebraic operator and expression capabilities for signed numbers, addition, subtraction, multiplication, and division. BUMP increments the value of an operand (which can be a counter) by 1. SIGN specifies an operand where a minus sign will be placed when a COMPUTE, BUMP, or other conversion is a negative number. There is a 1-byte FILL command. LFTADJ is a left-justification command for storage. Conversions available are PACK, UNPACK, HEX, MAKEBIN (convert to binary), and CONVERT (from binary). Also, BOOL and UNBOOL convert a string of binary data to/from a string of character zeroes and ones. Another conversion operation is a table-based TRANSLATE. In addition, the following logical masking operations can be performed at the bit level: LAND, LOR, LXOR (Logical Exclusive OR), and LNOT.

The UC/360 diagnostic commands are: DUMP (system core dump at end-of-job), SNAPDUMP (formatted and immediate display of UC/360 user storage areas), * MAP (print storage allocation map of user areas), * NO MAP (turns off * MAP switch), and * NO RUN (ends processing after compilation; otherwise the program will execute). The commands * PRINT ON or * PRINT OFF resume or suppress printing of a program listing, and * EJECT prints the next program card at the top of the next page; these three command cards beginning with the * are not themselves printed. Diagnostic messages are printed during compilation, under the printed image of any card discovered to have an error, and compilation continues. But if errors are found, the program will not begin to execute, but will terminate after compilation.

Because the preponderance of UC/360 users are, and will continue to be, OS users, the foregoing has been an outline of OS UC/360. There are a few restrictions that exist in the DOS version of UC/360: The INPUT, OUTPUT, DATE, TIME, BLANK, and VAR operands cannot be used. There are no DISPLAY, OPEN, TRUNCATE, or SKIPFILE operators. Data manipulation operators not in the DOS version are FOR, BOOL, UNBOOL, LAND, LOR, LNOT, and LXOR. All of the logic-flow operators of the OS version exist for the DOS version (except LOAD/CALL for branching to external subroutines), but the composite test operators SAME AS, DIFFERS FROM, and IMPLIES do not. The diagnostic operators SNAPDUMP and MAP are not present in the DOS version, nor is the declarative TIME element.

Naturally, device declaration and use are subject to DOS constraints in the DOS version of UC/360. But importantly, UC/360 does not support disk operations under DOS. It is fair to say that the DOS version generally provides the same basic repertoire of commands as the OS version, but that the more complex OS commands must be spelled out by means of sequences of the other available commands in the DOS version.

PERFORMANCE: Users interviewed state that UC/360 functions as advertised, saves a great deal of programming time (or, alternatively, spares programmers by letting others write their own programs), and appears to perform its utility functions at, or very close to, I/O device speeds.

HARDWARE/SOFTWARE REQUIREMENTS: An IBM System/360 or 370 computer under DOS, DOS/VS, OS, or OS/VS, with at least 65K bytes of main storage for the use of the OS version and 32K bytes of main storage for the use of the DOS version.

PRICING: The following are UC/360 prices, which include maintenance and user support. The month-to-month plan is a monthly payment, while the 1- and 3-year plans are by rental prepayment. UTILITY-CODER/360 cannot be purchased or perpetually leased.

Rental Plan	DOS or DOS/VS Price	OS or OS/VS Price
Month-to-month	$275/month	$375/month
1 year	$2,805	$3,825
3 years	$5,495	$7,425

INITIAL DELIVERY: January 1968 for both the OS and DOS versions.

CURRENT USERS: 25 as of December 1973. ■

Optimizer II
Capex Corporation

MANAGEMENT SUMMARY

While IBM's promotions for virtual storage convey the impression to many customers and potential users that availability of main memory is no longer a problem in the field of data processing, sophisticated analysts and users know that it ain't necessarily so. It's a simple fact that a program requiring less memory for its instructions (in COBOL, one could call this "Procedure Space") will benefit the user by leaving more of the system's memory resources for other jobs, and probably by running faster itself, than a larger program. This fact is immutable on a real memory system, and, unless the small program's locality of reference is incredibly bad, it is also true for a demand-paging virtual storage system. (Locality of reference is a program's tendency for its instructions to refer to memory areas that are in the same page, or at least nearby.)

Optimizer II is designed to automatically improve the efficiency of the object code generated by the IBM ANS COBOL compilers. It operates on System/360 and 370 configurations running under any version of OS or OS/VS. With very few restrictions, any valid, executable COBOL object program can be optimized through the use of Optimizer II.

The effect of using Optimizer II is a reduction in the amount of main memory required to hold the object code and in the CPU time required to execute the program. Capex claims overall main storage savings of about 20 to 30 percent and a 15 to 25 percent improvement in execution speed—figures which are substantiated by several users. Capex's confidence in its package is indicated by its inclusion of a statistical module that accumulates information on average optimized program sizes versus unoptimized program sizes. This provides a unique capability for monitoring how much good the package is doing for you, not only on evaluative benchmarks but in a day-to-day working environment.

USER REACTION

Optimizer II continues the outstanding record of Capex's original Optimizer, which itself had become a programming standard in the large EDP installations of several leading firms. Although it uses about 50 percent more main storage in its runs than the original version, Optimizer II has been demonstrated at several important test sites to save an average of about 10 percent more main memory space than the original product, and to further improve performance—typically by about 30 percent. These experiences, in fact, make the vendor's claims pale by comparison, and also demonstrate that the new product is an improvement on a good thing.

To further examine the capabilities of Optimizer II, one test user took a program and spent two man-weeks optimizing it by hand to the best of his expertise. The result of this effort was to decrease the CPU time required to execute the ANS Version 2 COBOL program by 80 percent. Then the same object code was optimized using

Optimizer II can reduce the amount of main memory required by a COBOL program by 20 to 30% and speed the program's execution by 15 to 25%. It does this by modifying the object code generated by the IBM ANS COBOL compilers on System/360 and 370 computers under OS and OS/VS.

CHARACTERISTICS

SUPPLIER: Capex Corporation, 2613 North Third Street, Phoenix, Arizona 85004. Telephone (602) 264-7241.

BASIC FUNCTION: To optimize the object code of IBM System/360 and 370 ANS COBOL programs on OS (MFT, MVT, VS1, and VS2) configurations. (The original Optimizer program, now called Optimizer I, remains available to optimize IBM COBOL F programs on System/360 and 370 OS configurations.)

OPERATION: Optimizer II takes the output from the IBM ANS (Version 2, 3, or 4) COBOL compiler, if it is executable; bypasses the standard output listing, replacing it with a similar but expanded listing; modifies the object procedure division code; and outputs an executable object program. Except for specifying a few extra options, the user hardly knows Optimizer II is there, except for the results it yields.

Optimizer II's techniques involve a total flow analysis that works through every path and loop in the user's program and devises, as a result, global strategies for general register usage and code that direct the procedure flow (PERFORMs, GO's, etc.). The package does not touch any user-defined data area.

Optimizer II reads the compiler-produced object module and thus determines the logical flow of control to instructions in the user's program. It breaks the user's Procedure Division code into "blocks" consisting of contiguous code, using the rule that all branches are into the beginning of a block. This block-branch structure is displayed as part of the optional procedure map (PMAP). The base addresses needed by the code within a block are loaded into registers at the beginning of the block (if addresses are not already in the proper registers), and they remain there throughout the block. Various algorithms are used to maximize the probability that base addresses for each block will already be in the proper register in the blocks that logically precede that block; this effectively minimizes the number of register load instructions in the object code.

Optimizer II's techniques, which include investigation in detail of every instruction in the object program, yield results for almost all COBOL verbs. These are some of the types of optimization performed: (1) subscript calculations are optimized; (2) repetitive calculations are eliminated; (3) TRANSFORM coding is optimized; (4) PERFORM linkage is reduced, often to as little as one instruction; (5) unnecessary synchronizing MOVEs are eliminated; (6) redundant branches are eliminated; (7) nonreferenced blocks of code are removed; (8) loops are optimized by removing as much code as possible and placing it outside of the loop; (9) MOVEs are consolidated wherever possible; and (10) literals are optimized according to usage.

An optimized program will produce exactly the same results as an unoptimized program, but faster and in less main memory.

The output listing includes block number identification for all verbs, along with cross-referencing information that identifies all blocks that branch to the subject block and all

Optimizer II
Capex Corporation

▷ Optimizer II, and in a few minutes the package produced executable object code that decreased the CPU time by 60 percent. While the results were not as dramatic as those of hand optimization, they were nonetheless impressive, and two man-weeks of arduous coding were saved in the bargain. (By the way, the ANS Version 2 COBOL compiler, supplied free by IBM, is very general and can be quite inefficient. For example, it generates multiplication by 1 in indexing instructions.)

A user of two large 370/165's paid Optimizer II what we believe is the ultimate compliment by saying that his company uses the product because it is fully cost-justified. Datapro's interviews with this user and several others also revealed total satisfaction among the users, both with Optimizer II and with Capex as a vendor. Datapro has had contact with beta test users from the outset, and they say the package is well debugged. Optimizer II is a product that no large COBOL user should ignore. □

➤ destinations of branches within the subject block. Base registers and contents are shown for each block. Standardized usage of registers is useful in debugging. At the end of the listing, a small table is appended showing the optimization results in terms of bytes saved.

Optimizer II will not optimize programs using: (1) the Sterling Currency feature of COBOL, (2) debugging or Batch Compilation features of ANS Version 3 COBOL, (4) Segmentation features of ANS COBOL, (5) Symbolic Debugging of Optimized Object Code features of ANS Version 4 COBOL (e.g., PARM = FLOW, STATE, SYMDMP, OPT, BATCH), and (6) non-ANS COBOL.

Programs using the ANS Segmentation feature, however, can be optimized by setting the segment limit to a value greater than the greatest priority number used. Often, the main memory saved by using Optimizer II eliminates the need for overlays. Optimizer II does handle linkage-editor overlays. Optimizer I, somewhat less powerful than Optimizer II, is available for users of IBM's COBOL F.

The optional Statistical Report is a summary of results for programs optimized. Data for current month, current year, and since installation is included. The number of programs optimized during each period, as well as average program size before and after optimization and average procedure size before and after optimization, are shown. Differences in percent are also shown. This report provides an excellent method for continuing evaluation of how much good the package is doing you.

PERFORMANCE: The degree to which Optimizer II's analysis of a COBOL object program improves that program's performance and reduces the memory space required by it naturally varies from program to program, but is typically in the range of a 15 to 25% reduction in CPU time and a 30% reduction in procedure space, which yields about a 20% reduction in the space required for the whole program. Only the compiler-generated instructions are optimized; the data space is unchanged, and external routines, including access methods, are unchanged.

This optimization is at the object level, and is consequently of the sort that could not be expected from a COBOL programmer; and the "people time" of an expert Assembly language programmer to attempt the same kind of optimization might be cost-prohibitive. Pressed for time, the expert programmer might not do as good a job as Optimizer II.

Benefits to the user of the optimization are as follows: (1) reductions in CPU time; (2) reduced partition or region size requirements; (3) a PMAP that is enriched for easier debugging; (4) ability to use larger I/O blocks or more buffers since space is now available, with a consequent reduction in run time; (5) possible elimination of some overlays; (6) availability of more space for the sort routine, and thus less time spent sorting; (7) possibility of executions in standard region sizes; (8) allowances for program expansion; (9) increased multiprogramming ability through the use of saved space; (10) elimination of some scheduling bottlenecks caused by demands of formerly large programs upon main storage; and (11) availability of main memory space to make more operating system modules resident, and thus further increase the speed of the system. Additionally, there are special benefits to the virtual storage user: (1) reduced paging; (2) reduced CPU time due to the reduced paging; and (3) automatic application of saved main memory to increase multiprogramming, thus reducing average job turnaround time on the system.

Please refer to the User Reaction section of this report for additional commentary about Optimizer II performance, including a comparison of the package to Optimizer I.

HARDWARE/SOFTWARE REQUIREMENTS: Optimizer II will run on any valid ANS COBOL configuration of an IBM System/360 or 370 computer operating under OS (MFT or MVT) or OS/VS (1 or 2). Current versions are running under each of these operating systems. Capex is committed to modifying Optimizer II as future OS and OS/VS compilers are released.

Optimizer II for ANS COBOL can operate in as little as 120K bytes of storage, but the best results are obtained when 150K is allowed; above 160K, however, diminishing returns begin. In a release scheduled for the first quarter of 1974, however, Optimizer II will operate in a minimum of 100K and an optimum of 128K bytes. This can be compared to a range of from 80K to 128K required for Optimizer I. Either version of the package can optimize modules up to 1024K bytes in size, and either uses about 15 to 30 cylinders of work space on a 2314-type disk. As a rule of thumb, either version of the package always requires a region at least as large as the original program or the COBOL compiler, whichever is larger.

PRICING: The cost of the Optimizer II package depends upon the size of the user's system, according to four defined levels. Level I is one or more 360/40's or 370/135's. Level II is one or more 360/50's or 370/145's. Level III is a single 360/65, 370/155, or 370/158. Level IV is any larger system. Level II-IV systems can include smaller systems. Rental terms are month-to-month, with a 90-day minimum. Lease terms are 60-day cancellable, and leases are available for 12-month or 36-month periods. Maintenance and a 90-day conversion-to-license credit are included under the rental and lease plans. License is perpetual, subject to payment of an annual maintenance charge after the initial year. Multiple-site discounts can apply only to the license fee, and not to maintenance. Prices are shown in the table below. The annual maintenance charge is 10% of the first-site license.

		Lease		License	
Level	Rental	12-month	36-month	First	Additional
I	$333	$300	$265	$ 8,000	$ 6,400
II	500	450	400	12,000	9,600
III	667	600	535	16,000	12,800
IV	833	750	667	20,000	16,000

INITIAL DELIVERY: Optimizer I for COBOL F was first delivered in November 1970. In 1971, support for ANS COBOL Version 2 (IBM's free 360S-CB-545) and ANS Version 3 COBOL (IBM's Program Product 5734-CB1) was delivered in Optimizer I. In 1972, Optimizer I support for ANS Version 4 COBOL (IBM's Program Product 5734-CB2) was delivered. Optimizer II was first delivered in November 1973.

CURRENT USERS: Over 180 in the United States as of December 1973, plus 25 in Europe and 10 in Canada and South America. ■

COTUNE
Capex Corporation

MANAGEMENT SUMMARY

COTUNE is a software product that pinpoints critical information necessary in COBOL program development, optimization, debugging, testing and validation, maintenance, and run documentation by means of an execution-time data gathering process. Its output is a source listing that contains counts showing how many times each statement was executed, a normalized histogram showing the percentage of CPU time spent in each source statement, and an indication of any source statement at which an ABEND occurred. Additionally, summary reports show all unexecuted paragraphs and which paragraphs consumed the most CPU time. Inportantly, this information can be understood by COBOL programmers and used by them to fix or improve the programs. Analysts with knowledge of Assembler or other languages are not required for this function.

COTUNE can be used for any valid COBOL program on any IBM System/360 or 370 under an OS or OS/VS operating system. Its use requires no change to the problem program, the compiler, or the operating system.

It is instructive at this point to draw the distinctions between COTUNE and Capex's COBOL Optimizer. COTUNE is a tool for COBOL source-code evaluation. Optimizer is a program that automatically optimizes object code. A number of users employ both of these fine products. The Capex line also includes FORTUNE, a FORTRAN source code evaluation tool, and AUTOTAB, a business and financial planning tool.

COTUNE can show a COBOL programmer exactly where and how a COBOL program can be improved – showing, for example, unexecuted (and thus either unnecessary or untested) code, the most efficient order for a series of IF tests, inner loops in a program or section, tests that may be unnecessary (for one-state conditions), infrequently used code that could be placed in an overlay, and logical groupings of code that could optimize virtual storage paging or ANS COBOL segmentation. This last point is called "locality of reference," and relates to what is sometimes called the "20-80 rule." That is, it is commonly the case that 20 percent of the code in a particular program accounts for 80 percent of the time in execution (e.g., does 80 percent of the work).

This 20 percent is called the program's "working set" by some experts. In real life, the working set is not the static entity that springs to mind from the 20-80 rule, though. In fact, the working set is usually dynamic, varying in content and size during a program's execution. Nonetheless, localizing the static working set (which is all any reasonably-priced tool can be expected to identify), and placing it into one page (or, if large, contiguous pages), is presently the best known way to begin program optimization under a paging supervisor.

COTUNE allows COBOL programmers to evaluate and improve the performance of programs by producing debugging, testing, maintenance, and optimization aids and reports. It can evaluate each Procedure Division sentence and/or paragraph. It runs on IBM System/360 or 370 computers under OS/MFT, OS/MVT, OS/VS1, or OS/VS2.

CHARACTERISTICS

SUPPLIER: Capex Corporation, 2613 North Third St., Phoenix, Arizona 85004. Telephone (602) 264-7241. Capex also has sales offices in Englewood, NJ; Washington, DC; Cleveland, OH; Boston, MA; Dallas, TX; San Francisco, CA; and Milwaukee, WI. Capex products are also marketed in Canada, Western Europe, Japan, and Australia.

BASIC FUNCTION: Measurement and analysis of COBOL programs on IBM System/360 and 370 computers under OS and OS/VS. COTUNE's use is oriented toward COBOL programmers, and its design objectives are clarity, ease of use, and simplicity of training and installation.

OPERATION: The COBOL source program under test is read by the COTUNE preprocessor, which produces a modified source program containing measurement statements and information. Counts on the exact number of times a statement or group of statements is executed are derived from inserted code that adds one to an accumulator for each execution. The system's timer is sampled to obtain timing information. This is all automatic in the modified program, which is simply compiled, link-edited, and executed.

The executed program produces results identical with those of the original program plus execution analysis data, which is written to a data set. The COTUNE post-processor then analyzes the data and correlates it with the original COBOL source code to produce the COTUNE analysis reports.

No changes to the source program, compiler, or operating system are required, and any valid COBOL program, from COBOL F through ANS Version 2 to 4 COBOL, can be so tested.

COTUNE responds to object modules written in other OS languages by relating the time spent therein back to the appropriate CALL statements in the COBOL program. Time spent in access method routines is related back to the corresponding OPEN, CLOSE, READ, WRITE, and other I/O statements. All executions of overlay modules are analyzed and reported, regardless of the number of times overlaid. An ANS COBOL program using Segmentation is reported on in its entirety, regardless of the overlay structure, since it is a single COBOL source module.

Output consists of the normal COBOL output plus a preprocessor summary report and several analysis reports. The Execution Profile Listing is a complete listing of the COBOL source program, with execution counts and CPU time information printed to the right of each source statement. The information on percent of time spent is presented numerically and as a normalized histogram (bar graph in which the longest bar takes up exactly the space provided on the page). Unexecuted sentences are denoted by a zero in the count column. The Execution Profile Listing can be produced at the sentence or paragraph level. ABEND locations are shown in place.

COTUNE
Capex Corporation

COTUNE also shows the number of records read and written and any incomplete testing of IF conditions. If an ABEND (IBM term for abnormal program termination) occurs, COTUNE pinpoints the place in the source code where it happened. If this were not done, the location would have to be determined by persons familiar with Assembler code from a core dump – and this is a costly, time-consuming process.

COTUNE can assist a newly assigned programmer in understanding the code of another programmer – a valuable asset in program maintenance. While it does not flowchart the program (AUTOFLOW and QUICKDRAW are well-known products that do that), it does show a program's main and exception paths, inner loops, etc., by denoting where time is spent in execution. Used comparatively on sampled runs of a program, this type of data can form a basis for run documentation and control. Data on algorithm and language constructions can also be gathered this way.

USER REACTION

Significantly, six out of six users of COTUNE interviewed by Datapro for this report commented that their programmers can use COTUNE's output to debug, improve, maintain, and document COBOL programs, even without knowledge of Assembler language. Savings resulting from COTUNE's use are often quite worthwhile. Some of the users had spectacular gains to report: examples included savings of 40 to 60 percent of the CPU time for a program, cutting a program's run time to 27 percent of what it was originally, and reducing a program's execution time from 28 minutes to 4 minutes. Since the improvements that can be made in a program depend heavily on the nature of the program itself and on the programmer's intelligent use of the COTUNE information, all these claims can be considered valid.

When a program can be improved by COTUNE, the resulting improvement is typically about 20 to 25 percent. But if "average performance improvement" were calculated by figuring in those programs that the package shows can't be improved much or at all, the percentage figure would be somewhat diminished. Yet, when COTUNE tells you that a program can't be improved, it's sort of like having your physician tell you after a check-up that you're in perfect health–the doctor's bill is money well spent, and the information is both reassuring and valuable.

COTUNE users agree that the package is easy to use and easy to install. (About one day is typical.) They also agree that Capex service and support is all that could be expected, and that the program is bug-free and performs as advertised. They use COTUNE for various purposes, and some have tips on its use: restrict COTUNE analysis to a controlled test atmosphere, and make COTUNE a program-checkout standard.

In summary, COTUNE is a valuable programmer tool that should be looked at seriously by all OS and VS 360/370 COBOL shops. □

The Execution Time Summary Report shows the 10 most time-consuming paragraphs in the program, listed in descending order. Shown are the paragraph names, percent of CPU time spent in each, source statement sequence numbers of the paragraphs, page numbers of the Execution Profile Listing on which each paragraph listing begins, seconds of CPU time spent in each paragraph, and a CPU time histogram.

Another report, printed two-up across the page, is the Unexecuted Paragraph Report. This report, which can be suppressed, also shows the user sequence numbers and each paragraph's starting location on the Execution Profile Listing.

Finally, the Analysis Summary is printed, giving the following: (1) COTUNE version date, (2) analysis date, (3) report options used, (4) number of executable statements in the program, (5) amount of storage COTUNE added to the program, (6) the sampling rate used, (7) total samples taken, (8) samples dropped and their percentage of the total samples, (9) estimated processing time in seconds, (10) the histogram scale, in terms of the percentage a full line represents, and (11) a warning if the sampling may be insufficient.

COTUNE is supplied on a card deck and magnetic tape, and installation is a simple cataloging procedure, using IEHMOVE. Full installation information and a test COBOL program are included.

PERFORMANCE: Use of COTUNE slightly degrades the execution time of a program, so it should be used only for special testing needs. According to users, it functions as advertised. Programs under test could run faster if the user had the option of selecting just count or timing information, instead of both, as has been suggested by one user. As it is, test runs can be made more quickly if paragraph-level testing, instead of sentence-level testing, is selected. COTUNE also has its own clear diagnostic messages.

HARDWARE/SOFTWARE REQUIREMENTS: An OS or OS/VS IBM System/360 or 370 with an interval timer. The analysis summary printed by the preprocessor requires that main storage for the program under test be increased by at least 8K bytes. The exact formula for enlargement is (8 + n/120)K bytes, where n is the number of Procedure Division statements. This add-on requirement can be reduced significantly by selection of the Paragraph or Detail options for large programs, but is increased slightly if the Overlay option is used (unless the program is processed by the Capex COBOL Optimizer). The preprocessor job step requires 64K bytes of main storage. The run-time processor step requires 8K bytes plus the prior COTUNE expansion.

PRICING: COTUNE can be leased for a monthly charge of $275. Maintenance is included with the lease, and the first 90 days' payments can be applied to a license. A license for COTUNE costs $5,750, with maintenance included for the first year and priced at $575 per year thereafter. License discounts are available for second and multiple sites.

INITIAL DELIVERY: COTUNE was first delivered by Capex in October 1972, but Datapro has spoken to users who obtained it as early as 1971 from Compu-Trend, of Cupertino, California, the developer and original vendor. Capex acquired the package in mid-1972, withdrew it from the market for a few months, and significantly improved it before the September 1972 re-release.

CURRENT USERS: 37 as of November 1973. ■

FORTUNE
Capex Corporation

MANAGEMENT SUMMARY

FORTUNE is a software product that pinpoints critical information necessary in FORTRAN program development, optimization, debugging, testing and validation, maintenance, and run documentation by means of an execution-time data gathering process. Its output is a source listing that shows how many times each statement was executed during a run, an estimate of the CPU time spent in each source statement, and the exact number of times the "true" path was taken on each IF statement. Additionally, summary reports show all unexecuted statements and which statements consumed the most CPU time. Importantly, this information can be understood by FORTRAN programmers and used by them to fix or improve the programs. Analysts with knowledge of Assembler or other languages are not required for this function.

FORTUNE can be used for any valid FORTRAN program on an IBM System/360 or 370 under OS or on a Honeywell Series 6000 under GCOS. It use requires no change to the problem program, the compiler, or the operating system.

FORTUNE is a tool for FORTRAN source-code evaluation – not a program that automatically optimizes object code. Capex's product line also includes COTUNE, a COBOL source code evaluation tool; AUTOTAB, a business and financial planning tool; and Optimizer, a program that automatically improves COBOL object code efficiency.

FORTUNE can show a FORTRAN programmer exactly where and how a FORTRAN program can be improved – showing, for example, unexecuted (and thus either unnecessary or untested) code, the most efficient order for a series of IF tests, inner DO loops in a program or section, tests that may be unnecessary (for one-state conditions), infrequently used code that could be placed in an overlay, and logical groupings of code that could optimize virtual storage paging. This last point is called "locality of reference," and relates to what is sometimes called the "20-80 rule." That is, it is commonly the case that 20 percent of the code in a particular program accounts for 80 percent of the time in execution (e.g., does 80 percent of the work).

This 20 percent is called the program's "working set" by some experts. In real life, the working set is not the static entity that springs to mind from the 20-80 rule, though. In fact, the working set is usually dynamic, varying in content and size during a program's execution. Nonetheless, localizing the static working set (which is all any reasonably-priced tool can be expected to identify), and placing it into one page (or, if large, contiguous pages), is ▷

FORTUNE allows FORTRAN programmers to evaluate and improve the performance of programs by producing debugging, testing, maintenance, and optimization aids and reports. It runs on IBM System/360 or 370 computers under OS/MFT or OS/MVT, and on Honeywell Series 6000 computers under GCOS.

CHARACTERISTICS

SUPPLIER: Capex Corporation, 2613 North Third St., Phoenix, Arizona 85004. Telephone (602) 264-7241. Capex also has sales offices in Englewood, NJ; Washington, DC; Cleveland, OH; Boston, MA; Dallas, TX; San Francisco, CA; and Milwaukee, WI. Capex products are also marketed in Canada, Western Europe, Japan, and Australia.

BASIC FUNCTION: Measurement and analysis of FORTRAN programs on IBM System/360 and 370 computers under OS and Honeywell Series 6000 computers under GCOS. FORTUNE's use is oriented toward FORTRAN programmers, and its design objectives are clarity, ease of use, and simplicity of training and installation.

OPERATION: The FORTRAN source program under test is read by the FORTUNE preprocessor, which produces a modified source program containing measurement statements and information. Counts on the exact number of times a statement is executed are derived from inserted code that adds one to an accumulator for each execution. Timing information is an estimate, based on a basic "cost" for each statement type (the "cost" of GO TO is 2), and a further incremental "cost" for: each operation in arithmetic statements, each argument in subroutine entries, and each invocation of intrinsic functions. This is all automatic in the modified program, which is simply compiled, link-edited, and executed.

The executed program produces results identical with those of the original program plus execution analysis data. The FORTUNE post-processor then analyzes the data and correlates it with the original FORTRAN source code to produce the FORTUNE analysis reports.

No changes to the source program, compiler, or operating system are required, and any valid System/360/370 FORTRAN IV Level G, Level H, or H Extended program or Honeywell 6000 FORTRAN IV or FORTRAN V program can be so tested.

Output consists of the normal FORTRAN output plus a summary report and an execution profile report.

The Execution Profile Listing is a complete listing of the FORTRAN source program, with execution counts, "time" information, and the number of times "true" was the result of a test printed to the right of each source statement. Unexecuted statements are denoted by a zero in the count column.

The Program Summary Report shows the relative amounts of time consumed by each routine in the program. Shown are the routine names, relative CPU time spent in each routine, and percent of CPU time spent in each routine. ►

FORTUNE
Capex Corporation

▷ presently the best-known way to begin program optimization under a paging supervisor.

Another important point here is the necessity for the programmer or analyst to possess accurate statistical knowledge about the data on which the program operates. Unless it is known with some certainty how often IF tests branch in certain ways, it is impossible to localize the code of the prominent branch path; but this knowledge can only derive from statistically valid test data in statistically significant quantities.

FORTUNE can assist a newly assigned programmer in understanding the code of another programmer – a valuable asset in program maintenance. While it does not flowchart the program (AUTOFLOW and QUICKDRAW are well-known products that do that), it does show a program's main and exception paths, inner loops, etc., by denoting where time is spent in execution. Used comparatively on sampled runs of a program, this type of data can form a basis for run documentation and control. Data on algorithm and language constructions can also be gathered this way.

USER REACTION

Significantly, every one of the users of FORTUNE interviewed by Datapro for this report commented that his programmers can use FORTUNE's output to debug, improve, maintain, and document FORTRAN programs, even without knowledge of assembly language. Savings resulting from FORTUNE's use are often quite worthwhile. Realistically, users typically report only modest improvements in most production programs tested under FORTUNE. Larger gains stem from treatment of troublesome programs, or of programs under test. Since FORTUNE adds to both the size and run time of the program it's analyzing, and since the improvements that can be made in a program so analyzed depend on the programmer's intelligent use of the FORTUNE information, differing claims are valid.

FORTUNE users agree that the package is easy to use and easy to install. (About one day is typical.) They also agree that Capex service and support is all that could be expected, and that the program is bug-free and performs as advertised. They use FORTUNE for various purposes, and some have tips on its use: restrict FORTUNE analysis to a controlled test atmosphere, and make FORTUNE a program-checkout standard.

In summary, FORTUNE is a valuable programmer tool that should be looked at seriously by all System/360 and 370 OS and Honeywell Series 6000 FORTRAN shops.□

➤ FORTUNE is supplied on a card deck and magnetic tape, and installation is a simple cataloging procedure, using IEHMOVE. Full installation information and a test FORTRAN program are included.

PERFORMANCE: Use of FORTUNE slightly degrades the execution time of a program, so it should be used only for special testing needs. According to users, it functions as advertised. FORTUNE also has its own clear diagnostic messages.

HARDWARE/SOFTWARE REQUIREMENTS: An OS IBM System/360 or 370 or GCOS Honeywell Series 6000. The analysis summary printed by the preprocessor requires that main storage for the program under test be increased by about 6K bytes. The preprocessor job step requires 45K bytes of main storage. The post-processor step requires 5.5K bytes plus the prior FORTUNE expansion.

PRICING: FORTUNE can be leased for a monthly charge of $200. Maintenance is included with the lease, and the first 90 days' payments can be applied to a license. A license for FORTUNE costs $3,950, with maintenance included for the first year and priced at $395 per year thereafter. License discounts are available for second and multiple sites.

INITIAL DELIVERY: FORTUNE was first delivered by Capex in September 1972. It was acquired from Compu-Trend in mid-1972, withdrawn from the market, and significantly improved by Capex before the September 1972 re-release.

CURRENT USERS: 19 as of November 1973. Of these, 10 are using IBM systems and 9 are using Honeywell systems. ■

Autotab
Capex Corporation

MANAGEMENT SUMMARY

The Capex Autotab package is a tool for financial and business planning that lets non-EDP personnel easily prepare various kinds of planning, projection, analysis, or management reports on a computer system. Autotab can be used with IBM System/360 and 370 computers running under OS or OS/VS, or with Honeywell Series 6000 computers running under GCOS. Its facilities can be made available to terminal users in a time-sharing mode. Autotab's output reports can appear on virtually any ASCII terminal, as well as on a system's line printer when the batch processing mode is used.

While Autotab is primarily designed to help its users build complete financial systems (e.g., by preparing various budgets, revenue projections, and cash projections, and then tying them together into an interrelated system), it can also be used for many other applications. These diverse applications to date have included engineering tabulations, manpower projections, depreciation schedules, and market research tabulations. Thus, Autotab can be used directly by corporate planners, financial planners, budget analysts, systems analysts, business planners, and, in general, anyone making financial plans or projections. The managers, engineers, and accountants who use Autotab need not have a data processing background.

Half of Autotab's functional strength derives from the ease with which it can be used. The other half lies in the extraordinary facilities it presents for the manipulation of data in tables.

Tables can be saved, so that judiciously formatted tables can be built into a system of pyramided reports. Data in a table can be changed, and the effects rippled through that table and related tables. Data in tables can be used in calculations, shifted, accumulated, and otherwise manipulated in ways that seem almost unlimited. The Autotab user's manual gives examples of how and why these things may be done. Autotab even has a "what if" facility that allows users to specify sets of reports based upon assumptions regarding the data and calculation rules. Table presentations can be made more meaningful by Autotab's ability to sort columns, rows, and totals in ascending or descending order.

With all these facilities, Autotab can be used for long-range and short-range planning and for a wide variety of other financial and engineering functions. It is also possible to regard tables as matrices and manipulate them according to the rules of that mathematical discipline.

USER REACTION

Autotab users all agree that Capex is vending a fine product and providing fine service. There were no complaints about performance, memory usage, or any aspect of the Capex-supplied documentation or service. Users all also agreed that the package can be installed without difficulty in one day, and that professional non-EDP personnel can use the package without learning difficulty. ▷

Autotab aids non-programming professionals in preparing business plans, projections, or management reports on terminals or line printers. It can be used on an IBM System/360 or 370 under OS or OS/VS, or on a Honeywell Series 6000 under GCOS.

CHARACTERISTICS

SUPPLIER: Capex Corporation, 2613 North Third St., Phoenix, Arizona 85004. Telephone (602) 264-7241. Capex has U.S. sales/service offices in Cleveland, Milwaukee, San Francisco, Washington, and in New Jersey (for the Philadelphia and New York metropolitan areas), and has representatives in Western Europe, South America, and Japan.

BASIC FUNCTION: Autotab is a tool that produces tabular reports for financial and business planning as well as other planning, projecting, and analyzing functions. Designed to be used by professionals without data processing backgrounds, it features ease of use and facilities for very creative use of tables to manipulate, process, and present data. Versions are available for the IBM System/360 and 370 under OS or OS/VS (batch and/or remote via CRJE or TSO) and the Honeywell Series 6000 computers under GCOS (also batch or remote).

OPERATION: Autotab can be installed easily in any of its prescribed environments, following the simple procedures in the user's manual. Installation, of course, does not concern the ultimate Autotab users, who operate with the package as described in the same manual, which explains the system's operation in detail and provides examples. Most people can train themselves from it. Also in the manual are examples of the job control language needed to run an Autotab program under the various environments.

Major table elements that must be described by the user for a basic Autotab-generated table are the row and column identities and the data. Naturally, Autotab also lets the user specify titles, row names, column headings, and other format elements, such as editing of data, spacing, footnotes, etc. Shorthand notation that will propagate a data value through all or part of a row or column is also provided, as are the ability to specify totals and accumulations of various items, and to make calculations upon data.

Up to 10 lines of automatically centered title information can be produced. Each input title must be enclosed in apostrophes. By coding DATE without apostrophes, the date (in mm/dd/yy format) will follow the title. Date and title can be forced to the left or right margin (individually) by coding LEFT or RIGHT. A title line can continue up to 60 characters.

Rows and columns must be *named*, as this serves to locate the data, but row and column *headings* are optional. The names are used in the Autotab coding to refer to the rows and columns, while the headings are for appearance. Headings can consist of 1, 2, or 3 lines per row or column and can have up to 60 total characters. Headings, like titles, are enclosed in apostrophes. Names are limited to 8 characters, and words that have meaning to Autotab cannot be used as names.

Autotab assumes a value of zero for any unspecified data. Thus, table formats can be checked without first supplying data; the table is then printed with dashes where the data would be. Basic data for a row or column is given by the row or column name followed by an equal sign (=), and then the data. The data values are *numbers* appearing in the first 72 columns of the card, separated by blanks or spaces. ►

Autotab
Capex Corporation

▷ Users state that the system's performance is as fast as could be desired, both in remote and local batch use. There were two items mentioned by Autotab users that fall into the category of enhancement suggestions: (1) make it possible for the package to interface with other software (i.e., provide user exits), and (2) continue to provide the Autograph package (once popular with Honeywell users) in current versions. The Autograph package, whose name is probably self-explanatory, also used to interface with Autotab.

In summary, Autotab provides a simplified, yet comprehensive, method of report writing on a computer facility that could represent a valuable service in many organizations. □

► The numbers are specified in decimal (floating-point) format; they may have up to 10 integer digits and 4 fractional digits. Numbers can be entered as percentages, but are printed as decimal numbers. If a number is specified for the same position more than once (e.g., both in the row and column), the last entered value is used in the table, and no warning is given. Shorthand notations for easing the task of entering numbers permit using slashes (/) to repeat a number in following positions, entering a number FROM one position THRU another (to the end of the row or column if THRU is omitted), or skipping positions (asterisks are coded) without changing already-entered values.

Calculations in Autotab are normally specified for entire rows and/or columns. For example, if A, B, and C are row names, C = A + B means that row C values will be calculated by adding corresponding elements in rows A and B. Calculation of C = A - B, C = A * B (multiplication), C = A/B (division), and C = A**B (exponentation) follows similar rules. C = B + SQR A would create a row C whose elements are the sums of the corresponding elements in row B and the square roots of the corresponding elements in A. The user can also specify A = SUM B THRU F, which would create row or column A with elements the sums of corresponding elements in rows or columns B, C, D, E, and F. Also, RULES can be stated. For example, if DATA is R1 = 12 24 37, and RULES R2 = R1 is coded, then row 2 (R2) will be set equal to row 1 (R1). And, as in FORTRAN, R1 = R1 + R2 means that new R1 values are to be calculated according to the right side of the equation.

Specifying spacing is very simple. B means place a blank line before a row; A means print a blank line after a row; O (the letter, not zero) means print a line of dashes over a row; U means print a line of dashes under a row; = means print a double underline (using the equals sign) after a row; and NP means start a new page.

Editing rules are also simple. For example, placing a $ before a row or column name causes each element to appear as a dollar amount. The coding .###, .##, or .#, respectively, causes numbers in a specified row or column to appear with 3, 2, or 1 decimal places. Rounding is automatic. Words (up to 5 letters) or symbols can follow each number in a specified row or column if the user codes the entry enclosed in apostrophes. Up to 15 items can be postfixed LBS, %, YARDS, etc. Editing statements can be combined; and where row and column editing instructions conflict, column editing takes precedence, with no warning provided. Also, rows or columns can be suppressed, and zeros can replace dashes in rows or columns, as desired.

Single-line footnotes appear by coding FOOTNOTES followed by the footnote statements enclosed in apostrophes (e.g., '(1) - TAX RATE IS 6%'). Remarks, which appear in the table description, are coded with an ampersand (&) beginning each line. Page dimensions are stated (e.g., LINE 80, PAGE 60 for 80 characters on a line and 60 lines per page) or defaulted.

To set up a system of table using Autotab, you tell Autotab to SAVE the completed table and to COPY it into a dependent table. Portions of tables can be SAVED and COPIED.

The WHAT IF facility allows the user to specify various sets of assumptions about the data and calculation rules. Then, in a single Autotab session, the user can receive a different tabular report for each set of assumptions. For each case, the user can give: (1) one more more DATA specifications, (2) one or more RULES specifications, and (3) one or more COPY specifications. The user just codes RULES, WHAT IF CASE 1, (conditions), etc., and states the rule(s) to be applied in that case.

Advanced Autotab features include cumulative sums for rows or columns, the ability to manipulate individual data elements, rules for partial rows or columns, averages, fixed and compound growth rate calculations, discounted rate of return calculations, and calculated returns on investments. Also, temporary data can be used, conditional calculations can be made (RULES with an IF/THEN clause), and rows and columns can be sorted using various criteria. In addition, the Autotab user's manual gives examples for special, advanced use of the system.

Autotab has 35 error messages: 5 for abnormal run terminations, 7 for errors in use of the COPY/SAVE feature, 1 for arithmetic errors, and 22 for general, format, and statement errors. In most cases, the error is ignored, and Autotab attempts to complete and print the table, so that the user gets as much use out of the run as is possible. Autotab does not notify the user if he overrides a row entry with a column entry, nor if he specifies division by zero or attempts to take the square root of a negative number. Inclusion of these and other warning messages would be a definite improvement.

PERFORMANCE: According to users, Autotab produces "instantaneous" results on terminals and very fast reports on batch-mode line printers while using little computer time. All users interviewed said that Autotab performs as advertised.

HARDWARE/SOFTWARE REQUIREMENTS: An IBM System/360 or 370 OS or OS/VS or Honeywell Series 6000 GCOS computer system. The main memory requirements for IBM systems depend on the product of the numbers of rows and columns in the table, as follows:

Rows x Columns	K-Bytes of Main Memory
50 x 13	42
100 x 13	48
150 x 13	57
100 x 25	64
200 x 13	65
250 x 13	74
250 x 50	162
250 x 100	262

For the 36-bit-word Honeywell 6000 machines, the main memory requirement is 50K words assigned to the timesharing system.

PRICING: Autotab can be rented on a month-to-month basis for a minimum of 90 days for $375 per month, with full credit of rental payments toward purchase of a perpetual license for the first 3 months. A 12-month lease for Autotab costs $330 per month, and the initial 3 months' payments can again be credited in full toward purchase. Maintenance is included in the lease and rental prices. A perpetual license costs $9,000 for the first site and $7,200 for the second and each subsequent site, with per-site maintenance after the first year costing an additional $900 per year.

INITIAL DELIVERY: January 1972.

CURRENT USERS: 22 as of November 1973. ■

DUCS-VI
C F S, Inc.

DUCS-VI is a local display control system for DOS and DOS/VS System/360 and 370 computers that is inexpensive, economical of memory, and capable of transparently simulating 2260 displays on 3270's. Support for remote CRT's and paged operation under DOS/VS will be released in December 1973.

MANAGEMENT SUMMARY

DUCS (Display Unit Control System) has a history dating back more than six years. Version 2.1 was an IBM no-charge Type III prior-use program that provided a simplified I/O method to support 2260-type CRT devices in a local environment. It required only about 1700 bytes, yet was reputed to provide 25 to 33 percent faster response than access modules written with BTAM modules. The package, IBM PID Number 360D-00.6.006, still has thousands of faithful users.

The original author of DUCS is Richard K. Goran, founder and president of C F S, Inc. His marketing of DUCS began with an improved version, DUCS-IV, offered at $25 per month.

DUCS-VI is the current version, and it is catching hold among users. Many users of the free version from IBM have graduated to DUCS-IV and/or DUCS-V, and others have initially adopted DUCS in one version or another.

DUCS-VI, a complete re-write, includes native-mode 3270 support and an accounting facility that provides the user with statistical data relating to the use of each terminal and the terminal network as a whole.

An optional simulation feature within DUCS-VI permits programs written for 2260's under previous versions of DUCS to be executed on 3270's. No changes are required in the user's programs. DUCS-VI provides new format facilities with a simple means of using all of the 3270 enhancements, such as full field manipulation, selector light pens, and operator ID card readers.

Using DUCS, a system can have multithread (concurrent) display unit operation–a mode unobtainable under BTAM. Moreover, DUCS is easy to use; it interfaces with problem programs written in COBOL, PL/1, or Assembler Language, so that programmers do not have to learn to use another language. No knowledge of Assembler is needed to use DUCS-VI.

Finally, DUCS-VI is very inexpensive, and it uses only about 4K to 6K bytes of main storage. (In fact, DUCS-VI is smaller than the previous version.)

DUCS-VI currently supports only local networks of 2260's, 3270's, or the latter simulating the former. It operates in a real (fixed-page) mode under DOS/VS and under DOS. But its capabilities will soon be greatly extended. In December 1973, support for remote 3270 networks and virtual-mode (paged) support under DOS/VS will be released. ▷

CHARACTERISTICS

SUPPLIER: C F S, Inc., P.O. Box 662, Brookline, Massachusetts 02147. Telephone (617) 731-3474.

BASIC FUNCTION: Control of local 2260 and 3270-type display systems without any need for the user to posses special programming knowledge. An optional feature permits user- and program-transparent simulation of 2260's on 3270 networks. DUCS-VI also can obviate the need for IBM's BTAM and provide multithread operation. The program runs on IBM System/360 or 370 computers under DOS or in a real mode under DOS/VS. The additional functions of remote 3270 support and virtual-mode operation under DOS/VS are scheduled for release in December 1973.

OPERATION: DUCS can be used to interface problem programs written in COBOL, PL/1, or Assembler Language. It does not require the use of BTAM or any of its facilities, nor does the programmer need to know Assembler Language or have specific knowledge about teleprocessing.

Any number of displays can be supported, with the sole restriction that if the relative addressing feature is used with the DUCS format facilities, all 3277's must have the same screen capacity. Any combination of 2260's and 3277's can be used. Another small restriction on 2260 simulation is that if 3277 Model 1's are used (the 480-character model), only 479 characters can be written to a simulated 2260 screen. This is because of the need to reserve an "attribute byte."

The current release of DUCS-VI supports only local networks. Remote support will initially support only 3270's and 3275's, and only on leased lines. Future releases will provide dial-up (switched) line support. The current version also must operate only in the virtual-real mode under DOS/VS. This restriction will be removed in December 1973.

DUCS-VI provides: (1) references to itself by means of standard CALL statements; (2) core requirements (for DUCS itself) of usually less than 4K bytes; (3) full support for all 3270 features, for 1053 Printers attached to a 2848 Control Unit, and for 3284 and 3286 Printers attached to a 3270 Control Unit; (4) ability to flag units as off-line, yet permit testing of the units concurrently with problem program execution; (5) ability to swap logical device addresses so that, in the event a critical unit becomes inoperable, time won't be lost; and (6) terminal tests that enable a field engineer or display operator to test the display unit without interfering with its programmed use.

All I/O under DUCS-VI is at the EXCP level instead of at the Start I/O level as in earlier versions. ➤

DUCS-VI
C F S, Inc.

▷ DUCS-VI is a sophisticated display control system that is steadily evolving toward a full-fledged data communications monitor. It can help users stave off the need for upgrades to complex systems such as IBM's CICS (see Report 70E-491-02). It can also allow present 2260 users to convert to 3270's for economy without immediate reprogramming; they can then convert to full 3270 usage in a smooth and leisurely manner.

USER REACTION

DUCS users comment readily about how easy the program was to install and is to use. Typically, it's been brought up on a Saturday morning, or between noon and 5:00 pm, or something like that. No trouble at all. There's no trouble in its use, either. One user mentioned 3 years and 9 months, so far, of trouble-free use.

While praise for the package is unvarying, user background is not. Some came on-board from the IBM version, some initially acquired DUCS from C F S, and a few have already left (or are planning to leave) DUCS for more ambitious communications monitor systems. Nonetheless, all reported the same ease of use and economy of main memory—about 4K to 8K bytes attached to the application program. Ambitious systems can be built with DUCS-VI, too; one user plans to let his 360/30 system, dedicated to production and inventory control, grow from 4 to 18 local 3270-type displays, all simulating 2260's and all in multithread operation.

Technical support from C F S is excellent, as one would expect when the program's author is president of the firm. About half of the DUCS users interviewed could not rate the support because they have simply never needed any. Mailed improvements have the reputation of working the first time, with no installation problems.

In summary, IBM computer users with small display systems and those about to implement display systems should take a look at what DUCS-VI has to offer at its attractive price. □

➤ **PERFORMANCE: DUCS has a reputation for performing faster than BTAM and for being about as bug-free as is possible. There have been no reported failures of the package, nor have there been reports of the package functioning other than as represented by its vendor.**

HARDWARE/SOFTWARE REQUIREMENTS: Any DOS or DOS/VS IBM System/360 or 370; there is no minimum machine configuration other than the device configuration upon which the user wishes to rely. The user makes a simple change to a DOS or DOS/VS Supervisor macro, as detailed in the DUCS documentation. An 800- or 1600-bpi magnetic tape unit is needed to read the program and its documentation, which are provided on magnetic tape at the specified density. The reel of tape must be supplied by the customer, or costs an additional $10.

PRICING: DUCS-IV was priced at $25 per month (it provided 2260 support). DUCS-V cost $25 per month, plus an additional $15 per month for 2260 simulation on 3270 displays. DUCS-VI is the current product, and its prices, which include maintenance, are as follows:

	Monthly Rental	Yearly Rental	One-Time Lease Fee
DUCS-VI	$75	$810	$1,687
2260/3270 Simulator	15	162	337
3270 Remote Support	55	594	1,237

INITIAL DELIVERY: The IBM Type III program version became available in September 1967. DUCS-IV was delivered in August 1972. This was followed by DUCS-V in January 1973. DUCS-VI was announced in mid-September 1973 and installed during the same month. Remote 3270 support and operation in virtual mode under DOS/VS will be released in December 1973. This release will support use of leased lines; support for switched lines will follow in a subrequent release.

CURRENT USERS: Not counting users of the free IBM version, there were more than 35 users of DUCS program products as of October 1973. ■

Total
Cincom Systems, Inc.

MANAGEMENT SUMMARY

Total has generated more inquiries from DATAPRO 70 subscribers than any other single proprietary software package. This, coupled with the active interest in data base technology in general, can be taken as a resurgence of interest in data base processing leading to management information systems (MIS).

Total is a host-language data base management system, implemented much along the lines of the CODASYL Data Base Task Group Report, except that the user may use other host languages as well as COBOL. Total provides an effective method for organizing and managing diverse data to make it both efficient and convenient for application programmers to maintain and retrieve the data for processing. Total performs in both a batch and on-line environment, with Cincom's ENVIRON/1 (Report 70E-132-02) available to handle the data communications.

As compared with IBM's IMS (Report 70E-491-01), Total offers a somewhat more simplified and certainly more streamlined approach to data base management, providing a more satisfactory solution for the majority of users with data base applications. Of course, it is incumbent upon each user to ensure that he is, in fact, within that majority. While IMS is still the most widely used data base management system in the industry, the gap between IMS and Total is narrowing rapidly. Although all versions of IMS currently installed outnumber the Total installations, there are now more users of Total than of the newest version of IMS (IMS-2, released early in 1971).

Total can manage virtually an unlimited number of data sets on an "integrated, non-redundant" basis and provides for association of each of these data sets with other data sets to form an integrated data base. Basically, Total is oriented toward simplifying application programming problems by handling complex relationships among the various data items that constitute the information base in a user's shop. Such complex relationships are quite common for most installations where multiple, but related applications, are implemented. One of the very nice conceptual features of Total is that users can start small, with perhaps a single application using two or three data files, and grow in modular fashion to very large and sophisticated data base/data communications structures with very little impact on the previously implemented systems.

The data structuring power of Total to handle complex relationships among data items is achieved because it embodies a network data structure. The network structure means that a data record may have any number of subordinate or "member" records—but, more importantly, each member record may also have any number of "owner" records. In hierarchical structures (e.g., IBM's IMS-2—Report 70E-491-01), an owner record may have many members—but each member can have only a limited number of owners.

Total is a widely used package for implementing data bases. It provides facilities for data base generation and accessing from COBOL, FORTRAN, PL/1, or Assembly language application programs and is operational on IBM System/360 or 370, Honeywell Series 200 or 2000, and UNIVAC Series 70 computer systems.

CHARACTERISTICS

SUPPLIER: Cincom Systems, Inc., 2181 Victory Parkway, Cincinnati, Ohio 45206. Telephone (513) 961-4110.

BASIC FUNCTION: Provides facilities for generation of a data base that permits automatic cross referencing among data records. A facility is also provided for accessing the data base from conventional application programs written in COBOL, PL/1, FORTRAN, etc.

ARRANGEMENT: The Total Data Base Management System is composed of several phases, one of which is also called Total. The basic system includes three phases: one is for generating the program for controlling data base structure, one is for pre-formatting the disk areas, and one is for controlling the access to the data base.

Special phases have been developed for particular purposes such as allowing dynamic loading of the appropriate phase at execution time under IBM OS/360, permitting data read/write only (no data base structural maintenance), reading data only, and generating master record addresses only (which speeds loading of the data base in some cases).

Total permits establishing two types of records—a single entry or master record and a variable entry record. Each group of records, of either type, forms a file (data set). Linkages can be set up that permit automatic retrieval of all variable entry records associated with a particular single entry record based on the linkage. A variable entry record can be part of many linkage paths or chains.

A data base is composed of multiple data sets or files. Linkages can exist between any master file and any variable entry file and, to a limited degree, between two master files. Multiple data bases can be established. A particular master file or variable entry file can be part of more than one data base. The multiple paths of access allowed by such a structure, called a network structure, simplifies the logic of application programs using the data. In the case of Total, it also reduces the amount of disk storage required to hold information by eliminating duplicate fields or records.

To one familiar with sequential, even hierarchical sequential, files, the benefits of a network structure are not immediately evident. It seems at first glance that the power of a network structure is limited because only one sublevel of linking is possible; i.e., master to variable entry. The real power of this structure lies in the fact that multiple master files can be established, each for a particular relationship.

An example is the universally accepted technique for illustrating abtruse points. However, let us stay away from the traditional bill of materials or order entry/inventory control applications to keep from getting lost in the details. Instead, let us take our example from an area dear to the hearts of our staff: the DATAPRO 70 index. For any EDP product, there are three principal attributes: the name of the supplier or manufacturer, the name of the

Total
Cincom Systems, Inc.

The network structure technique of Total allows the access paths to data to be different from those on the physical storage device(s). It is implemented by establishing two types of records; in the case of Total, these are called "single entry" and "variable entry" records. Access to single entry or master records is direct, according to the value of a control field. Access to a variable entry record is normally through a particular master record and a particular relationship, which is called a linkage path. There may be any number of paths to access variable entry records. Because relationships among data items can be specified directly, related items can be retrieved directly. A fuller exploration of these concepts is contained in the CHARACTERISTICS section of this report.

Total provides a simple and straightforward set of facilities for organizing a data base and for manipulating the data in the data base within an application program. To the application programmer, who can be coding in COBOL, FORTRAN, PL/1, Assembly language, or any other host language, accessing information in the data base is much like using a subroutine to calculate and return data values; file definitions and data structure definitions are not required for the information obtained from or written into the data base. All data base access from the host language is through CALL statements.

There are some very real and pertinent considerations that should be carefully explored before implementing a data base/management information system. In the traditional form of processing (i.e., sequential tape files), backup was readily attained by keeping the old master file and the transaction file. As direct-access processing has come into favor, the only way to obtain backup is to dump the master files, or data base, at periodic intervals and retain all transaction records processed since. Unfortunately, frequent dumps use up a lot of processing time. Also, special problems can occur in a real-time or data communications environment because no permanent record of transactions is automatically created.

A version of Total is available to handle such data communications environments environments with transaction logging. This facility, combined with periodic data base dumps, allows recovery from malfunctions. Certain features of Total also include dynamic data base logging which captures "before" images of data fields modified by transactions; this allows automatic restoration, in some cases, of the data base to its original form and processing of logged transactions up to the abort point at the operator's discretion.

Another general problem of data base processing is debugging new applications. Obviously, any application program that alters the data base must be thoroughly checked out before allowing it free access to the working data. One commonly used approach is to create a miniature test data base specifically for debugging the program. This approach introduces the secondary problem of checking out the test data base to make sure the program testing process is valid. While this is not a particularly difficult problem to handle with Total, it is

product, and the type of product. To find material within DATAPRO 70, each of these attributes may at one time or another serve as an entry into the index; i.e., a person may know a vendor's name, the name of a product, or the type of product and be looking for specific references under any one or all three of these. Postulating a file sequenced by vendors, and further assuming that a record is generated for each product that also includes the product type, the whole file would have to pass while comparing each entry to the key input if traditional, sequential processing were employed.

Progressive touches would include multiple keys for retrieving specific data items within several categories and batching multiple requests to reduce the number of times the file had to be passed. The throughput rate for such an arrangement would be slow. Retrieval logic becomes very complex in any extended application where more than one type of record is introduced into the system. The above arrangement would be fine if all we wanted was a listing of the products of particular vendors, especially if we wanted complete lists, or perhaps lists of all products of specific types.

To see how we can do the same task using Total, postulate one master file for vendors, another master file for product types, and a variable entry file (detail) naming the products. Linkages would be established between each vendor record and those product records associated with the vendor; other linkages would be established between each product type and the associated products. Each product record would then be linked with a vendor record and a product type record. To generate a list of products of a specific type, we would access the appropriate master file record, which would contain a pointer to the first product of that type on the product file. From there, it would simply be a matter of following the pointers from product to product until the last one were found. Return status from Total would let you know that the end had been reached.

If the product file were very large and the example were typical of the type of request processed against the file, the result would be efficient. But note that the only way to obtain an alphabetical listing of the products in the file would be to list the file. If the appropriate record addition logic were followed in the application program controlling the maintenance of the file, the file would be in that order or any other order desired; otherwise a sort would be required.

The above example hardly does justice to the concept of network structure of what Total can do for organizing and accessing a data base. Let us dream for just a second. Assume that we had access to complete user information for all data processing installations as to products used. To our data base of products we want to add usage information. The types and extent of information to be included can be rapidly reeled off by any marketing man. The problems of organization now become extreme because of the desire to access the information from so many directions. Again, we can postulate organizing it into one massive file, somehow reconciling the different record formats and contents and accessing it by passing a list of keys against the file and sorting the result—sort of a report generator gone mad. Using Total, you can identify these desired entry points and set them up as master files with multiple linkages to the detail records based on information needs. While definite benefits in processing speeds are available through Total for this application, don't overlook the equally important point that the job of the application programmer becomes much easier.

Data is transferred from disk on a block basis to an I/O buffer. Multiple files can share an I/O buffer, or independent buffers can be assigned. Data elements called by an applications program are moved immediately to the program area, releasing a shared buffer for use by other files. Any number of files can share one I/O area. Sharing of I/O areas is structured during data base definition and one that must be included in evaluating the time required to implement your applications. The logging facilities of Total provide a neat trace of the exact processing result of a program being tested.

One final but very serious problem of data base implementation in general is the impact of change. If a piece of data is used across a broad range of applications, then both the data processing impact and the corporate implications of change to the data or its structure can be significant in two areas. First, every program that uses the data item being changed will require, at a minimum, recompilation. Second, the data base and data structures will often be exposed to serious reorganization problems. This can involve the shifting of internal corporate responsibilities among user groups as well as large expenditures of data processing resources.

While the impact upon corporate organization is essentially a management problem (and one common approach to solving this problem is through establishment of a Data Base Administrator function), Total handles both of the EDP-related considerations very nicely by providing data independence at the data field or data element level. This means that new fields of data can be added to a record with no impact upon operational programs which used the old record in its old format. Total features integrated data base facilities, but with separate data files. This means that data within one physical data file may be logically related to data in a different physical data file. This approach minimizes the impact of physical change to the data base. For example, it is possible to physically reorganize a master data file which has records logically related to many other data files with absolutely no change or impact to the other data files.

The principal limitations of Total stem from one property: the structure defined is fixed, not dynamic, once the data base is loaded. Records can be added to and deleted from any of the data sets, but new data sets cannot be added, new relationships (linkage paths) cannot be established, and the disk storage area cannot be expanded without least partial regeneration of the data base, which entails reloading of all affected data files.

On the whole, however, Total is one of the most effective high-powered software systems in use today, and it is successfully displacing IBM's IMS-2 in numerous installations. In September 1972, Honeywell Information Systems announced an agreement to market the Honeywell Series 200/2000 version of Total through its own worldwide sales force, assuring further recognition and market acceptance for the already widely used Total system.□

generation stages. Master and variable-entry files cannot share the same I/O buffer. This facility for sharing I/O buffers provides the potential for saving a great deal of main memory space, but adds the burden of intelligent planning and possible future conflicts between space and performance as the applications processed against a data base grow and the combinations of files processed change.

Total allows you to set up a data base; it also provides the method for accessing the data base conveniently. However, you must write the application programs that determine what information is to be retrieved, and how the information is to be processed, and must perform all procedural processing required to maintain the data base.

The three topics relevant to discussing what Total can do for you are data base generation (DBGN), data base definition language (DBDL), and data management language (DML).

DATA BASE GENERATION: The DBGN program accepts the data base structural definitions to DBDL and outputs an assembly language program. After assembly, this module is catalogued as a subroutine to be core resident with the application program. After executing the FORMAT program, which formats disk storage, application programs can be run which include one of the access phases as a subroutine. The first application program to be run must be one that loads data into the data base; alternatively, the same program that normally would be used to add new records to a file can be used. It depends on the types of applications you are processing. Normally, there needs to be some sort of base to begin with.

Output from the DBGN program execution is a listing of the data base definition statements which serve as a guide to the contents and structure of the data base records.

A randomizing algorithm is used to calculate master record physical addresses based on the value of the control field. If duplicate addresses are calculated, a pointer is used in that record to show where the "duplicate" or synonym record is stored. Thus, the complete disk space allocated can be used. Once all space is used up, the data file must be reloaded with new parameters.

DATA BASE DEFINITION LANGUAGE: Writing the control statements for structuring the data base is not difficult. Essentially, it consists of a very short preamble identifying the data base by name and specifying whether concurrent update protection is required (to prevent conflicts and errors due to two application programs modifying the file simultaneously in a multiprogramming environment) and descriptions of each single-entry and each variable-entry data set. The exact format depends on the operating system Total is to function under, but in general the total number of logical records, record length, number of records per block, and number of records per track must be specified for each data set.

A part of the description of each data set is the name of the I/O buffer, the names of all linkage fields, and the structure of the records. Total processes only fixed-length records; i.e., all records in a data set must be of the same length. Only one format can be specified in single-entry data sets, but multiple formats are allowed in variable-entry data sets, each identified by a two digit record code. Linkages between a single-entry data set and a variable entry data set can be qualified by the record code, if desired.

Record structure is identified in a very straightforward manner by stating the name and size in bytes. No data format identification is required or can be specified. The data names refer to data elements, the smallest chunk of data that can be named for retrieval. Data elements may be subdefined to a maximum of 32 levels of structure (comparing favorably with COBOL). The data elements may or may not refer to individual fields used in an application program. For example, a complete data might be identified as a data element while the application program processed it as three fields (month, day, and year). Three separate elements could be ade out of the date if desired, but it would be unnecessary because the complete data is always transferred as an entity.

Comments can be easily incorporated in the data definition statements and are highly recommended to serve as helps to application programmers.

Total
Cincom Systems, Inc.

DATA MANAGEMENT LANGUAGE: DML provides the facilities for retrieving information an;passing it to an application program, as well as for opening and closing files, and for adding and deleting records from an existing data set. Absent from DML are any provisions for establishing new data sets or establishing new linkages among existing data sets.

DML functions at the CALL level in the host programming language. For data transfer operations, the parameter list includes an operation code, the name of the file (data set) to be used, the name of a field to hold the status returned at the end of each operation, the name of the master file record control field, the name of the I/O area, and a list of the data elements to be transferred. If variable entry files are being accessed, the linkage path and field are also identified. For some operations, such as opening or closing a file, not all parameters are required. The operation specified controls the parameters needed.

The exact procedures vary depending on the host programming language. The chief difference among the various languages supported is whether or not literals can be used in the parameter list.

A total of five sets of functions are provided: one for opening and closing files individually or via a list, one for serial retrieval of records from a single or variable entry file, one for resetting the serial record counters, one for working with single-entry (master) files and one for working with variable-entry files. Under IBM's OS/360, another set of functions is added for dynamically loading the various Total phases.

Serial processing refers to taking the records in the physical sequence in which they appear on disk storage, not sequentially according to the value of a control field. Serial processing of variable entry records can be according to actual layout on disk or according to any linkage path. Normally this would mean that there is some order to the records. Serial processing provides some efficiencies when all records in a data set must be handled.

Processing functions available for master files include reading, writing, record addition, and record deletion. All variable records linked to a master record must be deleted in a separate operation before a master record can be deleted. The space occupied by a deleted record is available immediately for reuse by Total.

A host of operations is available for working with a variable entry data set. All references to a variable entry data set are along a linkage path with a specific master record. Records can be read forward or reverse (i.e., starting from either the beginning or end of a chain), or a relative location can be specified for retrieval. Much flexibility is provided for adding a variable-entry record to a chain. A delete function is also included.

The use of the CALL subroutine method makes accessing data very much like the calculation of values in a subroutine. Instead of calculation, Total retrieves data. No file and data specifications are necessary in the host program other than identifying variable names where required. Because only a data element list is required in the CALL statement, modification of the data record structure to include new fields does not require that application programs referencing the affected files be changed.

MULTIPROGRAMMING: Total 4, the basic version of Total, can be used in a multiprogramming environment. Individual copies of Total appear in each partition or region. Only one partition at a time can add or delete records from a particular file; this is controlled through a lock established when a file is opened.

Total 5/6, the real-time version of Total, can operate in one of two modes: as a sub-task servicing all sub-tasks of a region, or as a stand-alone region servicing all tasks and sub-tasks. Total 5/6 can co-exist with multiple copies of Total 4 and the specialized phases. However, Total 5/6 and Total 4 cannot simultaneously have access to the same file. With Total 5/6, multiple buffers can be set up for each I/O area to provide flexible planning of I/O buffer sharing between files being accessed by multiple tasks and files being accessed frequently. Total 5/6 also provides transaction logging, including the "before" and "after" images of modified data records. This allows the data base to be reconstructed if required.

Under Total 4, one record will be held for each file until the record is written or another record is read. This prevents separate application programs from updating the same version of a record independently, which would cause the loss of a transaction. An application program attempting to access a file already having a record being proceed will stall until the record is written.

Under Total 5/6, one record is held for each file for each task. If subsequent access attempts to a "held" record cause application programs to stall, Total will monitor the occurrence of these attempts and will cause the record to be automatically released, allowing the application program to proceed, and a status indication will be returned to the first program indicating that reprocessing of the transaction involved is required.

A recently developed version, Total 7, provides additional facilities and increased modularity to Total 4. A Real-Time Version of Total 7 is scheduled to extend these facilities to on-line users. Both new versions will be available in addition to Total 4 and 5/6.

PERFORMANCE: There are several aspects to evaluating the performance of Total. The basic ones boil down to efficiency of disk space utilization, speed of data access, and speed of loading data into the data base. It is not surprising to find these elements interrelated.

Data is distributed randomly based on a control field and when two records calculate to the same location, a secondary location is used. Thus, all disk space becomes closer to full, the occurrence of synonyms (identical addresses) will become more frequent. This discussion pertains only to master files because direct pointers are provided for access to variable-entry files. Synonyms are placed on the track (or cylinder) as the home record if possible. Cincom quotes an average of 1.1 seeks per record, an excellent figure.

HARDWARE/SOFTWARE REQUIREMENTS: Total 4 is currently operational on IBM System/360 and 370 computers running under DOS or OS, Honeywell Series 200 and 2000 computers running under Mod 1 (MSR), Mod 2, or OS/200, and UNIVAC Series 70 computers running under TDOS or DOS. Host languages can be COBOL, FORTRAN, PL/1, and Assembly language. Total 5/6 is operational on IBM OS installations.

The amount of main memory required by Total is meager. The maximum required by Total 4 in addition to the application program and I/O buffers is only 8K bytes. If reduced subsets of the facilities are adequate, the requirement can be reduced to 3K bytes—which provides only read operations. Total 5/6 requires about 14K bytes.

PRICING: Prices vary with the target computer system. Total is available either on a purchase or rental basis. For IBM DOS installations, the purchase price is $24,500 and the rental price is $750 per month. For IBM OS installations, the figures are $28,500 purchase and $950 per month rental. Total 5/6 is available for IBM OS only and costs an additional $5,000 purchase or $100 per month. Unlimited support is provided for the first two months of installation at no extra charge. Total 7 pricing is not available at this time.

INITIAL DELIVERY: Early 1969.

CURRENT USERS: About 250 as of July 1972. ■

Environ/1
Cincom Systems, Inc.

MANAGEMENT SUMMARY

Environ/1 is a modern data communications monitor and task management system that makes good use of some of the most sophisticated and up-to-date programming techniques to be found in today's software marketplace. Importantly, Environ/1's initial development strategy recognized that the EDP user community was heading toward an approach to data processing that would include integrated data bases, with data base/data communications systems an important part of the ultimate picture.

Cincom is also the vendor of TOTAL (Report 70E-132-01), the leading alternative data base system to IBM's IMS. Thus, Cincom can supply a data base/data communications system that is integrated at the source and execution level for all users of IBM System/360 or 370 DOS, OS, DOS/VS, or OS/VS computer systems. (For a comparative examination of data base/data communications systems, the reader should refer to Reports 70E-491-01 and 70E-491-02, on IBM's IMS and CICS, respectively, and also to other data base systems and communications control routines listed in the Index.) Cincom's position as a single source for a data base/data communications (db/dc) software system considerably elevates the importance of Environ/1.

Typically, few data processing executives give proper recognition to the complexities of telecommunications when they develop their companies' plans for terminal-oriented computer networks or for distributed intelligence systems in which remote locations have processing capabilities that are tied together either by direct links or via a central processor. Many of the standard operating system facilities provided for the central computers require extensive interfacing in order to be operated from a remote on-line station. These communications-oriented considerations include system start-up and cycle-down, file and communication access methods interfaces, system recovery procedures, security measures, etc.

From an overall point of view, it is commonly desirable for telecommunications to be handled as one of a number of jobs under control of the computer's operating system, instead of dedicating an entire processor to controlling the multi-station network. As the number of terminals and concurrent tasks in a telecommunications network increase, the requirement for allocating and accounting for the system resources demanded by each user program becomes very complex. Solutions to resource contention depend upon the application of highly sophisticated programming techniques. These include the establishment and maintenance of message queues, assignable task priorities, program swapping into and out of main memory, sharing of common code through program "re-entrancy," etc. Furthermore, hardware changes to the central com- ▷

Environ/1 is a communications monitor and task management system that runs on any System/360 or 370 DOS, OS, or virtual storage system and uses advanced paging techniques to achieve impressive performance with comparatively small main memory requirements. Its interface with Cincom's TOTAL elevates its importance.

CHARACTERISTICS

SUPPLIER: Cincom Systems Inc., 2181 Victory Parkway, Cincinnati, Ohio 45206. Telephone (513) 961-4110.

FUNCTION: Environ/1 is a general-purpose data communications monitor that operates in a partition or region of an IBM System/360 or 370 under DOS, OS, or their virtual storage counterparts to control multiple on-line user terminals and applications. By consolidating all of the communication interfaces and I/O and control functions required in the telecommunications network, Environ/1 isolates the user's application programs from the communication environment. Environ/1 also provides paged or "virtual memory" support for both COBOL and Assembly languages, while FORTRAN and PL/1 are supported in a non-paged mode. Thus, on-line applications can be developed without significantly greater difficulty than development of similar batch programs.

OPERATION: As the interface between user-written applications programs, the operating system, and communications requirements, Environ/1 operates with the highest priority in a partition or region that contains about 30K to 33K bytes for Environ/1, about 2K to 3K bytes for each typical on-line application program, about 10K bytes for message buffers and interface code for the terminals, and a 3K to 5K "context" for each program that includes specific parameters for each user's version of that program. Of particular significance in the operation of Environ/1 is that only 2K to 3K bytes of main memory are required for each user's version of an application program, no matter how large the program really is; and only 30K to 33K bytes of main memory are required to hold the resident portion of Environ/1's several hundred thousand bytes of code. These small main memory residence areas are made possible by extensive use of paging and a virtual memory mode of operation, not only for Environ/1's own code, but also for user programs.

A variety of sophisticated techniques are employed that automatically divide each user's application area in main memory into 512-byte pages. Cincom notes that most competitive telecommunications monitors, including IBM's CICS, have been designed to handle pages from about 2K to 6K bytes in length, which creates an unnecessary overhead burden for data transmission and storage in an on-line environment. Also, studies made of on-line systems indicate that less than 10% of the code in most on-line programs is accessed more than 80% of the time; this makes it possible for very large programs to be executed efficiently from a much smaller area in main memory. Specifically, by allowing as much (or as little) as 10K bytes of a user's program to reside in main memory, Environ/1 provides good support for application programs that typically can be in excess of 100K bytes long. ►

▷ puter or an individual terminal, as well as changes in the network configuration or applications programs, must be possible without major system reprogramming.

Cincom provides answers for all these problems with either a "basic" or a full-blown "comprehensive" version of Environ/1. These two versions offer essentially the same functional capabilities, but the full-blown system provides higher throughput for a larger terminal network and is more effectively used on large processors. The basic version, however, can also be used on large processors when the user wants to implement a terminal network of limited size. Cincom claims that Environ/1 can more than hold its own against competitive systems, including IBM's CICS, by supporting an equivalent telecommunications network on a central processor one size smaller and with one-half the main memory. For example, the workload handled by CICS on a 512K-byte IBM 360/50 could be handled by Environ/1 on a 256K-byte IBM 360/40 – and with better throughput, according to Cincom.

One of the key reasons for the reduced main memory size is that Environ/1 is a "virtual system." This concept, of course, is not new or even unique – but Environ/1 does represent one of the best practical applications of virtual memory yet developed. Virtual memory techniques are based upon the use of paging; application programs (and Environ/1 itself, for that matter) are segmented into manageable pieces that are (1) small enough to minimize the handling of unnecessary code, and (2) large enough to contain a usable portion of a program. The selection of page size is of prime importance in optimizing the performance of a paged or virtual system, and Environ/1 uses a 512-byte page. Competitive systems in which paging has been employed – although without true virtual memory capability – generally use page sizes of about 2K to 6K bytes.

For typical on-line applications programs, Cincom believes 512 bytes is optimum. Furthermore, the uniform segmentation of available main memory on regular 512-byte boundaries avoids the problem of "checkerboarding", where sections of main memory are unusable by other programs because the pages for those programs are too large to fit. Checkerboarding (also often called "memory fragmentation") wastes main memory, with unusable areas easily reaching 50% of all the memory available for applications programs, depending upon specific program segment sizes, program logic design, system activity, etc.

With Environ/1's virtual memory scheme designed and applied in the days of real memory computers (e.g., the System/360), the question naturally arises as to the effect of running virtual memory within virtual memory (e.g., on the System/370). We're happy to report that there's no problem with this.

The 512-byte Environ/1 pages are paged in a software ▷

► The demand paging algorithms in Environ/1 keep track of the usage of each individual page and select the least-used pages to be overlaid by new pages brought into main memory upon user demand. Active pages remain in main memory as long as they continue to be referenced. This paging scheme minimizes the overall need for swapping, and then operates on 512-byte pages to minimize the amount of code that is handled with each swap.

Re-entrancy is also used by Environ/1, not only for the Environ/1 system itself, but also for most user programs. Toward this end a special terminal-oriented COBOL, known as TEBOL, and a set of procedures for assembly-language programs have been developed by Cincom. This allows COBOL and assembly-language programs to be developed that have "pure" or re-entrant pages of code which are used in read-only fashion by all users, together with a "context" or non-reentrant block of code that contains pointers to the main body of the program and other specific parameters associated with each user's version of the program. TEBOL automatically produces re-entrant object code in 512-byte pages. In assembly language, the user must insert a "page" card into the source code and use appropriate programming conventions to achieve re-entrancy. The page card activates a macro facility that causes the re-entrant object code to be segmented into 512-byte pages.

It should be noted that FORTRAN and PL/1 programs can also be run under Environ/1 in traditional nonpaged mode, although with considerably less efficiency than corresponding COBOL or BAL programs. A main memory area large enough to accommodate the entire FORTRAN of PL/1 program must be dedicated, in addition to other main memory requirements in the Environ/1 partition or region. The entire program is then locked into main memory during its execution. This technique can also be used in a test environment for COBOL or BAL programs if desired. It is also an excellent means for users to quickly convert from batch or other on-line systems with minimum transition impact.

Another particularly significant aspect of Environ/1 usage is the fact that statistics are compiled and maintained about resource utilization, including device and channel utilization. Statistics accumulated by the Environ/1 system measurement facility allow the user to determine where the system bottlenecks are and provide a basis for "fine-tuning" the system to achieve the best possible throughput.

Other features of Environ/1 include:

- System expandability across the model range of IBM System/360's or 370's from the Model 25 through Model 195 (including DOS to OS conversion) without reprogramming or converting the applications programs.
- Automatic restart for failures in the application programs, the Environ/1 control system, or the entire computer system.
- Data integrity, including a complete audit trail and automatic checkpoint/restart capability. This is an optional feature and typically degrades throughput by about 5%, according to Cincom, for normal checkpoint usage.
- Storage device independence, allowing the user to develop programs without regard for main memory limitations or device types. This feature fully supports ►

scheme, and this operates within, but independently of, IBM's 2K or 4K page hardware scheme. This raises two more questions: (1) how is the software "pointer" or "reference" page handled, and (2) what is the effect upon real memory usage and system overhead?

Again, tests happily show no problems. The necessary software reference page does not have to be fixed in real memory because it's called by the application program. And overhead is not increased because Environ/1 has a locality-of-reference algorithm. This algorithm dynamically places commonly used Environ/1 pages into contiguous logical memory, which in turn means within a hardware page. Thus, the system is self-tuning.

One significant aspect of Environ/1 usage is that in order to maximize throughput and provide a high level of task control and application flexibility in the communications network, special file facilities have been developed for Environ/1. The native Chained and Queued file access methods provide highly useful tools for such applications as message switching, high-volume data transfer from buffered terminals, and control of "batching" in such applications as remote job entry (RJE) and data collection. In addition, Environ/1 and TOTAL provide a powerful, self-contained data base management and telecommunications environment without any significant need for standard IBM file structures. It is possible, however, for Environ/1 to be interfaced to any file access method, but with performance taking on the performance level of that access method.

A new offering in Cincom's intergrated product line is a multithread version of TOTAL that is available with the new 7.0 release of the Environ/1 product line. This version of TOTAL was created exclusively for use with Environ/1 in a db/dc system, and Cincom claims that it provides up to 30 percent more throughput in data base accesses than previous versions, yet retains complete security and control of the data base. Like the single-thread version, it is completely recoverable and restartable under the optional Environ/1 checkpoint/restart feature.

With about 75 Environ/1 systems installed to date and TOTAL moving rapidly toward 500 installed systems, Cincom clearly has a market within its present TOTAL customer base for extensive further installations of Environ/1, as well as for new customers who find that Environ/1's advantages outweigh those of the various alternatives.

Thus, there are a number of data management options available in conjunction with the use of Environ/1: TOTAL can be used as the data base manager; the Chained and Queued file access methods can be used with Environ/1 to control the execution of data-dependent tasks (as with TOTAL, these files are automatically restartable); or IBM's standard access methods can be used in stand-alone fashion or in conjunction with TOTAL.

the IBM 3330 Disk Storage Facility with Rotational Position Sensing (RPS), and treats the addresses of all programs and data files as virtual extensions of main memory.

- Interface to Cincom's TOTAL Data Base Management System. (Report 70E-132-01).

A System Driver is also provided to simulate the Environ/1 terminal input message traffic for testing purposes to predict the effect upon the system of peak loads and to identify system bottlenecks. This simulation program can accept a previous day's transactions from the system log tape for high-speed peak load testing, or it can accept batch partition test data to debug user application programs.

Other support programs for Environ/1 include COBOL-XT (or TEBOL 2), a Dump Utility Program to dump main memory with diagnostic messages, and a Log Analysis Program that logs all significant system events and provides a detailed breakdown of system resource utilization. This system measurement facility can help the user adjust the system for better response time.

RELEASE 7.0 ENHANCEMENTS: Listed below are some of the enhancements made to Environ/1 with the advent of Release 7.0:

- $ISSLOG1, the Log Print Utility program, now prints TOTAL log records selectively from the common log tape. Selections printed can be varied and/or combined.
- The linkage editor for the on-line program data set was re-written. It provides DOS users with faster linkage edits and the OS user with disk work files that enable him to create program data sets of any size without increases to memory-resident tables. The modified internal structure allows addition of single programs or pages and changes or deletions to the program data set.
- Chained and Queued File Creation and Processing has been upgraded; it can now provide a flexible, easy-to-use queue file management system.
- TOTAL can now be signed on at execution time in a Read-Only mode. This lets users bring up read-only inquiry processing networks while TOTAL in batch mode simultaneously updates files.
- The log tape in DOS can now be set to alternate drive switching. Forced end-of-volume (FEOV) processing is compatible with DOS drive switching.
- A new RSTMSG macro allows users to define, at task sign-on (or any other) time, message conditions at restart time.
- The system generation macro CSIGIN is replaced by CSIGEN, which is expanded to allow the user to name modules at his discretion and specify explicit terminal device support.
- Sign-on service routines have been rewritten to provide improved usability and are now coded to be device-independent and free-form.
- TOTAL calls to "DATBAS" are now supported in Environ/1 assembler macros with the introduction of the TOTCAL macro.

Environ/1
Cincom Systems, Inc.

▷ The original architecture and development of Environ/1 was done by a group of telecommunications experts who drew upon experience acquired at IBM and with other telecommunications systems. The development costs now exceed $5 million, and Cincom is continuing to expand the system. Since first offering the package in 1971, Cincom has integrated Environ/1 and TOTAL, significantly expanded the terminal support to a wide range of devices and configurations, implemented an "unbundled" modular price strategy that permits users to pay for only those capabilities that their systems require, and installed over 50 additional systems.

In the latest release (7.0) of Environ/1, Cincom announced an improved and expanded COBOL compiler (COBOL-XT, or TEBOL 2), new system generation facilities for ease of installation and use, and virtual storage (VS) operating system compatibility. Cincom also claims that the engineering of this latest level provides it with an even more useful level of user error detection facilities and hardware/operating system error recovery.

USER REACTION

Based on its original technical evaluation of Environ/1 in early 1972, the Datapro staff concluded that the system looked very good and shaped up as an attractive choice for many companies with a telecommunications monitor requirement. Even so, when Datapro contacted Environ/1 users at that time it was with some astonishment that the following statements were received: "a data processing dream come true" – "most sophisticated commercial system available" – "has an incredible amount of power and is easy to use" –"system is golden." Granting that the technical virtuosity of Environ/1 is impressive and that many of the users contacted were sophisticated technical personnel who were likely to be enamoured of such a fancy system, the comments must nonetheless be considered an impressive measure of the effectiveness of Environ/1, particularly since a number of them came from management personnel with budget responsibility.

On the other hand, these early users stated that they felt that the level of COBOL then implemented in TEBOL (Terminal-Oriented COBOL) was too restricted a subset of ANSI COBOL. While TEBOL 1 was generally upward-compatible with most ANS COBOL compilers, the user had to live without the ALTER verb and accept restrictions upon the usage of the PERFORM verb. However, Cincom points out that these restrictions are actually helpful when TEBOL's paged, re-entrant nature is considered. In any case, the new COBOL-XT, or TEBOL 2, is vastly improved, as outlined in a paragraph in the Characteristics section.

Users of Environ/1 interviewed for this revised report are just as strongly satisfied as ever with the package. One user reported liking Environ/1 because "it's fast, easy to use, and easy to write for." Another user favors Cincom's "technical virtuosity" and states that the package is extremely powerful and fast. All of the users interviewed ▷

▶ COBOL-XT: This enhancement to Environ/1 was announced with Release 7.0 in July 1973. COBOL-XT (known to some as TEBOL 2) replaces and improves upon the previously available TEBOL 1 option. It uses paged, re-entrant code (unlike ANSI COBOL), and Cincom states that an ANSI COBOL programmer can learn COBOL-XT in 2 to 3 hours. The only reason it has to be different from ANSI COBOL at all, and thus learned by programmers, is that certain ANSI COBOL commands wouldn't be very sensible when issued under a paged, re-entrant compiler. Here are some of the new and specific features of the COBOL-XT option:

- All assembler-language macro facilities are available.
- Numeric and alphabetic class tests are included.
- Identification and Environment Division entries for documentation purposes are allowed.
- Data Division forms (as well as previous TEBOL syntax) are permitted.
- The COPY verb has been added, permitting COPY code stored in the user's source library to be inserted into source programs.
- 88-level entries are now permitted, and condition name testing is now allowed in IF statements.
- IF facilities have been enhanced to permit verbal (non-algebraic) test forms and ELSE, OTHERWISE, redundant NOT, and NEXT SENTENCE processing.
- Arithmetic verbs are now accepted with the ROUNDED option.
- The COBOL-XT compiler allows a generated job step execution under OS.

HARDWARE/SOFTWARE REQUIREMENTS: Environ/1 runs in a partition or region of as little as 45K bytes on an IBM System/360 or 370 under DOS, DOS/VS, OS, OS/VS1, or OS/VS2. Environ/1 itself occupies about 30K to 33K bytes of main memory, each different application program requires about 2K to 3K bytes, and a terminal interface and buffering area contains BTAM or other terminal device support code of about 10K bytes. A context area must be associated with each user's version of an application program and normally requires about 3K to 5K bytes. Any terminal supportable under BTAM will be supported under Environ/1 at no additional charge, while non-standard devices will be supported for a one-time additional fee; contact Cincom for details. The full size of Environ/1 is several hundred thousand bytes and fits easily on one pack of an IBM 2314-type disk drive.

PERFORMANCE: On a dedicated 64K-byte IBM 360/30 under DOS, with up to 30 busy terminals (1 message every 20 seconds) or 40 to 50 "walk-up" terminals, Environ/1 can support an average of 1 message per second, assuming about 150 characters per message, a formatted output display of about 300 characters, and about 6 to 8 disk accesses associated with the processing of each message. With similar message assumptions on a dedicated 128K-byte IBM 360/40 under DOS and up to 70 busy terminals (1 message every 20 seconds) or up to 140 "walk-up" terminals. Environ/1 will support an average to 2 to 3 messages per second – with one such installation experiencing a real-life throughput of 6 messages per second. Of course, all telecommunications message processing throughput estimates depend upon the channel configuration, and the system ▶

Environ/1
Cincom Systems, Inc.

▷ also have TOTAL and have integrated data base/data communications systems. An impressive aspect of this form of usage is the extreme parsimony of main memory as compared with the requirements of alternative software systems. For example, one DOS system uses 45K bytes for Environ/1, 18K for TOTAL, and 17K for TOTAL's buffer space, yet runs 4 tasks concurrently in 18-22K. About 100K seems to be entirely adequate for the combination of TOTAL, Environ/1, and the user's application in a DOS environment.

There seems to be some division of user opinion at this time regarding the quality and responsiveness of Cincom's on-site support. While a large OS user cited "fantastic vendor support at the installation of a relatively complex system", another user was critical of Cincom's post-installation response time to service requests, commenting that "Cincom is excellent in terms of (technical) capability, but their response times could improve." Still, users could find no fault with Environ/1 itself.

Cincom acknowledged some intermittent post-installation service response-time problems during the past couple of months. The firm told Datapro that this temporary situation was caused by an unprecedented acceptance rate for Environ/1 (orders during the past three months were triple the order rate for any previous period). Cincom is responding with accelerated staffing and training of its systems engineering group in order to meet all field support requirements.

In summary, Environ/1 lives up to the claims made by Cincom and is certainly a legitimate package that has ▷ much to offer the user with a telecommunications monitor requirement. The system facilitates the installation of a terminal-oriented network and is designed to provide the highest obtainable throughput in the smallest practical amount of main memory. Environ/1 is especially attractive when used with TOTAL from the same vendor. □

► **must have enough I/O simultaneity to keep the disk access queues from becoming a bottleneck and tying up the system. The above figures are based upon doing everything to Environ/1's best advantages, using either COBOL-XT or paged assembly-language programs. Other access methods or languages will reduce throughput, in some cases considerably, from the figures given above.**

As a comparison, Cincom estimates that the maximum CICS/OS throughput is limited to about 3 messages per second, while a hefty IBM 360/65 or 370/165 with an appropriate channel configuration can handle 20 to 30 messages per second under Environ/1. (For another comparison, the massive SABRE system can support 50 to 60 messages per second from a network of about 1200 terminals.)

PRICING: Environ/1 is available either on an indefinite paid-up lease or on a monthly lease basis, with title retained by Cincom for paid-up or "purchased" systems. A basic systems engineering allowance is provided with each installation. Additional SE support for customized work is separately charged. A one-sixth discount on the paid-up lease price is given for multiple installations. The prices and support allowances are summarized in the table below.

INITIAL DELIVERY: June 1970.

CURRENT USERS: About 75 as of January 1974. ■

Price Information*	Monthly Rental		Single-Use Charge		Annual Mainten-ance Charge	Man-Days of On-Site Support	
	DOS	OS	DOS	OS		DOS	OS
Basic Environ/1 (includes TOTAL interface)	400	600	22,250	29,500	1,500	4	5
Enhanced-performance Environ/1 with increased multi-thread capability (includes TOTAL interface)	700	1,000	28,200	39,500	1,500	4	5
Optional Features:							
Checkpoint Restart/Recovery	200	200	12,200	13,600	750	2	2
COBOL-XT (TEBOL 2)	200	200	10,600	11,200	750	4	5
Queue & Data Chain Management	175	175	6,500	6,500	750	2	2

* Prices include indicated number of man-days of on-site support, but such support must be requested within the initial 60-day use period. All prices and support are subject to change on a 30-day notice basis. Systems can be upgraded by applying the price differential currently in effect. The Queue & Data Chain Management feature cannot be obtained with Basic Environ/1. Multiple-installation discounts are available only under the single-use payment plan. For DOS/VS and OS/VS users, all prices of the VS versions of Environ/1 and their optional features are about 25% higher than the non-VS prices shown, with an across-the-board annual maintenance price increase of $250.

3705 Performance Packages
Comm-Pro Associates

MANAGEMENT SUMMARY

Comm-pro's four 3705 Performance Package programs, PP01, PP02, PP03, and PP04, can be used singly or collectively to save money in various ways for users of the IBM 3704 or 3705 Communication Controllers. The packages run on the 3704 or 3705 under the 270X Emulator Program.

PP01 is the Automatic Speed Select program. It selects the appropriate line speed oscillator for a terminal automatically, making it possible to use 110, 135, 150, or 300 bps terminals on a single line interface and thus reducing the number of required modems and line sets.

PP02 is the Code Conversion package that automatically translates Teletype code either to IBM 2741 BCD code or to Correspondence Code. With Code Conversion, Teletype paper tape can be supported under IBM's TSO (Time-Sharing Operating System). Used in conjunction with Automatic Speed Select, ASCII or 2741 devices can share one line interface. The packages make it possible for a user to save money on his terminals by supporting, say, APL on Teletype terminals instead of 2741's. The user can also support CRJE (Conversational Remote Job Entry) from devices that IBM doesn't normally provide for, or he can use the same terminals for both APL and CRJE.

PP03 is simply the designation for the combined PP01 and PP02 packages.

PP04 is the Network Facilities package. It is effectively a software terminal multiplexer, permitting communications lines to contend for System/360 or 370 unit addresses. It gives equivalent dial-in capability to hard-wired terminals. Automatic Speed Select and Code Conversion are included as selectable options within PP04.

The introduction of the IBM 3704 and 3705 Communications Controllers represented IBM's sanction of the front-end communications processing concept. The 3705 (or the newer, scaled-down 3704) would replace the former hard-wired 270X controllers and relieve the central processing unit of mundane communications tasks. The new programmable controller would also buffer input and output, perform necessary code conversions, and appear to the CPU very much like a normal I/O device. The IBM program to do all of this would reside in 3704 or 3705 storage and be called NCP (Network Control Program).

But delivery of NCP has been seriously delayed. It was to have been delivered by the summer of 1973, but has officially slipped until at least year-end, with support for TSO schedule for 1974. (Moreover, APL support is not currently planned for inclusion in the NCP concept.) Meanwhile, 3704 and 3705 users are operating in a 270X emulation mode, using the Emulator Program.

The Comm-Pro packages considerably broaden the scope of the 270X Emulator Program, but cannot relieve the CPU of housekeeping tasks, as NCP will someday do. But ▷

Comm-Pro's Performance Packages for the IBM 3705 Communications Controller can reduce line and modem requirements, permit use of lower-cost terminals for some applications, and provide a System/360 user with facilities that IBM offers only to System/370 users.

CHARACTERISTICS

SUPPLIER: Comm-Pro Associates, 638 14th Street, Suite 700, Manhattan Beach, California 92066. Telephone (213) 376-1344.

BASIC FUNCTION: The four programs in the 3705 Performance Package enhance the IBM 270X Emulator Program for the IBM 3704 or 3705 Communication Controller. PP01 provides automatic speed (baud rate) selection. PP02 is a code conversion (Teletype to 2741 BCD) program. PP03 is simply the former two programs combined. PP04 is a network facilities simulator, effectively a line multiplexer or concentrator, with code conversion (PP02) and speed selection (PP03) built in. Only one type of code conversion (translation) is allowed per package.

Depending upon which of the four packages is used, lines and/or modems can be reduced, terminal use can be expanded and made application- and language-independent, low-cost terminals can be used where otherwise prohibited, start/stop terminals can be run at higher speeds while using ASCII, line address positions can be eliminated, terminals can access different applications in the system, effective networks can be generated, and sometimes conversion to NCP or to a System/370 computer can be forestalled or avoided.

OPERATION: The packages are installed using instructions, generation macros, and object desks provided by the vendor. Changes to the IBM code modules are minimal and do not affect the installation of PTF's from IBM. Storage in the 3704 or 3705 must be provided in accordance a simple schedule, and two data sets must be moved from tape to disk. Only one macro is changed during Stage 1 Sysgen, and this results in automatic inclusion of the extended feature modules during Stage 2 Sysgen. Stage 1 macros supplied with the system allow installation of extended features on a line-by-line basis to facilitate testing.

PP01, Automatic Speed Selection, or ABR (for automatic baud rate), allows the bit rate for start/stop (asynchronous) terminals to be selected after a terminal has been connected. Thus, Teletype-compatible terminals can operate at 10, 15, and 30 characters per second using the same interface and telephone number. The operator simply sends an ABR character as soon as the telephone connection has been established. Characters preceding the ARB character, and the ABR character itself, are not passed on to the host computer system. To inform the host system of the line speed for billing purposes, the ABR character can be retransmitted. Only oscillators present on the IBM Type 2 Scanner and also present in a selection table defined during Sysgen can be selected.

Code Conversion, PP02, is simply the conversion of ASCII transmission code to 2741 BCD transmission code or, optionally, to 2741 Correspondence Code. Thus, Teletype Model 33 terminals can be used with systems that do not normally support them. In order to achieve full compatibility, certain functions are also translated. For example, Carriage Return on a Teletype unit sends the 2741 New Line Sequence to the host system, and a Line Feed code is transmitted to the Teletypewriter to perform the function inherent in 2741 operation. The function associated with activation of the 2741 ATTEN key when the keyboard is in the unlocked state is simulated. The ➤

3705 Performance Packages
Comm-Pro Associates

conversion to NCP means adding more memory to the 3704 or 3705 and also converting to new systems and methods (e.g., TCAM, VS). IBM has decreed that System/360 users must convert to a System/370 before they can obtain NCP. Thus, the Comm-Pro packages can save line, modem, and terminal costs for all 3704 and 3705 users, eliminate some conversion efforts, and extend the useful life of some System/360 installations.

Comm-Pro is a three-man partnership formed in March 1973. The principals possess impressive experience in communications software and turnkey communications design, implementation, and operation. They average over 10 years of experience in this field per man. It is Comm-Pro's present plan to operate in turnkey, consulting, and software development fields. The firm is also well equipped to handle custom modifications to communications software from IBM and other vendors.

USER REACTION

Comm-Pro provided Datapro with a list of user contacts at half of its current 12 customer installations. The users interviewed had only nice things to report about the package's cost-effectiveness, ease of operation and installation, and vendor competence and support.

According to one user, the full PP04 package provides the capability to dial only one telephone number and then select either APL or CRJE, as well as the ability to connect into CRJE from a Teletype terminal (which IBM software doesn't permit). For this user, using the package makes it unnecessary to dedicate a line address to a particular system, effectively doubling the number of lines. It thus saves dollars on line positions and modems and permits the use of cheaper terminals. This user keeps the IBM 270X Emulator and PP04 in the same library concurrently for backup. Another user reports that anyone who has ever generated a 3705 system before can learn PP04 in a half day and then generate the system in 15 minutes. In use, the package is said to be easy to install and to lessen the amount of coding required by the user.

The full-scale package gives users NCP-like features at a time when NCP is still undelivered, and also gives them to a System/360 user, who consequently may not need to convert to a System/370 as early as he once thought necessary.

In addition, the first-mentioned user says that the Comm-Pro packages enable him to hold his 3705 storage at 16K bytes, while the proposed NCP would require at least 40K in the 3705. A System/360 user reports using the packages to run ASCII terminals at 300 bits per second. For another user, the package made it possible to use a 3705 instead of a proposed second 2703 and an additional multiplexer channel on the CPU.

Datapro feels that users with communications network problems of the type addressed in the foregoing paragraphs should investigate the Comm-Pro packages, especially when standard 270X conventions have them backed into a corner and/or they're wary about adopting NCP. □

Teletype BREAK key is used in the usual manner, and Teletype paper tapes can be produced.

When PP03, which combines the PP02 and PP01 programs, is installed, conversion is invoked only for 110, 150, and 300 bps line speeds. Thus, Teletype-compatible terminals operating at 10, 15, and 30 characters per second can use the same lines as IBM 2741 terminals operating at 134.5 bps and using a different protocol.

PP04, the Network Facilities Package, equips the 3705 Emulator Program with the ability to switch a low-speed line interface so that it can communicate with different host System/360 or 370 subchannel addresses. Any asynchronous terminal can then use any start/stop line interface to gain access to any applications system in the host 360 or 370. One telephone rotary group, for example, could support multiple host systems and different terminal types.

Using PP04, the terminal user enters a character after dial-up but prior to communication with the host system to identify a specific subchannel pool. Thus, a single line interface could be connected to TSO, APL, or ATS, depending on the user's selection. The available choices are established at Sysgen, and the feature can be used by hard-wired terminals.

To handle contention for subchannels in each pool, PP04 uses first-come, first-served priority; any number of line interfaces may attempt allocation in the pool. Speed selection (ABR) is included, and operates as described above. The system can be configured so that TWX and 2741 terminals attempt subchannel allocation in different subchannel pools.

Code conversion, also described above, can be included with PP04. Finally, the package can supply "system down" or "system busy" messages to be associated with each Sysgened subchannel pool.

HARDWARE/SOFTWARE REQUIREMENTS: An IBM System/360 or 370 computer capable of supporting a 3704 or 3705 Communications Controller is required. The 3704 or 3705 must have at least 16K bytes of storage, a Type 1 Channel Adapter (required by the Emulator Program), a Type 2 Scanner, and, for ABR, a 110-bps oscillator plus an additional oscillator for each line speed used. The Emulator Program for the IBM 3705 is required (any level except version 1, modification level 0).

PP01 requires 218 additional bytes plus 4 more bytes of communications controller storage for each scanner. PP02 requires exactly 728 bytes of additional 3704 or 3705 storage. PP03, naturally, combines the requirements of PP01 and PP02. PP04 requires at least 2218 additional bytes in the 3704 or 3705 plus 4 more bytes for each scanner. At most, PP04 imposes the foregoing requirements plus those of PP02 when Code Conversion is included. Naturally, PP04 has the same oscillator requirements as PP01. Each program also makes minor changes and/or additions to IBM routines and macros.

The packages support a wide range of interactive terminals including the IBM 2741, Teletype and Teletype-compatible units, and equivalent CRT's.

PRICING: One-time prices for the packages are as follows: PP01 – $400, PP02 – $1,000, PP03 – $1,200 (a saving of $200 on combined PP01 and PP02), and PP04 – $3,500. Maintenance is provided, as are manuals, installation instructions, and modifications to any code affected by any software change from IBM.

INITIAL DELIVERY: The first three programs were installed in April 1973. PP04 was installed the following month.

CURRENT USERS: As of July 31, 1973, there were 12 users of Comm-Pro packages, mostly PP04. ■

GMT
Computer Information Management Co.

GMT (Generalized Multi-Tasking system) is a teleprocessing control system for IBM System/360 and 370 computers under DOS, OS, and their virtual storage counterparts. It is macro-generative and thus capable of supporting varied configurations, applications, and terminal types.

MANAGEMENT SUMMARY

GMT (Generalized Multi-Tasking system) is a software system that interfaces users' on-line application programs to the IBM DOS or OS operating system. It is aimed at eliminating about half the programming effort and much of the overall development time for new applications.

The GMT system also runs under IBM's virtual storage counterparts to OS and DOS, and programs written under GMT/DOS need only be recompiled to run under GMT/OS. This is because the functional layouts for the two versions are very similar.

GMT manages all teleprocessing resources, including all terminals and communications lines, user files, storage, and program scheduling and execution.

The system is macro-generative, meaning that the user generates only those modules he needs to meet his installation's particular requirements. GMT keeps storage requirements at a minimum, and because user applications programs are interfaced to it through standard CALL statements, GMT remains largely independent of the programs. This means that the user can modify and add programs without modification to GMT.

The package's macro-generative nature also means that new terminals, new applications, and new task areas for program execution can be added almost at will, permitting planned growth with predictable results.

GMT's flexibility is complemented by its ease of use. It is said that programmers inexperienced with on-line systems can learn to use GMT within one week. No special knowledge other than that imparted by the vendor, CIM, as a standard part of its installation assistance is required. CIM maintains that on-line programming under GMT is easier than batch programming.

These, in summary, are the system's advantages, as represented by the vendor:

- Elimination of more than half the programming effort for on-line systems.
- Extensive test facilities that significantly reduce the usual development and implementation time for on-line applications.
- Ease of use.
- Application and terminal independence, which lend the system to multiple applications.
- Orderly, simplified system expansion because of the macro-generative facility.

▷

CHARACTERISTICS

SUPPLIER: Computer Information Management Company, 325 Oak Plaza Building, 3707 Rawlins Street, Dallas, Texas 75219. Telephone (214) 526-4280.

BASIC FUNCTION: Software teleprocessing control system for on-line data entry and inquiry applications. GMT can be used on IBM System/360 and 370 computers under DOS, OS, and their virtual storage counterparts.

OPERATION: The key functional features of GMT are as follows:

- The system is macro-generative; it can incorporate the specific features required by its environment.
- GMT is application-independent.
- Multi-tasking is possible for any number of programs, even in high-volume applications.
- Testing facilities and terminal simulation are available in both on- and off-line modes.
- The numbers and sizes of task areas are determined by the user.
- GMT supports any valid mix of communication lines and remote and local terminals.
- The system has inherent restart and recovery features.
- A command terminal is supported to give the user dynamic control of the system.
- GMT/DOS interfaces applications programs written in Assembler Language or COBOL; GMT/OS interfaces with any language supported under OS.
- GMT supports CRT display terminal paging.
- Multiple accounting routines are provided within the system to measure performance and use of system resources.
- Resources are scheduled and allocated to optimize efficiency of use.
- User programs can be core or disk resident, reentrant, reusable, or relocatable (or can employ none of these techniques).
- A special GMT resource scheduling algorithm places a program into the smallest available task area for processing.

➤

- Multiple-terminal support.
- Ease of conversion, both from GMT/DOS to GMT/OS and from other on-line systems to GMT.
- The system's reputation for performance and support.

CIM was founded in December 1968 and specializes in on-line sytems, management information systems, systems design, and programming. The firm also offers Datacom, a data base manager with a GMT interface and logical growth path to IBM's IMS (Report 70E-491-01); Data Entry 1, a means of entering data directly into disk storage from on-line terminals; several financial application systems; and TCOM/7, a software system that allows IBM System/7's to function as front-end communications processors and/or remote concentrators, interfacing with a full range of terminals. Datacom and Data Entry 1 are recent developments, with about three users apiece at this writing.

GMT was first marketed in November 1970. It had been installed at 7 locations within one year and is now serving 17 users, most of them in Texas.

USER REACTION

One user interviewed had carefully weighed GMT versus IBM's CICS/DOS (Report 70E-491-02) for nearly a year before deciding on GMT for his 360/40 DOS system. He rates GMT as efficient and economical of core (52K bytes at first, and now 128K with expanded applications), and says that the vendor has provided outstanding service with numerous efficient enhancements.

A banking user was pleased enough by the work he got out of GMT to purchase the system. His installation, an on-line DOS 360/40 with a DOS 370/145 for backup, has 78 terminals and 8 audio lines, using 150K bytes of core and four task areas. This user must like the vendor, too; he's currently installing CIM's Data Entry 1. □

- Task areas are multi-use; they can process any combination of user programs, including concurrent processing of the same program.

GMT/DOS and GMT/OS are functionally very similar, and recompilation is the only step needed in an operating system upgrade. The functional layout involves three main logical blocks: Task Control, Terminal Control, and File Control. In DOS, these all reside within one partition with the user task areas, with Task Control handling task areas and subroutine areas. In OS, the three modules reside in one partition or region separate from the user task areas.

Task Control provides the facility for concurrent transaction processing, using first-in, first-out scheduling based on resources available. In OS, the interpartition storage protection and interpartition communications facilities are used to create a multitasking environment. These are the steps taken by Task Control: (1) originate the user program (task) as requested from a terminal; (2) queue core resources from tasks that can't be immediately serviced; (3) schedule task areas; (4) service requests for file and terminal I/O and other user-program requests; (5) delay processing of a task until its file I/O is completed; (6) process tasks on the basis of the type of request and order in which I/O events are completed, using a first-in, first-out queue; (7) provide program check and loop control (timeout) facilities; (8) provide facilities for cancelling a task; and (9) terminate completed tasks. In addition, GMT/OS allows addition to or deletion from GMT of a region or partition to allocate resources dynamically in accordance with variations of the load on the system.

Terminal Control uses IBM's BTAM or PIOCS to handle all teleprocessing read and write requests from the user. It handles message translation, terminal scheduling, polling, and paging.

File Control reads and writes all user data sets, gathers statistical data, and schedules requests for exclusive control of records being processed. It prevents simultaneous updates of direct-access records, while supporting exclusive control at the record level rather than the file level. ISAM functions supported include sequential retrieval of blocked or unblocked records, random retrieval of blocked or unblocked records, new key addition of blocked or unblocked records, and random or sequential reading or writing of the files.

PERFORMANCE: GMT has earned a reputation among its users for efficient, bug-free performance. Without exception, the interviewed users stated that the system has performed for them as advertised.

HARDWARE/SOFTWARE REQUIREMENTS: IBM System/360 or 370 under any standard operating system. GMT storage requirements and configurations are entirely dependent upon the application.

PRICING: GMT/DOS leases for $600 per month and can be purchased for $22,000. GMT/OS leases for $800 per month and can be purchased for $28,000. Prices include two weeks of installation assistance, conversion assistance, and training. Maintenance is included with rental of the system and with the first year's use of purchased systems. Thereafter, it is available either on a time-and-materials basis or for an annual fee of $1,100.

INITIAL DELIVERY: November 1970.

CURRENT USERS: 17 as of October 1973. ■

General Ledger II System
Computer Sciences Corporation

MANAGEMENT SUMMARY

GL/II provides a comprehensive (but fixed) selection of company reports ranging from balance sheets through comparative income and expense statements to department reports. Suitable for use by service bureaus or large companies, GL/II runs on a System/360 or 370 under DOS or OS.

Accounting theory and practice evolved with two goals in mind. One is the presentation of information about a company's business in such a way that an accurate picture of costs and profits can be obtained. (Attitudes about what "accurate" means may vary, depending on whether the manager of operations or an investor is reading the books.) The second goal is an arrangement that allows multiple accountants to work effectively when the scope of the accounting task grows beyond the capabilities of one man to make all the entries.

The General Ledger II System addresses itself primarily, but not entirely, to the second goal–posting entries. It is designed for use by a service organization handling many clients, with an administrator serving as a buffer between the user's accounting department and the EDP center. The package could also be effectively used by a large company with many independent plants, subsidiaries, or other cost centers.

In addition to account information, budget and payroll data can be maintained on the master file. Helpful features contained in GL/II include automatic asset depreciation posting, automatic accrual account posting, and single entry for offsetting credit and debit entries.

Many different reports can be generated at the client's request, including a comparative balance sheet, a comparative income and expense statement, and an income and expense trend analysis, in addition to the normal balance sheet, income and expense statement, and transaction register. Budget variance, aged accounts receivables, and payroll schedules are also included.

Each client can choose the number of accounting periods per year and the frequency of updating the master file. All reports are optional except the Transaction Posting Register.

The chief advantage of GL/II is that minimum changes need be made in the customer's accounting procedures. Three levels of accounts allow subsidiary and controlling account relationships to be maintained. Posting of entries to a keypunch form instead of to the books should not be a major hurdle for the accountants. However, the chart of accounts may need to be altered. Information in the reports is presented by account number; to get the desired order for the balance sheet or income statement, the client's chart of accounts will probably need to be resequenced.

Conversion to GL/II is a lengthy procedure, but well organized. Particular care has been taken to ensure a smooth cutover. ▷

CHARACTERISTICS

SUPPLIER: Computer Sciences Corporation, Marketing Division, 650 North Sepulveda Blvd., El Segundo, Calif. 90245. Telephone (213) 678-0311.

BASIC FUNCTION: To automate the posting of journal entries to the general ledger and the preparation of balance and income/expense statements. Ancillary information stored in the master file and many optional reports provide for analysis of the company's operations.

OPERATION: The GL/II System is designed with the idea that there will be an administrator acting as the liaison between the user (client) and the EDP center; i.e., for a service bureau operation.

The master file, kept on magnetic tape, holds all company and transaction information. In addition to the normal journal entry information, accumulated in up to three levels of ledger accounts, budget and payroll data can be maintained to add a dimension to the reporting capabilities. Asset depreciation can be automatically calculated and posted each accounting period. Automatic entries can be made to accrual accounts.

All records are kept on a single master file; separation is obtained through a four-character alphanumeric client number. Each six-digit ledger account number can be extended by a one- or two-element department number, which allows costs to be collected under one or two cost centers for added flexibility of reporting.

Up to 26 consolidated companies and their subsidiaries can be accommodated. Each subsidiary company is assigned a client number beginning with one of the 26 alphabetics; all subsidiaries of the same consolidated company use the same letter, and the accounts for the consolidated company are automatically posted based on transactions in the subsidiary companies. Processing schedules, charts of accounts, and department numbering should be the same for the consolidated company and all subsidiaries to minimize rounding errors and to make the reports meaningful.

Entry to the system is through transaction data submitted to the service center on preprinted forms. These entries are keypunched and the master file updated. Correcting, accrual, deferral, and reversing entries are made as journal entries posted to the ledger accounts, just in a manual system.

Special forms are used to record payroll information, to make adjustments to file information other than account balances, and to identify the reports and other processing considerations for each processing cycle. Additional forms ►

General Ledger II System
Computer Sciences Corporation

▷ GL/II users contacted by Datapro report that the system is working fine and is a good moneymaker when used in a service bureau environment. No problems were indicated at any of the installations interviewed by Datapro. □

➤ are used to conveniently record information for entering new clients, new accounts (three levels), and budget changes. One special form simplifies the entry of offsetting debit and credit amounts by identifying both accounts in the same entry.

An Edit and Validation Register is printed for each client for each update cycle. If no errors are found in the input entries, the transactions are posted. Format errors, such as an alphabetic entry in a numeric field or an out-of-balance condition between debit and credit totals, cause the update processing for that client to be aborted, with errors identified on the Register. The client can specify the allowable out-of-balance tolerance. All unreconciled balances are posted to a single suspense account.

A Transaction Posting Register is automatically generated each time a run is made. This report lists all transactions for the just-completed period; sufficient information is requested in the original source document for journal entries to make this Register adequate to serve as the original book of entry. The source documents input to the service center can serve temporarily until the Transaction Posting Register is produced.

The three levels of ledger accounts are called General, Subsidiary, and Voucher/Invoice. An example of these three would be Accounts Receivable (General), a particular company (Subsidiary), and a particular invoice for that company (Voucher/Invoice). All lower-level entries include the account numbers of the corresponding higher-level accounts, so that posting to all levels can be made from one entry.

MASTER FILE: The master file for each client can be sizable. Two header records are included for each client. A separate record is maintained for each General Ledger account (236 bytes), each Subsidiary Ledger account (140 bytes), each Voucher/Invoice account (76 bytes), each General Ledger account for which budget information is input (92 bytes), each asset that is to be automatically depreciated (140 bytes), and each employee for which payroll information is accumulated (204 bytes).

The General Ledger account record includes balances for each period of the current fiscal year as well as for each period of the previous year. Historical information is automatically revised during end-of-year processing.

The file is maintained in account-number order. Additional dummy accounts are maintained (without balances) to provide titles and subtitles for reports.

OUTPUT: The many available reports can be conveniently grouped into the Transaction Posting Register (original journal), Balance Sheet, Income Statement, nine types of analysis reports, and three types of department reports. All reports except the Transaction Register are optional and are produced only at the Client's request.

The Balance Sheet and Income Statement can be prepared on 8.5-by-11-inch paper if desired. Otherwise, no special forms are required; all necessary titles are printed. The Income Statement includes columns for the current period and year-to-date amounts, with amounts expressed in both dollars and cents and percent of sales or revenues.

In the group of nine analysis reports, trial balances can be prepared for General Ledger accounts and for Subsidiary and Voucher/Invoice accounts. The Subsidiary Trial Balance shows detail transactions and serves as the principal ledger reference source. A Comparative Balance Sheet can be prepared showing previous-year data, differentials, and percentages. Similarly, a Comparative Income Statement includes prior-year period and year-to-date information with differentials and percentages.

An Income and Expense Trends Report shows data for the current period and year-to-date for current and prior years along with the previous period of the current year; differential percentages are included. The Income and Expense Budget Variance Report compares actual amounts with budgeted amounts for the current period, previous period, and year-to-date. Again, differential percentages are included.

The three remaining analysis reports are an Aged Accounts Receivable Schedule, Asset Depreciation Schedule, and a Schedule of Employee Wage and Tax Information. The Aged Accounts Receivable information is derived from the date input with journal entries. (Date of entry is thus missing for these entries, but this can be picked up from the Transaction Posting Register.) In addition, there is an option to request aged accounts receivable statements on preprinted forms.

For those clients whose business is organized along multiple cost centers (e.g., a retail chain with several stores and several departments within each store), the department reports add a much-needed bit of clarification to the company reports. The three types of department reports, Income Statement, Income and Expense Budget Analysis, and Income and Expense Trend Analysis, correspond to the same reports for the whole company except in the order of presentation of items. These reports are organized using the department number portion of the account number as the controlling field.

In the most complex form, department numbers consist of two elements. For example, assume the use of office as the first element and cost center as the second element. The four variations for each department are then a report by office, by cost center within office, by cost center, and by office within cost center. The need for such specialized reports is clearly evident when viewing the standard company-level reports, which list similar accounts for each office together in one group, making it difficult to judge the performance of any one or group of offices or cost centers.

HARDWARE/SOFTWARE REQUIREMENTS: The package will run on an IBM System/360 or 370 configuration, operating in a 56K partition under DOS or in a 90K partition or region of an OS configuration. The minimum recommended configuration includes two disk drives and three tape drives; adjustments can be made.

PRICING: The license fee for the complete package is $20,000. The fee includes a source deck, object deck, utility routines for file maintenance, complete documentation, a 180-day warranty, and a 1-week training program. Optionally, a user can receive any updates for about $1,000 per year. CSC has various audio-visual materials available to assist service organizations in training customers.

INITIAL DELIVERY: Summer 1967.

CURRENT USERS: More than 30. ■

Personnel Management Information System
Computer Sciences Corporation

MANAGEMENT SUMMARY

Once the pay and deduction registers have been printed after the payroll run, the personnel department seldom benefits from the shiny computer. Some hardy souls do apply a report generator to the payroll file to generate pay rate analyses, but this usually benefits the accounting department more than the personnel department. Frequently heard are statements like "Someday we've got to get organized and do a skills inventory so that we can put the right people on the right jobs and quit going outside for extra-cost help and stop ending up with 27 engineers who are whizzes at designing gears but not one who has ever seen a bearing." But such plans seldom get beyond the questionnaire stage.

Computer Sciences' Personnel Management Information System is one way of improving the situation. At the very least, the ordeal of setting up the master file practically guarantees the availability of complete, comprehensive information on each employee plus a relatively painless way to update it and obtain fresh copies for the files. In addition to this basic function, PMI includes a flexible reporting capability for extracting information about selected employees.

Over 20 standard reports can be generated. A special language allows relational tests and logical combinations (but no arithmetic computation) to determine which employees will be included in the report. The reports form a good selection of listings and analyses.

For those one-shot quickie reports, any six fields can be combined in a special report, with the same facility for selection and sequencing.

Notification of employees becoming eligible for benefits, options, insurance, etc. is handled automatically. Once the system is installed and running, such eligibilities are checked each time the system is run. Office or home address mailing labels can also be generated.

Some of the standard report formats are too extensive to fit into a single line on the printer, so they are doubled back into two lines. This leads to two lines of heads at the top of the page. Staggering the heads helps some, but several of the reports will take a little getting used to.

Users contacted by Datapro report that PMI is a very effective system and has a wide range of flexibility. There is, however, one serious consideration concerning the usability of PMI. Ostensibly for use by non-programmers, the User's Manual and standard forms do a good job of easing the problem of the need for exactness in the specifications. But there are over 130 data names and 22 types of standard reports (identified on the input forms by number only) which the user must be familiar with to fully use the data stored in the master file. Some sort of "crib sheet" is definitely called for. The tables in the User's Manual are adequate, but they consist of several pages of detailed data. ▷

This CSC package provides flexibility in storing information about a company's employees and generating a variety of reports. Written in COBOL, it runs on an IBM System/360 or 370 under DOS or OS.

CHARACTERISTICS

SUPPLIER: Computer Sciences Corporation, Marketing Division, 650 N. Sepulveda Blvd., El Segundo, Calif. 90245. Telephone (213) 678-0311.

BASIC FUNCTION: To store information about employees and generate reports concerning various groupings. The system provides a highly flexible reporting method, is written in COBOL, and runs on an IBM System/360 or 370 under either DOS or OS.

OPERATION: Comprehensive data about each employee is maintained in a master file, usually on magnetic tape. The user can select from 28 standard reports or create his own. Each standard report is a particular selection of information to be output for selected employees: the special reports allow selection of both the information to be presented and the employees to be included in the report. Selection of employees is made through a special Personnel Report Request Language which permits relational tests (e.g., equal to, greater than, etc.) and logical connectors (AND and OR) to be used.

In addition, data about authorized job positions can be maintained on a separate master file, usually on disc. The primary use for this master file is tto generate a report listing employees by authorized job.

INPUT: Standard reports are requested by filling out forms corresponding to four punched cards. The specific type of report is identified by a two-digit number only.

Cards 1 and 2 detail the criteria for selecting employees to be represented in the report. Card 3 identifies the sequence in which the information will be grouped by identifying up to four data-field names such as LAST-NAME, EMP-NO, and HIRE-DATE. The criteria for sequencing need not be the same data elements as those reported. Each data field named in the sequence criteria can represent a control break with totals if appropriate and/or the beginning of a new page. Printing of the sequence criteria can be forced or inhibited. Card 4 provides several useful options, including provision for writing the report on tape in addition to printing, for inputting a three-digit security code to release salary information, and for printing mailing labels for employees in the report. In addition, a PID (Personnel Information Document) containing all data in the Personnel Master File for an employee can be requested for each employee in the report.

Two additional cards are needed to create a one-time report. Cards 5 and 6 identify up to six data fields for reporting. Cards 1 through 4 provide the same flexibility for selection and sequencing criteria and options as for the standard reports.

Creation of the master files and maintenance are handled by utility routines. The majority of input and changes can be accomplished by marked-up copies of the PID, which ►

Personnel Management Information System
Computer Sciences Corporation

▷ PMI doesn't really do anything for you that couldn't be accomplished with a good report generator and file maintenance package—except save you the time and trouble of setting up information and reporting requirements. The dearth of in-house programs for personnel management seems to indicate that this is a formidible task indeed. □

➤ is normally printed on a preprinted form clearly identifying all necessary card formats. Information to be input is coded where possible to obtain economies in space and processing.

MASTER FILES: Comprehensive information is stored for every employee. A total of over 130 items are represented for each employee, amounting to about 560 bytes. The employee's name and address are stored along with other personal information and current job and salary information. Items such as minority group, hire source, citizenship, and federal job category are included, as are all prior job positions, salary adjustments, and performance ratings. Space is also provided for vacation and other time off, and for up to eight benefits. Provision is included to add other information to the master file if desired. (Skills inventory is one addition frequently made.) CSC includes a standard set of codes for most coded items, but these can be changed if necessary.

PERSONNEL REPORT REQUEST LANGUAGE: This simple formulation of rules adds great flexibility to PMI, but it has only one purpose: selection of employees to appear in a particular report. The three principal aspects of the language are the data names, relational tests, and connectors.

Data names are fixed and must be used exactly. Some flexibility is provided by a hierarchical arrangement for some groups of data items. For example, LOCATOR refers to a data item composed of division, department, and location; each of the sub-items could be referred to by name if only that piece of data was required for reference (DIV, DEPT, and LOCATION). All dates can be referred to this way.

Relational tests compare the value of a data field against a literal value in the report request. Tests are provided for equal to, greater than, less than, greater than or equal to, less than or equal to, not greater than, not less than, and not equal to.

Connectors allow compounding relational tests. Two are provided: AND and OR. Parentheses can be used to ease the writing of complex relationships. A compound OR series can be simplified by the use of commas.

One word, SKILL, is reserved to simplify coding selection of employees with particular skills. (Skills inventory is not a standard part of PMI, but can be added.)

OUTPUT: The output of PMI can be grouped as follows: Personnel Information Documents (PIDs), standard reports, special reports, system-triggered reports, and auxiliary output.

The PID is really the basic document for PMI. It shows on a preprinted form all information on the master file for an employee. A corrected copy of this document frequently serves as input for updating the master file. It can be produced by itself for selected employees or in conjunction with any other report.

The 28 standard reports fall into 5 groups: personnel, compensation, benefits, manpower, and miscellaneous.

Eleven different personnel reports can be requested, including new hires, separations (short), terminations (long), anniversaries, complete current employee status, name and address listing, telephone directory, job history, salary history, and performance rating history. Also included in this group is a turnover analysis report listing a count of the employees terminated for each reason along with average service and salary; total counts are supplied for each specified grouping, such as department.

Six compensation reports can be requested, including listings of pay rates by employee or by job class and a listing by employee showing past salary performance to be used in planning future compensation adjustments. Three analysis reports complete this group: pay rate distributions by job class and other criteria, compensation versus experience, and salary increases.

Five manpower planning reports can be requested, including manpower classification, seniority listing, manpower age by department, equal employment opportunities, and a position authorization control report.

Two benefits listings show employee participation for a particular benefit and all benefits applicable to the selected employees.

Up to nine different special reports can be requested during one run. Each report can show up to six data fields for selected employees and can be organized according to criteria entered in the report request cards. If desired, detail printing can be suppressed and only summary totals shown. The last two fields can be totalled.

System-triggered reports are set up when the system is installed or by special changes. The criteria are checked each time the system is run and all employees satisfying the criteria are listed. Normally these reports identify employees becoming eligible for benefits, stock options, etc.

Auxiliary output provisions include writing the reports on magnetic tape and preparation of mailing labels, either company or home address, for employees listed in a report.

Only the PID and the mailing labels require special forms. Others are printed with titles.

The total number of employees in the master file, the number active, and the number shown in the report are printed at the bottom of each standard and one-time report.

HARDWARE/SOFTWARE REQUIREMENTS: PMI can be run on an IBM System/360 or 370 under DOS or OS in a 56K partition or region. The recommended minimum configuration is two disk drives and three magnetic tape units, but adjustments can be made.

PRICING: The $22,500 one-time license fee includes a source deck, object deck, utility routines for file maintenance, complete documentation, a 180-day warranty, a 1-week training program, and continual support at the local office level. Optionally, the user can subscribe to an updating service for about $1,000 per year.

INITIAL DELIVERY: Early 1969.

CURRENT USERS: More than 20. ■

Payroll System
Computer Sciences Corporation

MANAGEMENT SUMMARY

The Computer Sciences Payroll System is a straightforward implementation of the payroll function, oriented toward a service organization but also usable by large companies. It was one of the first proprietary payroll systems marketed and is among those with the largest number of users.

The CSC system is adaptable to virtually any type of payroll, although a bit of manipulation will be required in some cases. Basically set up to handle salaried and hourly wages, the Payroll System can accommodate piecework because of the generous size of the hours worked and rate fields. A no-cost optional module permits handling a minimum guaranteed pay situation.

The system features extensive special pay and deduction capabilities. There are a total of 25 allowable special pay categories, each with separate tax handling. Up to 16 of these categories can be defined by individual companies. Up to 33 different deductions can be identified, 24 standard and 9 individualized. Up to 8 or 16 deductions can be specified for each employee. One-time deductions can be specified on the time-card.

The operation of the system is built around a document called the work card. This preprinted document, output from a previous cycle, has a multitude of uses ranging from timecard to specifying salary changes, as well as recording time worked and labor distribution. Input records are keypunched from the work card. An alternate method of payroll information recording, a worksheet, is available but is less flexible.

The basic output is a standard and adequate complement of validation listings, pay and deduction registers, and tax ledgers. Many of these reports use preprinted forms. While no provisions are included for selecting the data to be presented in these reports, the order of presentation (sequencing) and totaling facilities are quite good.

The optional Labor Distribution module provides some selectivity as to items to be included and allows custom labeling of the data columns. The sequencing of the items in the reports is quite flexible, and parts of the labor distribution number can be used as controls.

All in all, the CSC Payroll System is a very reasonable implementation of the payroll function. Just the information required to prepare payroll and related reports is included in the master file—no more and no less. The package was originally designed for banking service organizations, and has been successful in this area.

If you are looking for a way of building an integrated data base for employee information which also generates paychecks, you won't find it here. But if you just want to get paychecks generated with minimum fr·lls, the CSC Payroll System is worthy of consideration. ▷

CSC's Payroll System, also known as Pay3, provides extensive flexibility for pay, tax, and deduction calculations and limited but adequate reporting facilities. Written in COBOL, it runs on an IBM System/360 or 370 under DOS or OS.

CHARACTERISTICS

SUPPLIER: Computer Sciences Corporation, Marketing Division, 650 N. Sepulveda Blvd., El Segundo, California 90245. Telephone (213) 678-0311.

BASIC FUNCTION: To generate paychecks and accumulate totals for reporting purposes. An optional module furnishes the capability for generating labor distribution reports.

OPERATION: The Payroll System is designed to accommodate multiple companies within one master file and is primarily intended for a service organization. The system can be used by a large company, but there are no provisions for consolidating reports.

A punched-card-size document called the work card is the principal form for transmitting payroll data to the data processing center. The data on this card is keypunched into a series of cards for each employee. Several other documents supplement the work card input for specifying processing options, batch totals, and adjustments.

Salaried or hourly employees can be paid automatically with no work card input required. One-time changes or additional pays or deductions can be input through work cards if desired. Overtime pay can be figured based on a rate or factor.

A total of 9 standard special pay classifications are included in the system; each company can identify up to 16 more. Each special pay carries separate tax handling for all taxes. Any employee can receive any of the up to 25 special pays by appropriate work card entries. In general, a special pay can be a fixed amount, a rate per unit (hour), or a percentage of standard pay. Negative pay amounts (fixed or hours) can be entered to handle docking of an employee's pay.

A total of 24 standard deductions are identified in the system. Each company can specify an additional 9 deductions. Up to 8 (DOS) or 16 (OS) deductions can be specified for each employee in his master-file records. (A special additional module provides 16 deductions for DOS installations.) The deductions can be specified as a fixed amount, a rate per hour, or a percent of the standard wages. In addition, up to eight one-time deductions can be specified on the work card. Several options are available for frequency of each deductions, such as each pay period, a specific pay period, or first and third pay periods. Deductions are inhibited in a fixed priority if the pay is insufficient to cover all deductions.

All taxes for all 50 states are provided. As new taxes come into effect, all customers automatically receive all updates. Keeping up with all tax changes is facilitated through CSC's tie-in with Commerce Clearing House, a publisher of legal, tax, and other business reference services. Federal and state income taxes are calculated in one of three ways for each employee: based on exemptions, percentage of gross pay, or fixed amount. Tax exemptions can be specified for each employee for each tax. In addition, up to four different local taxes can be accumulated for each employee to handle multiple work locations. ➤

Payroll System
Computer Sciences Corporation

▷ Users contacted by Datapro report that while the CSC Payroll System is basically good and quite reliable, the system is starting to show its age. Although the technical support provided by CSC is considered excellent, including a timely tax rate update service, the system is not easily modifiable by the user. Over a period of nearly six years since its development, the system has become somewhat awkward to use, especially in view of changing tax structures and associated payroll standards. Most of the current installations are several years old, and the CSC system was generally acquired on a one-time payment basis when there were very few proprietary packages to choose from. Prospective users today should perform a careful cost/performance comparison before selecting any payroll system. □

➤ With the oprional Labor Distribution module, current period, period-to-date, and year-to-date amounts can be accumulated for each assigned labor distribution number, which can be up to 16 digits long. A separate master file is created for labor distribution data.

INPUT: Several documents are used to collect master-file information in an orderly fashion. After file creation, these documents can be used to specify changes in master-file fields not accessible through the work cards.

For production pay runs, two primary documents are used: the work card (or worksheet) and the Company Transmittal Form. Additional forms are used for adjustment entries.

One of the outputs of the system can be work cards for each employee printed with identifying employee information, standard labor distribution number, standard salary or standard hours and rate, plus standard overtime rate. Alternatively, a worksheet can be prepared which lists employees together and provides more space for showing specific pay items and deductions, but includes no provisions for identifying labor distribution. The work card can serve as a timecard, whereas the work sheet would not normally be filled in by the individual employees.

The work card provides a great deal of flexibility in specifying an employee's pay for a particular period, as well as identifying one-time or permanent changes.

There is no limit to the total number of regular pay items, special pay items, special deductions, or change items per employee per period that can be specified on the work card. (Additional cards can be used if required.) An item can be identified as pay only with no labor distribution, or as labor distribution only with no pay.

Multiple checks can be prepared for one employee for a single pay period. There are no control checks in the system except accumulating pay, deduction, and adjustment totals and comparing them to input totals. (A check digit is computed and appended to each social security number.)

The Company Payroll Transmittal Form identifies the current pay period and such items as period ending date and check date. Totals for hours, salaries/rates, and overtime rates/factors, which serve as control totals, are included. A limited number of options can be specified, such as 941A Tax Report only (no paychecks) and an employee address listing.

OUTPUT: Most of the output reports fall into four groups: basic, advice of payment, deduction registers, and workman's compensation. Optionally, labor distribution reports can also be prepared. Each of these groups of reports can be sequenced in a variety of ways; all reports in the same group are sequenced the same. In general, three sort keys are permitted. Each sort key can be the department, status (active, on leave, or terminated), social security number, employee number, employee name, distribution code, or state. Each of these breaks can result in column totals, with or without starting a new page if desired.

The basic reports include Payroll Register, Adjustment Register, Tax Ledger, Employee Listing, and Turnaround Document (work card). The advice of payment report group includes paychecks, deposit advices, and supplemental statements. A separate deduction register is output for each deduction identified in the company's master file information. The basic report package includes a minimal facility for reporting workman's compensation; to get reports that satisfy state requirements, an extra-cost module is required. Extensive use is made of preprinted forms for the output reports.

Up to eight different labor distribution reports can be defined. The principal options available are the item selection, sequence of report, control breaks, report title, and column headings.

Item selection is based on the labor distribution number. Any group of up to five digits within this number can serve as the selection field. The operators are exclude or include, the relationships are equal to, greater than, or less than. The specified segment of the labor distribution number is compared to a literal.

The sequence of the report is based on up to six fields, which can be the department number, employee number, social security number, or any contiguous portion of the labor distribution number. Overlapping of fields within the labor distribution number is permitted. Totals can be taken at each control break, with single, double, or triple spacing or a skip to a new page specified for each control field.

Other outputs can include several types of tape files. One is for federal taxes. Another is a listing of checks for check reconciliation. Still another is a list of pays for automatic deposit through a demand deposit program.

HARDWARE/SOFTWARE REQUIREMENTS: The CSC Payroll System will run on an IBM System/360 or 370, Model 30 or larger, under DOS (64 bytes) or OS (90K partition). The full configuration utilizes five files including working areas. The normal configuration is two disc drives and three tape drives (primarily for sorting), but other arrangements can be made.

PRICING: The $10,000 one-time license fee includes a source deck, object deck, complete documentation, a 90-day warranty, a one-week training program, and the Labor Distribution module. The Workman's Compensation Reporting module costs an extra $1,000. The Minimum Wage and DOS 16-Deduction modules are no-cost options.

Maintenance beyond the 90-day warranty, including system and tax updates as well as fixes for bugs, is available for about $75 per month.

INITIAL DELIVERY: 1966

CURRENT USERS: More than 70. ■

SCERT
Comress, Inc.

MANAGEMENT SUMMARY

SCERT is a computer system simulation program that aids trained analysts in predicting the performance of batch, real-time, multiprogramming, multiprocessing, virtual storage, and time-sharing processing tasks based on highly detailed system definitions and a proprietary factor library. Its use can encompass most EDP systems.

SCERT (Systems and Computers Evaluation and Review Technique), as originally delivered and used in 1962, was designed to simulate the running of sequential batch processing applications and to provide information for deciding which computer configuration represented the best buy for a projected workload. Since then, SCERT has progressed and become more sophisticated by adding capabilities to predict the performance of multiprogramming, real-time (communications), multiprocessing, virtual storage and time-sharing systems.

In a nutshell, SCERT takes highly detailed configuration and application programming specifications and breaks them down into individual processing events. Then, from its factor library, timing and capability information is taken for equipment components and for standard software such as operating systems and/or input/output control routines.

For sequential batch processing, all events are stacked, taking into account only parallel I/O operations within the same program.

The real-time analysis takes queuing into account and calculates probable response times.

The multiprogramming simulation steps the system through the workload in job-by-job fashion, calculating delays in the queue and turnaround time from point of entry to completion. Availability of main storage and peripherals, operating system limitations, and assigned priorities are used to determine when a job can start.

Multiprocessing is simulated by distributing the processing workload between two processors. Systems containing like or unlike processors with a well-defined processing division (e.g., communications front end) are handled on a routine basis. Systems containing unlike processors with dynamic processing assignments require special handling.

Time-sharing is simulated on a discrete basis; SCERT can get down within individual time slices.

SCERT output is a series of reports containing detailed and summary information about the simulated system. Each simulation run provides results for one system design running on one configuration. To build up comparative statistics, a simulation must be performed for each system-design/configuration possibility. (The system design is the collection of application programs that perform the required workload.) Even then, answers are not immediately apparent. A trained analyst is required to determine what approach is best. ▷

CHARACTERISTICS

SUPPLIER: Comress, Inc., 2 Research Court, Rockville, Maryland 20850. Telephone (301) 948-8000.

BASIC FUNCTION: To simulate the actual running of sequential batch, multiprogrammed, real-time, multiprocessing, or time-sharing computer programs for purposes of configuration evaluation or program review. Timing, cost, and other parameters are output in the form of reports; comparative evaluation is then performed by a trained analyst.

OPERATION: The SCERT simulation program is based upon a series of algorithms for converting equipment, standard software (operating systems, compilers, sorts, utilities, etc.), file, and procedure specifications into timing relationships. Basic timing information about hardware and standard software items are drawn from a huge factor library. The factor library also includes such items as equipment costs and environmental conditions required.

The user prepares for a SCERT run by first defining the configuration of the computer system to be simulated; model numbers of each component, feature numbers of options, and program numbers of the manufacturer's software packages to be used are required. For real-time systems, the number and type of communications terminals are identified. Next, the file definitions are prepared, including file structure with average and maximum record sizes, size of files, and frequency of usage. Finally, the actual processing is specified by means of a set of verbs that specify particular processing operations.

Operations can be specified at several levels ranging from gross macros, such as Sort, Merge, and Update, to individual operations such as Move, Add, and Multiply. Separate sets of verbs are oriented toward commercial file processing and toward scientific computations. The wide range of processing definitions permits simulation of programs that have been designed but not written, and permits existing programs to be more precisely modeled. For real-time systems, procedures are specified in terms of events related to five queue points: processor, main memory, peripherals, terminals, and communications lines.

Once the equipment and application have been specified (modeled) and keypunched, the SCERT simulation run is performed and the output reports obtained. An analyst takes these reports and applies his judgment, experience, and cleverness to arrive at conclusions that will form the basis for decisions by the user. ➤

SCERT
Comress, Inc.

▷ Among the projections that SCERT provides are the running times of programs, the utilization of each component, breakdowns of program time (showing where the time is spent among arithmetic calculations, data handling, and input/output control), cost of components and configurations (showing the breakeven point between leasing and purchasing equipment), programming time and cost, response times for real-time systems, effective utilization for multiprogramming systems, and excellent summary reports showing the system design.

SCERT does provide some answers directly, such as optimum block lengths, memory allocations, and peripheral channel assignments.

Certain factors of importance in the evaluation of computer systems are not, and probably cannot be, provided by SCERT. Among these factors are the accuracy of vendors' delivery forecasts, reliability of equipment, competence of the vendors' maintenance personnel, and speed of the vendors in reacting to system problems. Also, SCERT cannot forecast new developments in computer hardware and software, so you'll have to continue to take your chances there, just as before.

The accuracy of SCERT output is a direct function of the amount of information known at simulation time and the care taken in defining the programs to be run. Files for operational applications must be defined in detail, including volume, average and maximum record sizes, number and type of fields, and frequency of use. Programs are defined in what looks like a high-level programming language through the use of verbs representing processing functions such as Extract, Update, Sort, Move, Add, Compare, etc. Floating-point arithmetic (including most FORTRAN functions), matrix manipulation, sort/merges, and elemental processing functions, can be identified.

Sufficient flexibility is provided to model an application that has only been outlined in order to provide a forecast of throughput performance. Analysis of the output and interactive simulations will provide guidelines toward the development of an optimal system.

The application to be run (system design) is defined independently of the configuration on which it is to be run. This simplifies the task of modeling several system designs on several configurations.

However, many "what if?" questions are cumbersome to handle. For example, determining the individual and cumulative effects of using a faster card reader, printer, magnetic tape units, and random-access units could require separate runs for each combination of units (16 combinations in this case). An analyst could really be worth his paycheck here by identifying critical components (from the data provided by SCERT) and thus ▷

► The basic approach taken by SCERT is to identify processing elements, to generate a time for each element, and to identify all parallel paths, such as processing-I/O overlap. These elements combine to form the simulation. The difficulties of simulation increase as the type of processing increases in complexity; when modeling multiprogramming, real-time, random-access, and time-sharing processing, the number of events and possible paths increase enormously. To accommodate these complex possibilities certain algorithms have been built into SCERT for arranging the sequence of events. The reports generated for the more complex processing techniques effectively form snapshots of the system taken at various points in time. Results are usually presented as ranges, such as average and maximum queue lengths for real-time systems.

INPUT: The method used to set up a simulation is akin to a high-level programming language. There are four divisions: Data Definition, Procedures, Configuration, and Environment.

Data definition requires identification of each input and output file to be used. Files are identified as input, output, work, or master. The peripheral device is identified (punched card, magnetic tape, or random-access device). The number of alphanumeric fields, number of numeric fields, and average and maximum size of the records must be specified, as well as the total number of records in each file.

The Procedures division requires the complete definition of the processing to be done in each program or real-time event. There are four general types of operations, as identified by SCERT: sort/merge, mathematical, matrix, and functional. For sort and merge operations, multiple keys can be specified. Mathematical operations identified are similar to the basic FORTRAN functions. Basic matrix operations such as add, invert, transpose, and multiply can be identified. The functional items include a wide range of basic processing functions such as compare, move, add, edit, multiply, and table lookup. For each operation, the number of fields involved, average size (functional operations), and number of times executed are specified. High-level macros, such as Update, Extract, etc., can be used in place of detailed functional specifications. The specific file or I/O device (printer, communications terminal, etc.) to which each operation is related is also identified. The result is a complete picture of the processing activity of an installation.

The Configuration division identifies the exact equipment complement to be modeled. Components are identified by model number along with any optional features. The standard software to be used, such as operating system and sort programs, are identified. This information allows entry into the SCERT factor library for timing, cost, and other information about the system to be modeled.

The Environment division is used to define the staff that will support the configuration and applications already defined. Entries such as experience level, type of experience, and salaries are required here. This information is used principally to estimate the programming effort to implement defined applications.

Preprinted forms are provided for easier and more uniform entry of required data and parameters.

The approach taken by the designers of SCERT results in independent definitions of the application (Data Defini- ►

reducing the number of non-profitable combinations examined. This points out a significant characteristic of SCERT: in general, it evaluates your design; it doesn't make recommendations.

The magnitude of the output produced by SCERT can be formidable. Most of the reports are relatively short, seldom exceeding a few pages even for large systems, but the detailed analysis outputs a separate page for each program or real-time event; a hundred-page report is not unusual when simulating whole systems. And it is to this report that the analyst must go for a real understanding of what the system is doing.

The possible profitable uses for SCERT are many; convenient groupings include system design, equipment selection (including upgrading), program review, software selection, and workload scheduling.

USER REACTION

The majority of early users used SCERT for equipment selection in competitive bidding and proposal review situations. Now, however, a growing number of users are using SCERT to stimulate existing programs and thereby discover inefficiently coded sections.

SCERT has caused some controversy among computer professionals. In its early days, many analysts took the viewpoint that it couldn't be done. (It's rumored that some of the same analysts also said man wouldn't reach the moon.) Of course computer systems can be simulated. The real question is how well.

The U.S. Government took a very favorable position and required in many competitive bidding situations that the proposal be evaluated via some type of simulation method. For a number of years, that meant SCERT. Consequently, most of the major computer manufacturers are continuing users of SCERT – although it is less frequently used outside the federal procurement area, primarily due to a lack of trained analysts. However, many commercial users of computers have used SCERT with favorable results, citing savings in equipment procurement and program run time attributable to factors learned from SCERT.

The obvious approach for a computer user to take when first using SCERT is to model his existing applications and equipment configuration to test both the simulation program and the analyst's understanding of SCERT and the user's installation. Comress states that the accuracy of simulation is typically within about 5 percent overall for a specified workload, with individual runs being up to 10 percent off. Verifying this is very difficult.

Batch systems, even multiprogramming ones, are usually heavily input/output oriented for typical commercial

tion and Procedure divisions), the hardware and software used (Configuration division), and the staff to operate the system (Environment division). This adds flexibility in the use of SCERT and allows various elements of the simulation to be conveniently varied to accomplish different aims. For example, different processing approaches can be examined on the same configuration with a view toward optimizing the operation of an installation. Alternatively, the same processing workload can be modeled on different configurations in an equipment procurement task.

OUTPUT: Four series of reports comprise the visible output of SCERT. Three of these correspond to the three types of modeling that SCERT performs: sequential, random/real-time, and multiprogramming/processing. The other series is a group of summary reports.

The reports on sequential processing total four. The first shows the program operational time for running and for set-up and the memory used for each run, along with the frequency of each run as specified in the input. The second report distributes the runs into daily, weekly, and monthly cycles and shows the running and set-up time for each cycle along with totals for average and peak months. The third report is optional and breaks down the time for each run into various central processor and peripheral times. The fourth report is actually a series of reports, one for each batch run or real-time event; a complete description of each event is listed, including the affected files, file blocking details, I/O channel assignments, and peripheral timing details. This report is useful for non-sequential processing because it provides the basic information used in the other SCERT simulations.

Four real-time projections are produced. The first report displays each real-time event and projects the running time ignoring queuing delays and multiprogramming considerations. The second report incorporates a statistical probability analysis and shows the utilization of various hardware components in the system; the hardware is divided into central processor, data channels, random-access devices, communications terminals, and communications paths. The expected and worst-case queue lengths are shown for each set of components. The third report shows response times for each real-time event. Program active time (the time from the end of the input message to the beginning of the output message) and total response (the time from the beginning of the input message to the end of the output message) are shown in a probabilistic manner; the expected response times are shown for 99, 95, and 50 percentile groupings to accommodate various design levels. The fourth report shows the memory requirements for each event, including any background processing.

The multiprogramming series of reports is based on time zones, an arbitrary period defined in the input. The first is a history presentation showing the status of the system at each change (i.e., each time a program is started or stopped). The limitations of the system are displayed at each point, along with the memory being used and available peripherals. The limitations accounted for include shortage of memory or peripherals, empty queue, and operating system, such as a limitation on the number of programs that can be active at any one time. The sequence of scheduling is based on priorities assigned in the input, by the operating system scheduling algorithm, and by the time the analyst assigns as to when the run is available to the system. The second report identifies each

processing. Here knowledge of I/O times and system limitations for parallel operations is the key, and one would expect Comress to be able to come up with accurate information. For real-time jobs involving random events, evaluation is far more difficult. The results presented by SCERT show average and worst-case situations (as is proper); checking an operating communications network to derive comparative results is chancy at best. Suffice is to say that many users appear to be satisfied with the results, and that it beats manual analysis hands down.

To sum up, SCERT has proved beneficial to many customers. The relatively high cost of the package means that it will be hard to justify for a small installation. Probably the chief use for a small user would be equipment procurement anyway, so maybe a one-time special study could be justified for that purpose. (Comress notes that one-time studies for small users have by no means been limited to the equipment procurement function.) For the large user who is willing to put the time and effort into proper usage, SCERT is a useful addition to the analyst's bag of tools. □

run and the turnaround time associated with it; this time is composed of two parts, time on computer and time in queue. The time in queue is affected by the time originally assigned by the analyst as to when the program is available to the system. The last report in this series shows the utilization of the processor and memory for each time zone.

The five summary reports include an optional report on system requirements, which summarizes the complete specifications of the applications to be simulated, including file and processing definitions. Another summary shows the complete listing of hardware components, including special features, along with purchase prices, monthly rentals, and environmental requirements such as cooling and power. A third summary shows the time utilization for each major component, along with a projection comparing rental with purchasing plans and showing the time to break even. A fourth report shows an estimate of the programming time for each program. The last of the summary reports shows an estimate of the programming time and cost for each application grouping.

All in all, a total of 26 different reports can be generated in a SCERT simulation. All simulations, whether they are for sequential, random/real-time, or multiprogramming, include the detailed report on each run and real-time event. This report is typically very voluminous, since a page is generated for each run or event. The various summary reports and the several reporting stages are helpful in getting an overall picture of what is going on.

FACTOR LIBRARY: The key element of the SCERT package is the factor library, for here reside the timing parameters for hardware components and software elements. Up to about 350 factors are recorded for a piece of hardware, and the record for a particular component can be as long as 1200 characters.

Essentially all computer systems of commercial interest are presently included in the library. A new tape is distributed to each continuing customer each month. Included in each new release are data about newly announced systems and updated information about previously included systems. Comress's target is to provide customers with data within about 30 days after announcement by the computer manufacturer. Initially, Comress is dependent on information released by the manufacturer, which can be somewhat sketchy at the time of announcement, particularly in the area of software. Timings are estimated initially, based on similarities to previously covered systems. As more information becomes available, the factor library is modified.

The factors in the library for central processors are the result of algorithms specially designed for each system. For example, IBM's announcement of the System/370 meant new logic had to be developed to handle the buffer memory, which can drastically affect central processor timing.

Proprietary software packages such as Autoflow, Mark IV, and Score are not a standard part of the library, but they can be included; sometimes these items are offered at extra cost. Likewise, computer systems not in the standard library can be added at extra cost.

PERFORMANCE: Run times for SCERT are difficult to present in meaningful fashion due to the large variation. In general, computer time for simulating batch processing runs will be quite small in relation to the time required to set up the definitions for input. Real-time (communications network) simulations and multiprogramming simulations can take an appreciable length of time if the number of events or programs is large (say on the order of 500 or more). A non-standard modification is available that makes use of additional core storage (above the required 110K bytes) to reduce run time.

HARDWARE/SOFTWARE REQUIREMENTS: SCERT is currently designed to run on an IBM System/360 or 370 with 110K bytes of core storage under DOS or OS or their virtual storage counterparts. It will also run on an equivalent UNIVAC Series 70 (formerly RCA Spectra 70) under TDOS, as well as on an NCR Century 200 and on a UNIVAC 1100 Series computer under EXEC 8. Standard interface programs are used so that all systems can use the same monthly factory library release.

PRICING: SCERT is available two ways. A user can license the system or he can arrange to have special studies made. In either case, the user can elect to run SCERT on his own system or have Comress do the running.

SCERT license agreements range from one to five years at a cost of from $24,000 to $50,000, there is a monthly maintenance and updating charge of $1,000. Special studies start at about $5,000 and can go way up from there. Included with a license arrangement is training for the user's analysts; currently, the training program lasts two weeks.

INITIAL DELIVERY: 1962 for batch processing simulation, running on an RCA 301. A version running on the IBM 1410 became available in 1964, and a version for the UNIVAC 1050 somewhat later. The IBM System/360 version was delivered in late 1968, and the UNIVAC 1108 version in late 1969. Real-time and multiprogramming simulation were added in 1963 and 1965, respectively, and have since been refined. Time-sharing was added late in 1969, and virtual storage capability in October 1972.

CURRENT USERS: There were about 200 continuing users as of October 1973. Over 400 users have had special studies made. ■

AMIGOS
Comress, Inc.

MANAGEMENT SUMMARY

AMIGOS is "plug-compatible" software. It replaces IBM's standard ISAM (Indexed Sequential Access Method). Actually, AMIGOS ties into the IBM operating systems through the standard provision for user-coded I/O routines, making it a logical replacement rather than a physical replacement. In fact, ISAM and AMIGOS can coexist – an advantage when programs and files are being converted.

AMIGOS offers substantial advantages in terms of the amount of main memory used, speed of accessing data, and efficiency of disk utilization. Versions are available for all of the popular System/360 and 370 operating systems.

The advantages of AMIGOS can be summarized as follows:

- OS users get substantial savings in core and disk utilization, and often large performance gains.
- DOS users benefit primarily in performance gains and disk utilization. (Some smaller DOS installations of AMIGOS may show little gain in any area except file reorganization.)
- Users having applications with a large volume of record additions, the requirement for both basic (random) and queued (sequential) accesses, and the need to perform a large volume of random accesses will benefit the most. (Conversely, applications using disks for sequential update processing, essentially as logical replacements for tape drives, will probably show little gain in performance.)

ISAM has always been heavily promoted by IBM. Yet, some users are selecting another access method, writing their own I/O control routines, or electing not to implement certain applications because of the space and/or time required by ISAM. IBM's own VSAM (Virtual Storage Access Method) promises worthwhile improvements over ISAM to VS users. But AMIGOS can deliver these improvements now, to real and virtual storage users alike.

A couple of results from Comress demonstration runs using most of AMIGOS's features illustrate why the company is being so lyrical in its promises to users. One job that took 28 hours to run using ISAM was cut to 3 hours with AMIGOS. The response time in one inquiry system was cut from 3 minutes to 6 seconds.

USER REACTION

On the strength of its performance, AMIGOS earned a place on the 1973 Datapro Software Honor Roll (see ▷

AMIGOS, a 1973 Datapro Software Honor Roll package, replaces IBM's ISAM and offers better storage utilization and higher performance to many System/360 and 370 users through more efficient coding and indexing techniques. Versions for DOS, OS, and virtual storage users are available.

CHARACTERISTICS

SUPPLIER: Comress, Inc., Two Research Court, Rockville, Maryland 20850. Telephone (301) 948-8000. (Originally developed by Data Art Corporation with partial funding by Comress.)

BASIC FUNCTION: To replace the IBM Indexed Sequential Access Method (ISAM) modules on any System/360 or 370 computer system operating under DOS, OS, or their VS counterparts. AMIGOS features reduced core requirements, higher performance, and better disk space utilization.

OPERATION: AMIGOS fits into the IBM operating system as a user-coded I/O routine; installation requires adding AMIGOS to the systems library, converting all ISAM data access statements to AMIGOS calls (on a one-for-one basis), reassembling or recompiling, reloading the disk files, and producing new JCL cards.

AMIGOS can accommodate fixed-length or variable-length records under DOS, OS, and their virtual storage counterparts, and also under VM/370. Handling records of undefined length is not planned for AMIGOS at the present time. (Availability will, of course, depend on demand.)

Conversion of BAL programs from ISAM usage to AMIGOS usage is a manual procedure. Comress offers a conversion program to automatically convert COBOL programs. The conversion involves only replacing each BAL or COBOL call to ISAM with an equivalent call to AMIGOS; all replacements are one for one.

Evaluation of AMIGOS should be considered in three parts: effect on main memory, effect on disk utilization, and effect on performance.

EFFECT ON MAIN MEMORY: AMIGOS requires 11,000 bytes of main storage to hold the resident control coding. This coding includes provisions for random accessing (equivalent to BISAM) and for sequential accessing (equivalent to QISAM). IBM OS ISAM, on the other hand, requires from 3,500 to 14,000 bytes for the basic method (BISAM) and from 6,300 to 8,000 bytes for the Queued method (QISAM). Typical usage would be about 11,000 bytes for BISAM and 7,000 for QISAM. Many inquiry-type applications require both BISAM and QISAM portions.

An even larger chunk of main memory can be taken up in OS ISAM by the channel programs and control blocks required for each active file. Approximately 1,500 bytes are required for each file for each of the two methods, BISAM and QISAM. AMIGOS on the other hand, requires only 600 bytes per file, and that includes both access methods.

DOS ISAM requires substantially less main storage space than OS ISAM; in most DOS installations, AMIGOS will save little memory for control coding and control blocks.

AMIGOS generates a master index each time a file is opened and stores it in main memory; precise size will vary with the size of the file, of course, but in general only 40 bytes are required for each 2316 Disk Pack.

►

▷ Report 70E-010-40). Additional users contacted for this report also had praise for the reliability and file integrity presented by AMIGOS. These qualities greatly ease restarting a teleprocessing system after a "system crash," because there is no tendency to have lost records.

One user, with 2.5 million records on-line to CRT displays, rates AMIGOS as "the best software package I've ever bought." He says it has yielded about a 37.5 percent reduction in display terminal response time. Users seem to agree that AMIGOS typically occupies about 10K to 20K bytes of storage.

Any user of ISAM files on a System/360 or 370 should seriously consider AMIGOS. □

➤ DISK UTILIZATION: The utilization of disk storage space is affected by the elimination of the master index on disk, immediate elimination of deleted records, format of overflow area, and handling of file reorganization due to overflow.

The savings due to not holding the master index on disk will be a few tracks at best.

In ISAM, when a record is deleted, it is flagged, but the space becomes available only when the file is reorganized. In AMIGOS, a record is physically deleted at execution time by the simple expedient of eliminating it from the block. If a record is added that falls within a block that has an empty space due to a deletion, then the record is written directly in that block, not in the overflow area.

AMIGOS blocks records in the overflow area; ISAM records them individually. Blocking allows much greater storage density in the overflow area and reduces the frequency with which reorganization of the file is required.

File reorganization is handled dynamically on an individual cylinder basis. As the overflow area for one cylinder becomes filled, that cylinder (prime area and overflow area) is split between two cylinders. Thus, the overflow space used is proportional to the activity of the various portions of the file.

Of course, all good things come to an end, and eventually AMIGOS reaches the point where no more file space is available from the initial assignment. The whole file is then reorganized just as in ISAM; i.e., a sequential tape is written and the file is reloaded.

Another important point is that the user is notified when the total space is becoming inadequate, based on a user-input percentage so that reorganization of the file can be planned for a slow period. ISAM merely aborts the program when the out-of-space condition is reached.

PERFORMANCE: The most spectacular results from replacing ISAM with AMIGOS are in the area of performance. Contributing to the increased performance are holding the master index in main memory, skip recording, dynamic reorganization, and block addressing.

ISAM requires three disk accesses to obtain a record randomly from the prime data area (master, cylinder, and track indexes), while AMIGOS requires only two (cylinder and block indexes). If a record is located in the independent overflow area, ISAM may require many more disk accesses to get it into core. (Later releases of DOS permit the master index to be held in main memory.)

AMIGOS does not record blocks in sequential order on the tracks. Instead, they are recorded in every other position, with the second half of the data being put into the spaces thus created. (This is called interleaving with a factor of two.) The purpose of this is to extend the amount of processing time available between sequential accesses to the data. This is of no advantage in a purely random-access application or in an application which involves updating the records (because you must wait a full revolution anyway so that data can be rewritten).

Comress states that about six to nine seconds are required to reorganize a cylinder. Reorganizing a complete file under ISAM varies with the size of the file, but you can count on it taking several minutes minimum.

Block addressing is used in place of ISAM's track addressing in the final index stage. Essentially, this means that a block of data is accessed directly. Under ISAM, reading begins at the beginning of a track only. On the average, about one-half revolution is saved for each random access. Additionally, the user has available true file and program compatibility between the OS and DOS versions of AMIGOS. AMIGOS I/O statements or files designed for a DOS environment will function identically in an OS system (and vice versa) without any modification or reloading.

EXTENDED FEATURES: The following features are either not available or not fully supported under ISAM but are supported under AMIGOS:

1. RPS device support for the IBM 3330 and IBM 2305.
2. OS secondary allocation during AMIGOS file loading.
3. Alternate track allocation for defective tracks.
4. Trace table of at least the last 9 I/O events.
5. Shared buffering for low-activity files.
6. Floating indexes for automatic assignment of the cylinder index to a lightly-used disk drive.
7. Generic search capability to access groups of records whose record keys contain some but not all identical parameters.

HARDWARE/SOFTWARE REQUIREMENTS: AMIGOS will run on any IBM System/360 or 370 computer operating under DOS, OS, DOS/VS, OS/VS1, OS/VS2, or VM/370 (separate versions, of course). See the discussion under Effects On Main Memory for more details on main storage requirements.

Programs must be modified and reassembled or recompiled, and the disk files must be reloaded to use AMIGOS.

PRICING: AMIGOS is available under either a 1-year or a 3-year license. The 1-year license costs $8,000, with monthly use after the first year costing $500 per month. Conversion credit from a 1-year to a 3-year license is provided. Under the 3-year license, the first copy of AMIGOS costs $17,000, including installation by COMRESS and user documentation. Multiple-copy discounts are available. After the 3-year license period, the first copy of AMIGOS costs $2,500 per year for program and documentation maintenance. The variable-length option costs an additional 10% of the basic license or lease fee.

INITIAL DELIVERY: Summer of 1970. AMIGOS is currently running in production under DOS, OS, OS/VS1, OS/VS2, and VM/370. The DOS/VS version is currently in test, with release expected December 1, 1973.

CURRENT USERS: About 90 as of October 1973. ■

PLUS (Program Library Utility System)
Cullinane Corporation

MANAGEMENT SUMMARY

The PLUS system provides both a unified source-program library and batching test procedures and audit-trail documentation.

PLUS is available in two versions: sequential and direct-access. The chief differences are performance at low load levels and price. For large libraries where only a small percentage of the programs are accessed at any one time, the direct-access version can show substantial performance advantages because the sequential version requires a complete pass of the library tape for each run: the direct-access version also costs more.

Working from card decks supplied by the individual programmer, PLUS allows modifications to be made to standing programs or the insertion of completely new programs into an installations's source program library. Still under the control of the programmer, and without the operator being involved, these programs can then be compiled, link-edited, and executed using standard or specialized test data as an individual job or as part of a batched operation.

Use of the batching operations for the library maintenance is designed to cut down on the time involved when each programmer sets up his own particular test procedures. Computer time is also saved through reduction in the number of jobs which are separately run. In an open shop, the direct-access system is sometimes desirable.

The PLUS documentation is designed to provide an installation with both working and audit-trail program documentation of all source programs while minimizing the workload involved. It is also designed to require virtually no intervention by the computer operator.

The direct-access version also includes special data compression features along with reuse of old program areas to conserve disc space.

Maintenance of test decks along with the source programs permits full testing of a program when any modification is made to it, without having to reassemble the original test material or risk an incomplete test being made before the program goes back into production status.

The PLUS program takes care of all procedures through library update and compilation or assembly using one control card. If testing is also to be accomplished, the test programmer inserts all necessary control cards for compilation and for the conduct of the actual test itself.

PLUS works under, and is compatible with, IBM DOS and OS as well as their virtual storage counterparts. Any ▷

PLUS is a utility program that updates and maintains COBOL, FORTRAN, or PL/1 programs and other 80-column data on a magnetic tape or disk library, together with such test data as is useful. Sequential and direct-access versions are available for IBM System/360 or 370 DOS, OS, or VS Systems.

CHARACTERISTICS

SUPPLIER: Cullinane Corporation, One Boston Place, Boston, Mass. 02108. Telephone: (617) 742-8656.

BASIC FUNCTION: The Program Library Utility System (PLUS) provides a source-program and test-data library and documents the various changes that are made from time to time. It also provides for simple program-test procedures by allowing test files and control cards to be included in the files. These are then organized into a job stream under programmer control, with minimal operator action.

A major highlight of the program is that it allows for the holding of programs in any mixture of languages: PL/1, COBOL, FORTRAN, Assembler, etc.

OPERATION: Programmers using PLUS prepare punched cards indicating all changes and modifications to all programs, quoting the program name in the library and the sequence number of the instructions where necessary. The modification decks of a programmer or a group of programmers are batched, and in a single run the library file is updated and listings of complete programs, or simply of the changes which occur, and other reports are produced as requested. During the updating run, a job stream is automatically created using control cards pre-established for various languages or supplied by the various programmers in their modification decks. These compliation control cards may be produced by the PLUS program itself.

PLUS-Sequential is written in COBOL. PLUS-D/A is written in BAL.

INPUT: The input to PLUS is one Action card plus program changes, if any. The single Action card identifies the program, specifies the actions, and indicates any options desired, such as resequencing of statements, making of copies, etc.

For the sequential version, programs on the file are stored in strict order, and the input created by the programmers must be organized in that order before they are batched. Identification of programs on the various listings includes a ten-digit field for a short description—such as "Card To Tp" — as well as the official name and the author. Sometimes the author's name is not used, and instead the space allocated for it is used to amplify the program description.

Included with the input are any control cards, other than those for actual compilations, that the programmer wants to place in the job stream, and any data files that may be needed. This allows for automatically proceeding to compile and test the program in a controlled manner. ➤

PLUS (Program Library Utility System)
Cullinane Corporation

▷ language can be used for the programs. Programs can be printed out, punched out on cards, or placed on tape or disc. No lock-in is therefore involved, and an installation using PLUS would be able to change to another library and documentation system at a later date without restrictions.

In view of its low price and lack of lock-in characteristics, PLUS can easily be justified for any installation that requires a complete audit trail of source programs or an organized and working testing procedure.

USER REACTION

PLUS lags behind The Librarian and Panvalet in number of users (110, 850, and 964, respectively), but one of the Cullinane package's enthusiastic users feels that the large difference is attributable mainly to the lack of an equally strong marketing force for PLUS. In conversations with users, Datapro learned that PLUS is commendable for its performance, support, and low cost. Only one complaint, a minor one, arose in those discussions: one user expressed displeasure with the documentation index. He reported that it was initially difficult to find things in the documents, and that his engineering department still approaches the PLUS manuals with trepidation, and usually just tries to get an answer from the programming staff instead. (The engineering department has its own library of FORTRAN programs under PLUS.)

Otherwise, users cited program security and control, plus the ability to speed program production by using parts of other programs to build new ones. PLUS is also credited with saving errors, eliminating much time wasted in searching documentation, and providing a valuable audit trail.

Some users have modified PLUS, and some break PLUS-Program Manager into its two parts (Sequential and D/A) to serve the respective purposes of: (1) security with control of the source program library with recoverable copies of source code, and (2) as a programmer productivity aid with source copies on disk.

Users stated a liking for the package's simple command structure and its availability of source code, as well as the ability to easily carry extra copies of source code. □

➤ OUTPUT: The output produced by PLUS consists of three major reports; an updated library; copies of specific programs on cards, tape, or printer as requested; a job stream for compilations and program executions; and perhaps a backup tape for security.

The three reports are a Change Report, an updated Library Index, and a Job Scheduler to help the operator handle the workstream that the program has created.

The change Report shows, for each modification, the exact instructions that have been received, including what listings were prepared, whether there was a jobstream created, etc. The program automatically updates the modification number used each time a change is made, and also notes the date of a modification (taken from a leading card or the operating system). These details, as well as copies of the old and new versions of affected individual statements, are printed on the Change Report.

The Library Index lists the names and identifying data for all programs that are in the library. It gives the date of the last revision, the modification number, etc., so as to warn any user about possible untried changes. It is oriented to the data processing manager and normally will be filed with the library documentation.

By comparison, the Change Report is program oriented, and will be used by the programmer and filed with the program documentation. The program modification number provides a check on the completeness of the file for audit purposes, and at suitable times copies of specific modification levels can be included.

The Job Schedule is a listing of the job control cards that are prepared by users or generated by the PLUS program, so that an operator can see what jobs are to be run. As an additional method of assisting the operator, console messages can be included with the control cards supplied by the user.

Where the instructions call for programs to be reproduced on cards, tape, or listed on the printer, these are also part of the output.

The workstream formed by the program, including compilations, assemblies, link-edits, and test runs, is entered automatically after the Job Scheduler is printed. Provided that the user-supplied controls allow, compilations and tests can be handled without operator intervention.

PERFORMANCE: Operating speed of the system for tape libraries is dependent on the length of the library and the speed of the tape units. The program consists of a single pass over the entire library file without a rewind. The direct-access version can be appreciably faster because only those programs required are accessed.

HARDWARE/SOFTWARE REQUIREMENTS: 65K IBM System/360 or 370. The system must be running under IBM DOS, OS, or VS. PLUS-Sequential will also run on UNIVAC Series 70 systems. Users report running the system in 40K to 80K partitions.

PRICING: The sequential version goes for a one-time charge of $1500. The direct-access version sells for $2500 for a two-year arrangement and costs $375 per year thereafter.

SUPPORT: Program support at the time of installation consists of installation and an on-site briefing, together with the provision of a user's manual, the program, and operating instructions. Malfunctions of the program will be corrected without charge indefinitely.

INITIAL DELIVERY: June 1969.

CURRENT USERS: About 110 as of July 1973. ■

IDMS
Cullinane Corporation

MANAGEMENT SUMMARY

Cullinane's IDMS (Integrated Database Management System) is an important new development in the data base software sweepstakes, even though there are currently few outside users of the package (and, at this writing, only two who have acquired the package from Cullinane). IDMS is the only data base system specifically designed to meet the CODASYL Data Base Task Group's language specifications that is available for both IBM System/360 and 370 and UNIVAC Series 70 (ex RCA Spectra) equipment. Moreover, it was purchased from its developer, B.F. Goodrich (the "no blimp" company), for conversion into the DMS/90 system for the new UNIVAC 90/60 and 90/70 computers.

IDMS was developed at B.F. Goodrich in 1970, when that firm decided to convert its hardware from the GE 200 Series to the IBM System/370. Goodrich had been using GE's IDS on a GE-235 and was pleased with the system's facilities. This is understandable, since IDS was favored by the 1969 CODASYL Data Base Task Group as the basis for their specifications, and IDS is still in popular use on the Honeywell Series 400 and 6000 computers. The object of IDMS, at Goodrich, was thus to be the vehicle for data base software conversion from IDS to complement the hardware conversion.

Cullinane purchased IDMS from Goodrich, and is now marketing the system and providing continuing enhancement, development, and maintenance. In charge of the project is Mr. Thomas F. Meurer, a Vice-President of Cullinane who frequently serves as chairman of the ongoing Seminar on Data Base Design of the American Management Association.

Although IDMS and UNIVAC's DMS/90 stem from the same acquired program, they are no longer connected, by any business interests or otherwise. UNIVAC 9000 Series customers, that is, have no connection with Cullinane. UNIVAC Series 70 customers, can, however, obtain IDMS for their systems from Cullinane.

IDMS was originally written in COBOL and a special intermediate systems software language, in about a 60/40% ratio. While some of the Goodrich IDMS customers purchased the source code, Cullinane supplies object code to its customers. The UNIVAC Series 70 DOS version was easily developed because of the similarity to IBM System/360 DOS. IBM System/360 and 370 users should note that IDMS is treated as a problem program and functions exactly the same under any supervisor, real or virtual (paging). It has successfully run under all 360 and 370 operating systems. ▷

IDMS implements a subset of the CODASYL Data Base Task Group language specifications and runs on any IBM System/360 or 370 computer and on most UNIVAC Series 70 computers. Designed as an extension of ANS COBOL, it offers flexible data base organization and processing facilities.

CHARACTERISTICS

SUPPLIER: Cullinane Corporation, One Boston Place, Boston, Massachusetts 02108. Telephone (617) 742-8656. Detailed inquiries should be directed to Mr. Thomas Meurer, Project Leader.

BASIC FUNCTION: Data base management on IBM System/360 and 370 systems, under DOS, OS, and their virtual storage counterparts; and on UNIVAC Series 70 systems, under DOS. IDMS is designed in conformance with the CODASYL Data Base Task Group language specifications report of April 1971. The system includes a data dictionary (DD), a data manipulation language processor (DML), a data base management system (DBMS), and data base administration utility programs (DBA utilities).

Also available are a generalized communications interface and a special IDMS/Culprit combination for Culprit-type retrieval from an IDMS data base. The combination includes Culprit, which is described in Report 70E-491-03. The generalized communications interface (GCI) includes a central access monitor program that threads calls to the DBMS, on-line utilities for such functions as rollbacks, and a DBMS interface program, which handles problem program ABENDS and resides with the communications program.

OPERATION: IDMS is a comprehensive data base management system developed under the guiding principles of the 1971 Report of the CODASYL Data Base Task Group. It is designed to satisfy the need for standardized data management techniques that provide: (1) separation of the data definition and data manipulation functions, (2) an acceptable degree of data independence, and (3) data base protection and integrity. IDMS has three principal components: a Data Description Language, a Data Manipulation Language, and a Data Management Routine.

The *Data Description Language (DDL)* is a stand-alone language whose record descriptions are compatible with those of COBOL. The schema DDL input provided by the data manager completely defines the data base. The data base description, or "schema", is composed of areas, records, and sets. The Schema Compiler validates and stores the Schema DDL information in the Data Dictionary. The Subschema Compiler generates a series of tables (Data Base Descriptor) which are maintained in a catalogued file in mass storage for later interpretation by the Data Management Routine.

The concept of "areas" in DDL provides the means for associating the data base with the physical mass storage devices in which it resides. A "set" is simply a named collection of records. A record can be stored in the data base ➤

IDMS
Cullinane Corporation

According to expert users, IDMS has several valuable strengths and relatively few weaknesses. The strong points are:

- It is a subset of the CODASYL Data Base Task Group language specifications.
- It is designed as an extension to ANS COBOL.
- It can be used with any host language that supports the CALL statement or its equivalent.
- It permits users to specify various data structures including network, hierarchical, "bill of material", or a combination.
- It allows any number of user-specified entry points into the data base.
- It provides control over the physical placement of records so that performance can be optimized.
- It eliminates the need for periodic reorganization of the data base.
- It provides utilities to monitor data base performance and storage density.
- It provides data base rollback/recovery facilities to assure integrity of the data base following either a hardware or software failure.

The major IDMS weakness involves a core storage size versus performance tradeoff, but this is based on the amount of buffer space used, and is therefore relative. Tests of IDMS reveal no significant performance degradation even at 90% utilization levels. IDMS is not re-entrant. However, a user reports that the average IDMS application uses only about 120K bytes. Compare that to re-entrant IMS (the IBM Program Product, see Report 70E-491-01) code for the same application, which requires about 190K bytes.

In addition, Cullinane points to IDMS retrieval capabilities based on that firm's Culprit system (see Report 70E-272-03), a built-in data dictionary, ability to interface with any generalized communications software, and utilities for data base administration as advantages of IDMS.

USER REACTION

Meaningful strong points of the IDMS system, from the point of view of users experienced with data base software, were discussed earlier. All that's left is to mention the few remaining possible trouble spots. Early IDMS users complained about poor documentation—but Cullinane is

through keys (DIRECT), based on values (CALCULATED), or near an owner occurrence (VIA). A given record can be both an "owner record" of, and a "member record" in, one or more sets, and a different ordering procedure can be used in each set.

General placement of a record type in the data base is achieved by specification of an area in which all occurrences of the record type will be stored. Additional storage placement control within an area is achieved by one of the following location mode options, which allow the user to determine the number of entry points into the data base and control access performance:

- The DIRECT option allows maximum user control over logical storage location of a record occurrence within an area. In IDMS, the user can supply a "suggested" data base key before a record is to be stored. If the suggested key is available, it will be assigned to the record occurrence. If unavailable, the next available key will be assigned by the DBMS.
- The CALCULATED (CALC) option stores the record occurrence based on the value of one or more data items within the record. A mathematical transform algorithm uses the item values to produce a logical storage position within the area associated with the record.
- The VIA option specifies that record occurrences will be accessed primarily as members of a named set and stores the records as near as possible to the owner occurrence.

Many options are provided for ordering member record occurrences within a set. The intra-set position a new record occurrence will occupy when stored in the data base can be specified as the first record (top of a push-down list), the last record (bottom of a push-up list), or a position immediately before or after a record occurrence established by the user program. A set can also be ordered by a record type if more than one record type is included in a set as well as by a data base key.

Each record occurrence can be automatically maintained in ascending or descending order based on the value of one or more data items included within the record. The logical ordering of member record occurrences in a set is independent of the physical placement of the records. Furthermore, and most important, the same record occurrence can participate in any number of different sets, each with different ordering criteria.

The type of membership can also be different in each set. Set memberships for a record type can be mandatory or optional depending on whether the user program is permitted to insert, or remove, an occurrence of this record type from the set. Mandatory set membership means that a record occurrence is a permanent member of the set as long as the record is present in the data base. Optional set membership allows a record occurrence to be removed from a set without being deleted from the data base. Set membership can also be specified as an automatic function performed by the DBMS when a record occurrence is stored in the data base, or as a manual function to be performed by the user program.

Members of a set are automatically linked in the forward directions (next pointers). The user can optionally request that the set be linked in the reverse direction (prior

IDMS
Cullinane Corporation

▷ producing new documentation that is said to be very good, and it will be complete by January 1, 1974. Also, Cullinane promises an enhanced, improved IDMS re-release on the first of the year.

A user also sounded a small note of displeasure about the data base administration utilities, which were said to be non-general, and currently unable to operate under all operating systems. Contacted about this, Cullinane reports having just finished solving this problem, which stems from the fact that users who acquired the package from Goodrich also were supplied with Goodrich's uniquely cataloged procedures.

In summary, Datapro feels that IDMS is an interesting data base system that should be borne in mind by IBM and Univac users, especially those interested in a data base system that is basically an extension of COBOL. □

➤ pointers) so the chain can be processed in either direction. The user can also specify that the owner pointers be carried in the member record occurrences so that the owner record occurrence is known and can be accessed without following the chain to its head. Therefore, the options for set linkages are to have next pointers, next and prior pointers, next and owner pointers, or next, prior, and owner pointers.

The application programmer can only access a data base through a "subschema" which has to be provided for him by the data base administration staff. The subschema is made available to a program by an INVOKE declarative statement. Since the programmer cannot touch any data not invoked as a subschema, data privacy and integrity are assured, at least at the application program level.

The imperative data manipulation statements for control, retrieval, and modification, and the INVOKE declarative, are referred to collectively as the *Data Manipulation Language (DML)*. It is the procedural language used by individual programmers to access the data base. It is used by connection with a host language – usually COBOL – which describes the procedures for processing the data once it has been accessed. The functions of DML can be generally described by listing its commands: OPEN, CLOSE, FIND, GET, OBTAIN, MODIFY, STORE, DELETE, INSERT, REMOVE, and IF. The programmer inserts the appropriate DML commands into the syntax of his COBOL source program. A DML Preprocessor validates the syntax and logic of each DML command and then converts the DML commands into a COBOL-compatible format and builds the necessary record descriptions and communication areas in the user's Data Division. The altered syntax is passed on to the COBOL compiler, which produces an executable program. The program can be linked with others to form a run unit. The DML Processor is unique in that it flags many errors in data base processing at compile time, rather than later on at test time.

The *Data Base Management System (DBMS)*, the key operational component of IDMS, maintains the data base and preserves its integrity. No run unit is allowed direct access to the data base; instead, all DML commands are funneled through the Data Management Routine. In addition to its storage and retrieval functions, DMR includes "before" and "after" image logging, rollback, and recovery routines that prevent loss of data through hardware failures, software bugs, or erroneous input.

As in the case with COBOL, IDMS can be used with any host language that supports a CALL macro or its equivalent. Since the DML processor is available only for ANS COBOL, user communication with IDMS from other host languages is through arguments passed via CALL statements.

HARDWARE/SOFTWARE REQUIREMENTS: An IBM System/360 or 370 computer under DOS, DOS/VS, OS/MFT, OS/MVT, OS/VS1, or OS/VS2, or a UNIVAC Series 70 computer under DOS or VMOS. Average overhead is about 50K bytes per application program.

PERFORMANCE: According to users, IDMS perfoms as represented, and conforms well to the CODASYL specifications mentioned earlier. IDMS is not expressly designed for speed in transaction processing. However, it is an excellent data base manager, and can be linked to the Culprit series of output processors.

PRICING: IDMS is available only under license. In addition to the one-time license fee, the user must pay a yearly usage fee of 10% of the license fee. Existing users are protected against any price increase under this scheme. The usage fee entitles the user to all new releases and documentation, technical support, program maintenance, and membership in the IDMS users' group.

The license fee for IDMS is $30,000. This includes object code for the schema and subschema processors, the data dictionary, the DML processor, the DBMS, and DBA utilities. It applies to any number of CPU's at one site, and can include IDMS/Culprit if the user already had Culprit. Multiple sites are discounted to half-price.

IDMS/Culprit, the retrieval package, alone carries a $12,500 license fee. The GCI (Generalized Communications Interface) option to IDMS, which includes the central access monitor program, on-line utilities, and DBMS interface programs, bears a license fee of $7,500.

INITIAL DELIVERY: Cullinane began marketing IDMS in May 1973, but official announcement of IDMS from Cullinane was not until August 15, 1973. A new version will become available on January 1, 1974.

CURRENT USERS: 5 as of November 1973. ■

CULPRIT
Cullinane Corporation

MANAGEMENT SUMMARY

The CULPRIT Output Processor system has been designed primarily to fulfill the reporting needs of non-EDP user departments. It possesses not only the ability to produce complex reports equal to those produced by COBOL, but also the ability to access unusually complex file structures. Thus, the system brings the benefits of a report writer to files that often serve the major applications of an organization and that had previously been considered too complex for retrieval systems.

A few of the files that CULPRIT and its user-department versions, such as EDP-Auditor, can access via sophisticated Cullinane-supplied interfaces are: IDMS, IMS, and TOTAL; UCC TEN Data Dictionary; IBM's CIF; UCC's RDMS; and IBM's ALIS/CFO, BOMP, DBOMP, MRP, and CFMS.

The system can be used effectively with files that are already in existence. Indeed, it is designed to be used by both programmers and nonprogrammers who simply have a file definition in front of them. One-time reports, mailing labels, and other desired products can be quite easily specified in the parameter cards, and CULPRIT can produce the desired results, normally with little programming delay.

But CULPRIT can also play a second, totally different role that should be of considerable interest to management. It can provide a new way of writing programs—one which could be cheaper both in the original writing and in program testing and modification.

Looked at in this way, CULPRIT provides a method of creating the outputs needed from an updated file, whether they are to be printed outputs or other computer files. CULPRIT's output capabilities are quite flexible, so that an applications programmer can consider his main job completed when he has created an up-to-date file. At that point he can hand the operation over to a CULPRIT programmer, who will arrange to produce the necessary outputs by simply preparing the CULPRIT parameter cards.

From a programming point of view, this approach has many attractions. To start with, the outputs created can be very easily changed. In fact, using this method it would be standard operational procedure to produce one set of outputs for program testing and verification purposes and a totally different set of outputs for actual use. Previously, this approach was usually impractical. Checking over printed outputs that have already been structured for specially printed forms has never been easy, and has often been skimped. Again, a number of tests of accuracy can be built right into the test output (but not the final output) in order to automate some part of the process.

After the output has been created, it is still easy to modify for later operations. Predesign of forms has always been a bugbear in the creation of computer programs. Using CULPRIT, a number of different output designs can be produced quickly and economically, even after the program is in operation.

CULPRIT is an output processor system for creating reports from IBM System/360 and 370 and UNIVAC Series 70 files, independently of the original file-creation programs. Swift turn-around time and good totaling and titling features are emphasized. CULPRIT can also serve as the output portion of independently written programs, leading to potential reductions in programming and testing time. Numerous versions are now available.

CHARACTERISTICS

SUPPLIER: Cullinane Corporation, One Boston Place, Boston, Massachusetts 02108. Telephone (617) 742-8656. Also, 3250 West Market Street, Fairlawn, Ohio 44313. Telephone (216) 867-8840.

BASIC FUNCTION: CULPRIT allows access for report-writing purposes to existing files. In addition, it can be interfaced to newly written programs to take the place of the normal output operations. The operation of CULPRIT is controlled by a series of parameter cards, which describe the input and output files.

INPUT: The input to the system consists of a series of cards describing the various items necessary to an output routine. These include:

- Input File Description. A short description giving the file number, the record size and type (fixed or variable), and the number of records to the block. The file description also includes a brief description of the device concerned—simply whether it is a magnetic tape, disk file, etc., and the abort routine to be used if there is a breakdown.
- Record Description. Separate from the file description is a definition of the records involved. The records in the file need not be homogeneous—files having more than one record type present no problems. The record description provides for an eight-character name for each field that is used, together with a definition of its length, starting position, and format.

OUTPUT CONTROL: Output, like input, is broken down into a number of different parts. In the first part, the output device and blocking are specified, together with the details of any sorts needed within the given report. The sorts are defined by simply specifying the name of the field to be sorted on, as given in the input parameters of the program. All sorting is performed in one step, with the sort JCL never varying. The sort can be skipped. In addition to this basic information, some specialized information for printer output is given on the same card. This defines the number of lines per page, the use of totals in the final output, etc.

OUTPUT FORMATS: After the output control information, the next group of parameter cards provides the additional information that is needed to generate a report. Details for the headings are given, first in a Title block and then on cards which provide for the headings for the individual columns of the reports.

Where the data currently in the file is to be used for the body of the report, these items are simply listed on further cards, giving details of the editing required: zero suppression, use of signs for numeric operations, etc. However, in addition to this, a series of procedures can be defined to create new fields in the report which are not directly included in the file data, or for selection and other processing.

USER REACTION

Datapro conducted telephone interviews with 10 users of CULPRIT. CULPRIT versions in use among these users included EDP-Auditor, IMS, DBOMP, TOTAL, and IDMS. They rated the system as follows:

	Excellent	Good	Fair	Poor	WA*
Overall satisfaction	6	4	0	0	3.6
Throughput/efficiency	2	7	1	0	3.1
Ease of installation	6	4	0	0	3.6
Documentation	3	5	2	0	3.1
Vendor support	4	6	0	0	3.4
Training	3	6	1	0	3.2

*Weighted Average on a scale of 4.0 for Excellent.

The 10 users had the following computer configurations: IBM 360/50–1, 370/145–2, 370/155–4, and 370/158–3. The system performed as advertised immediately for seven of the users and eventually for the other three.

One user, a large insurance company, used the EDP-Auditor version of CULPRIT to perform a special study on draft processing. They studied the distribution of drafts (especially those with low dollar amounts) by auditing 11,000 drafts. From this they determined a cheaper method of handling the low-dollar drafts. They estimate that the results of this study saved them more than one million dollars a year.

However, it was CULPRIT's fulfillment of its two basic design objectives–ease of use and interfacing with complex file structures–that most impressed its users. One EDP-Auditor/CULPRIT user found that "With a staff of 8 and over 1,000 operational programs plus audit responsibility in new systems development, to write our own audit programs would be prohibitive." Another user found that educating nonprogrammers in the use of TOTAL/CULPRIT often caused them to go overboard in their report writing without consideration for the impact their runs had on computer production. But this user is so pleased with CULPRIT's ability to interface with TOTAL files that he is now evaluating IMS/CULPRIT.

All the users stressed that CULPRIT is a flexible, user-oriented output processor system, and that Cullinane's support, past and present, assures the system's future development. □

PROCEDURES: The CULPRIT procedures, in the form of three-address instructions, are in many ways the key to the success of the system. The eight-character names used in the previous parameter cards identify the operands and the result positions. The form is the familiar Operand No. 1, Operation, Operand No. 2, and Result. All standard arithmetic and logical operations are allowed. As a result, the procedure cards, when coded, are no more difficult to read than any of the symbolic assembly languages. For example, the coding QUANT X UNPR GRSA, which breaks down into "Quantity X Unit Price = Gross Amount," is fairly understandable to anyone who takes the time to familiarize himself with the system. These procedures can be used for a number of different purposes, including selection of records for specific reports and creation of entries for the final or "total" lines. In addition, a series of procedures can be defined to create new fields in the report which are not directly included in the file data, or for selection and other processing.

BATCHED REPORTS: The CULPRIT system is capable of preparing a number of reports that are required from a single file or multiple files all at the same time, even though the reports are quite distinct. The program allows the batching of the reports, and effectively culls out all the records that are selected through any of the standard methods for any of the reports. Then these are sorted in the appropriate manner and placed on the appropriate output with or without processing and editing, depending upon the specifications.

The basic system, then, is to provide, in a single grouping of parameters, the selection mechanism, the calculation definition, and the wording for headings, totals, etc. This provides an effective report-writing capability.

AS AN OUTPUT SYSTEM: The second level of use of CULPRIT can be more important than the report creation operations that have so far been described. The second level is to provide an output specification system which, working from files that have been updated in the course of a computer run, can create the necessary outputs on the printer and/or other files. CULPRIT's function here is to permit separation of the output portions of programming from the file maintenance and generation portions.

The technique is to use exactly the same grouping of control cards, specifying the form of an output, and to use them to create tape records, disk records, or punched card records as necessary, as well as printer records. The definition of the record contents and format is handled exactly as in the definition of the records for the reports, but is extended to include the other devices and to allow for the case where a particular output file is to be written on more than a single device.

An installation using this concept would instruct the programmers to write those portions of all programs that involve file generation or updating, but to provide definitions only, in the standard CULPRIT format, for the necessary output. Indeed, it might be unnecessary for the problem programmer to do even this–instead, a special programmer could have the job of creating the CULPRIT specifications for the required outputs.

PROCESSING: The parameters supplied to CULPRIT generate core-resident machine coding which directly results in the output of the necessary files, and such subsequent report creation as is necessary, without any special user programming.

PERFORMANCE: CULPRIT uses the standard IBM sorts, and it is generally I/O-limited for other operations. The program is written in System/360/370 Assembly language for optimum performance.

HARDWARE/SOFTWARE REQUIREMENTS: IBM System/360 or 370 or UNIVAC Series 70 with a 52K-byte DOS partition or 80K-byte OS partition or region, and enough tape and/or disk drives to support necessary sorts. The programs operate under IBM OS (MFT, MVT, VS1, and VS2, including HASP and ASP), DOS, and DOS/VS, as well as under UNIVAC Series 70 DOS and TDOS.

PRICING: The CULPRIT system costs $15,000 for a two-year license, plus a 15 percent renewal fee each year thereafter. The renewal fee includes new versions, consulting services, maintenance, updates to manuals, and keeping the users current with any changes to the manufacturer's operating systems or hardware. Special interfaces, such as the ones for IMS, TOTAL, RDMS, etc., cost an additional $5,000 each.

The EDP-Auditor system, which is CULPRIT plus application material and special training specifically for auditors, is available at the same price as CULPRIT. If an installation already has CULPRIT, EDP-Auditor costs only an additional $5,000.

At installation time, at least two days of hands-on training at the user's site is provided by Cullinane Corporation. Subsequently, the program is guaranteed for the life of the agreement against any malfunction.

INITIAL DELIVERY: April 1970.

CURRENT USERS: Approximately 200 systems were installed as of April 1975. ■

JASPER
Datachron Corporation

MANAGEMENT SUMMARY

JASPER is a tool designed to permit smooth transitions from batch processing to multiprogramming in DOS and DOS/VS environments. Certain problems inevitably arise when multiprogramming is brought in – scheduling, cost allocation, and computer utilization reporting, for example. Additional, less obvious areas of concern are the creation of effective job mixes, configuration analysis, performance evaluation, and management control.

JASPER's heart, its Computer Operations Master File (COMF), is a single repository for all information collected by the system, and it is built and maintained automatically. The information contained therein is available for scheduling, cost allocation, and performance evaluation.

Not only does JASPER provide reports that detail computer and operator activity, but the system is noted for the depth of detailed information available from its reports. It would be possible, for instance, to use JASPER to find the right program (e.g., required run time, CPU utilization, peripherals needed, etc.) to fit an available multiprogramming slot.

JASPER provides the $JOBACCT program that accesses the Job Accounting Table available with IBM's DOS Release 25 or later. Although a user could supply his own $JOBACCT program, it is doubtful whether many DOS users possess the sophistication and/or can afford the investment in personnel resources to create a program of anywhere near the comprehensiveness offered in JASPER. Moreover, the vendor, Datachron, not only has the experience gained in creating and maintaining JASPER, but also enjoys a reputation for flexibility and cooperativeness. This can be important to those wishing to learn to interpret JASPER's reports for special purposes, or wanting to tailor new reports. Datachron was founded in 1967.

The lyricist who wrote that the best things in life are free must not have had much knowledge about data processing. "You get what you pay for" is the more appropriate adage to apply to JASPER. Its facilities are very strong, and so numerous that it is difficult to do more than outline them in the Characteristics section of this report. Quality counts too, and JASPER's ease of installation, reliable operation, and simplicity of use are undoubtedly worth the user's dollar.

USER REACTION

JASPER receives very strong user endorsements. A New York City department store uses it to schedule work, and claims that it releases jobstreams from queues. Daily ▷

JASPER, a job accounting system for DOS and DOS/VS IBM System/360 and 370 computers, uses an accounting data base to provide informative daily and periodic reports. It is also quite flexible, and offers POWER, GRASP, ASAP, and Sprint interfaces.

CHARACTERISTICS

SUPPLIER: Datachron Corporation, 174 Fifth Ave., New York, New York 10010. Telephone (212) 675-5333. Sales offices are also located in Ft. Lauderdale, FL, and Milwaukee, WI.

BASIC FUNCTION: Provides job accounting, scheduling, charge-back, and computer resource analysis in IBM System/360 and 370 DOS and DOS/VS systems.

OPERATION: JASPER is a $JOBACCT program, written to interface Job Control in a DOS Release 25 or later supervisor, and a series of applications programs to provide a hierarchical reporting system. It is written in BAL and ANS COBOL, uses PIOCS (Physical I/O Control System) to interface and access the data in IBM's Job Accounting Table, and stores it on disk for later processing by JASPER routines. This data capture function of JASPER is not core-resident; it exists as a logical transient and is called in between programs.

Daily job execution statistics are used by JASPER to create and update the Computer Operations Master File (COMF), which builds a table of scheduling information that includes peripherals required, average run time, core size, percent compute-bound, I/O factors, and rerun and incomplete allowances of programs. Charge codes, frequency information, and other static information required by COMF can be extracted from the //JOB cards if currently available; or, instead of reformatting JCL, this information can be entered directly into COMF, which can also accommodate other information that wouldn't fit on the cards.

Control fields in COMF can be customized, increasing JASPER's flexibility. It can, for example, combine individual jobs or steps for reporting purposes without changing the //JOB cards. It can provide year-to-date averages for some jobs, use only the last period's statistics for others, and terminate updating of averages at any desired point. It can ignore job run statistics when a job is run on an abnormal machine type (backup, etc.) so as not to introduce statistical distortions. It can generate job surcharges for such items as multiple-part forms, and it can also generate inventory requests for special supplies.

Each job run is checked against its data base record to see whether all required steps are completed; then, run time and CPU time are compared to prior averages to determine whether the job has been degraded by multiprogramming mix or operational considerations. The user can specify a deviation from the average that will cause deviant runs to be flagged and reported. Other exceptions reported are terminations (by code), reruns, jobs requiring link-edit or emulation, jobs over/under time limits, no matches in master file, jobs not completed, and others requested by the user. ➤

JASPER
Datachron Corporation

reports are said to be clear and graphically explicit, letting a manager see immediately any gaps in system utilization. The user expects more assistance from the package soon, in the form of a pre-scheduler that he expects Datachron to release for JASPER.

An industrial user of the full JASPER system and the POWER interface has both DOS and DOS/VS experience. He especially likes the two system and device utilization graphs, reports that his operations manager makes good use of the job characteristics reports, and says that his programming manager uses the programmer reports to advantage. This user is presently building up the COMF to begin charge-back reporting. The reporting is very flexible; nearly any facility can be charged for in various ways. This user also reports that the system was extremely easy to install and that he was able to use vendor-supplied JCL for all standard reports. In use, the operator just selects one of three decks: daily, weekly, or monthly, as appropriate.

Datapro also interviewed users with the GRASP interface. These users prefer JASPER to GRASP's Job Accounting, even though one mentioned that JASPER and GRASP became somewhat incompatible when GRASP's Partition Balancing is employed. This, however, was a first-time problem, caused by the need to use GRASP's Job Accounting File instead of IBM's. The problem has been permanently fixed now, and the reports are the same, even though the data sources are different. Users also feel that JASPER generates equitable charge-backs.

JASPER users seem to have only praise for the vendor. All say that the technical support is expert and timely, and that Datachron cooperates with users to a very high degree. This remains true for this small company despite distance considerations.

In summary, JASPER is a system well worth consideration from DOS and DOS/VS multiprogramming users, especially larger users. □

A very clear system utilization graph shows device and partition utilization in one-tenth hour (6 minute) increments. The report includes all devices by address, their usage by partition (multiple partition use within a period is indicated), and partition usage of CPU time.

A daily summary report presents numbers of runs, run times, percent of CPU use, CPU time, overhead time, and more, by categories such as normally and abnormally terminated jobs, total throughput, complete and incomplete production totals, tests, reruns, outside work, partition assignments, charge code, and other factors selectable by the user.

A daily log report goes into even more detail, showing numbers of devices used, start and stop times, types of devices used, and so on. It can be an invaluable tool, when properly analyzed, to optimize multiprogramming throughput.

For scheduling, JASPER presents comprehensive job name directory and scheduling index reports. The former is an alphanumeric catalog of jobs, their requirements, and their operating characteristics. The latter provides a cross-reference between operating requirements and job identities, and can be produced in as many sort sequences as required for effective relationship determination.

In addition, JASPER presents a device utilization graph, an operator activity report, and a cost allocation report. It can also aid in operations and security by requesting and validating system and operator identities, job names, reasons for job failures and IBM cancel codes, and valid IPL date and time.

For DOS/VS users, JASPER can compute the paging degradation for every user program. It can benchmark real versus virtual performance.

HARDWARE/SOFTWARE REQUIREMENTS: An IBM System/360 or 370 operating under DOS, Release 25 or later, or DOS/VS with the IBM Job Accounting supervisor option macro.

PRICING: JASPER is available either under perpetual license for a one-time fee or on a monthly rental basis. It is supplied with documentation and operating instructions. Maintenance, which includes compatibility with all DOS releases and program updates, is free under rental and for the first year under license; thereafter, it costs $200 per year. Discounts are available for multiple-CPU installations. Spooler interfaces cost $250, a one-time fee.

	License Fee	Rental
Basic System ($JOBACCT, COPY CONVERT, DAILYLOG, and daily Summary Report on shift and daily totals)	$1,595 (DOS) 1,995 (VS)	$ 85 (DOS) 105 (VS)
Daily Exception Report/ Operator Activity	250	15
Daily System Utilization Graph	250 (DOS) 350 (VS)	15 (DOS) 20 (VS)
Cost Allocation (summary & detail)	400	22
Job Name (summary & detail)	550	30
Job Characteristics	350	20
Device Graph/Period Operator Activity	250 (DOS) 275 (VS)	15 (DOS) 18 (VS)
Period Summary Report	200	12
Computer Operations Master File (Edit, Update, Print, and Match)	750	40
Entire JASPER System (all above)	4,000 (DOS) 4,500 (VS)	200 (DOS) 225 (VS)

INITIAL DELIVERY: January 1972.

CURRENT USERS: About 40 as of December 1973. ■

Data-Man
Data-Man, Ltd.

MANAGEMENT SUMMARY

Data-Man is a general-purpose software system that provides file management and report retrieval facilities. It is a Canadian product that is currently in use in 29 leading Canadian installations, on systems ranging from a 64K-byte 360/30 to a 2M-byte 360/85 and a 4M-byte 370/168. It is currently enjoying use in individual installations as well as in large data centers, and is also installed in England and the United States.

Report requests in Data-Man are extremely free in form, and the average request can be coded in half a day by typical non-EDP-oriented users after only two days' training. With requests thus generated, the average number of tests to a successful run is reputed to average about 2.3. Importantly, input can come from multiple files.

The package is in continuous development. The current release, Version 5, was delivered in March 1975 and offers the ability to read and write up to 32 sequential or indexed sequential files at any point in a Data-Man procedure, as well as a user exit facility that allows a call-return from Data-Man to any user-written subroutine and automatic return to the next Data-Man statement. Data-Man has other strengths. Its users rate it as one of the most cost-effective systems of its type, and it has won out over strong competitors in quite a few situations. It is said by many to be the least expensive package of its type. Moreover, it can be rented as well as purchased, and a free trial is available.

The four basic parts of Data-Man are File Definition, Retrieval-Reporting, Table Creation/Maintenance, and File Maintenance.

File description is a necessary part of any software of Data-Man's type, but this one has a few new wrinkles. To save time, only those fields required for specific reports need be defined to the Data-Man system. Based on this input, data can subsequently be extracted conditionally.

To retrieve and report, Data-Man reads the input file or files, creates an internal select output file, sorts that intermediate file by report number and sequence within report, and then transfers control to a report generator that prints the requested reports. All this is done automatically.

Data-Man provides access to tables of numeric factors or alphabetic descriptions designed by the user through specification statements. Table data is entered via cards, and the resulting internal table file can be used automatically in table lookup procedures during reporting and/or maintenance.

The file maintenance facility within Data-Man is used in much the same way as the reporting facility. It reads a master file and a transaction file to create a new master

Data-Man is a powerful, flexible file management and report retrieval package that offers ease of use and multiple-file input to IBM System/360 and 370 users without regard to operating system considerations.

CHARACTERISTICS

SUPPLIER: Data-Man Ltd., 1160 IBM Building, Calgary, Alberta, Canada. Telephone (403) 266-6358. Sales and support are provided in Eastern Canada, Vancouver, and London, England. Contact supplier for further information. The system is also available in the United States through the supplier.

BASIC FUNCTION: File management and report retrieval for IBM System/360 and 370 configurations under any operating system. Data-Man can be used by non-programmers for the production of requested reports and by others for file management that includes creation and/or updating of files.

OPERATION: There are three steps in preparing a report from a data base using Data-Man: (1) define the data base, (2) code the report specifications, and (3) execute the reports.

In defining the data base, specifications provide the name assigned to the data base, physical characteristics of the files within the data base, names assigned to data elements within the files, provision of the attributes (field length, etc.) of the data elements, and file fields that are to be used by the system to coordinate reading of multiple records and input files.

Coding the report specification involves providing the name of the data base to be used, the conditions for selecting the report data, the report sort sequence, heading information, and the print format for the report's detail and total lines. The coding form is quite simple, having only line numbers, a directive column, a name column, and a parameter column.

For processing a report request, specifications give the data base to be accessed for execution of the report, the identification of the report(s) that are to be executed, and run-time information such as the date and other parameters.

The Data-Man system consists of several processing components that, in general, can be run independently or in conjunction with other components. The translator converts the user's source-language statements into a form that can be used by other phases of the system. Its output is a printed source listing with error messages and an internal program library or control file that contains the compiled version of the source-language procedures used by all other components.

The table creation component of Data-Man reads the user's table data cards and creates or updates the table library. The file tables are maintained in proper sequence, and can be used by all other system components.

The retrieval component is responsible for retrieving data from the user's data base that is required for report file and an audit file. The facilities offered range from simple maintenance using a few statements through complicated maintenance and updates.

Data-Man provides all these features: (1) ability to process existing files; (2) complete conditional logic, e.g., if, else, go to, etc.; (3) coordinated reading of up to three files for a report; (4) ability to read and/or write up to 32 subfiles; (5) complete calculation facilities, including Boolean commands; (6) creation of any number of reports in one pass of the files; (7) overlay structures, work fields, and conditional field definition; (8) table lookup facilities for single- or multi-column tables; (9) user exit facilities; and (10) reports with automatic formatting, headings, subheadings, details, and totals, with any number of lines and unrestricted calculations, conditional logic, and editing.

USER REACTION

Datapro interviewed four users of Data-Man. They rated the system as follows:

	Excellent	Good	Fair	Poor	WA*
Overall satisfaction	3	1	0	0	3.8
Throughput/efficiency	2	2	0	0	3.5
Ease of installation	4	0	0	0	4.0
Documentation	1	2	1	0	3.0
Vendor support	3	1	0	0	3.8
Training	2	0	0	0	4.0

*Weighted Average on a scale of 4.0 for Excellent.

The users represented the following computer configurations: IBM 360/40–1, 370/135–2, 370/155–1. Their operating systems were IBM DOS–3, and OS/MFT–1.

The system performed as advertised immediately for two of the users; the other two were not present when the system was installed. All the users interviewed commented on the system's powerful retrieval capabilities. By using the system's data glossary, one user found he could define "virtual" data items. For example, if he wanted a "net" field (and none had been defined), the system would calculate net as a percentage of "gross" (which resided in the files as a defined physical item) and make it available to him without changing the physical item "gross."

The system's formatting capabilities were described as "dynamic." That is, if in reading a record there is a change in the control file, there is a provision to automatically change the report heading. With such features, another user felt Data-Man was best suited to those installations doing development work as opposed to ongoing systems maintenance. He felt that by using Data-Man in this manner he could cut his COBOL programming time by one-fifth.

Each of the users interviewed had evaluated Data-Man against a minimum of three comparable systems. One conducted a literature search of 25 of these systems, narrowed the field down to 6 reasonable choices, and benchmarked 3 of them before selecting Data-Man. Its ability to win out in such rigorous evaluations should make Data-Man a worthy candidate for consideration by nearly anyone considering a data management system. □

printing. Up to three coordinated files can be used as input, and any number of reports can be produced in one pass of the input file(s). The process is invoked via control cards. Report items are selected, the table library is used to bring in the required tables, and the program library is used by this phase.

The sort component reads the output file of the retrieval component and arranges output into the demanded sequence before passing control to the report component. The report component handles the report generation, using report specifications from the program library and tables from the table library. It ensures that printing continues until all reports are completed.

Additionally, external subroutines written in any language can be called.

PERFORMANCE: Data-Man is said by its users to perform as advertised, especially in report creation based on data retrieval. This is its most popular usage, but the system also wins endorsements for its other functions. Please refer to the "User Reaction" section of this report. About 80 percent of the Data-Man jobs in operation are routine repetitive tasks.

HARDWARE/SOFTWARE REQUIREMENTS: IBM System/360 or 370 under DOS, OS/MFT, OS/MVT, or their virtual storage counterparts. The system is release-independent to these operating systems and is transparent to OS and DOS. Minimum core requirements are 34K bytes for DOS and 44K bytes for OS.

PRICING: Monthly rental is $450 per month plus an installation fee of $400. The system can be purchased outright for $11,000 to $15,000. Use of Data-Man can also be obtained through a data center for a surcharge rate; contact the vendor for further information.

INITIAL DELIVERY: The initial version was delivered late in 1969. Enhancements have since arrived about twice yearly, and the current Version 5 was first installed in March 1975.

CURRENT USERS: As of March 1975, there were 35 copies of Data-Man in use at 32 locations: 29 copies in Canada, 5 in England, and 1 in the United States. ■

DSI COBOL 20
Decision Systems, Inc.

MANAGEMENT SUMMARY

DSI COBOL 20 is a COBOL compiler for the IBM System/360 Model 20. As such, it is potentially useful to a large number of Model 20 users who in the past have had to rely on RPG, BAL, or (lately) PL/I to prepare their programs. While there may be an argument as to whether RPG or COBOL is better for specific applications, there can be little question that COBOL is the more flexible of the two languages–and that it is, in any case, better to have two business-oriented languages available than to have only one.

The ease of reading COBOL programs is the major reason for the language's great popularity. COBOL's English form enables programmers to discern a program's approach and purpose with greater ease. This "self-documenting" feature facilitates program maintenance.

DSI COBOL 20 provides a subset of standard COBOL tailored to the limited hardware facilities of the Model 20 computer. The language is similar to the Honeywell COBOL B language which was introduced for the small Series 200 computers in 1966. Experience with COBOL B has shown that such COBOL subsets can be used quickly and effectively by people with little or no computer background. Because of the similarity of the languages, DSI COBOL 20 should also be quite satisfactory with regard to ease of learning and use.

Comparison with the PL/I language is more difficult. PL/I does offer advantages in coding complex calculations, as in scientific computation. But the Model 20 is not well suited for scientific processing, so that advantage is minimal. The prospective user will have to weigh the increased procedural strengths and elegance of language expression of PL/I against the legibility and familiarity (to most programmers) of COBOL.

It should also be mentioned that the DSI COBOL 20 documentation, in our estimation, is the simplest and clearest set of user documentation for a programming language yet produced. Separate short documents provide a reference manual, a quick-reference summary, and a complete program listing. The documentation of DSI COBOL 20 should make this package much easier to use than the standard RPG language.

The majority of experienced commercial programmers are now familiar with COBOL, and many prefer it to other programming languages. Accordingly, the availability of DSI COBOL 20 may make it easier to attract and retain programmers, as well as reduce programmer training requirements, if programmer turnover has been a problem. ▷

DSI COBOL 20 is a COBOL compiler for tape- or disk-oriented IBM 360/20 computers. It is fully integrated with existing Model 20 software and is upward compatible to other IBM System/360 models. DSI COBOL 20 is currently the only way to use COBOL on a Model 20.

CHARACTERISTICS

SUPPLIER: Decision Systems, Inc., 200 Route 17, Mahwah, New Jersey 07430. Telephone (201) 529-1440.

BASIC FUNCTION: DSI COBOL 20 is a COBOL compiler which operates on, and produces code for, an IBM 360/20. Its function is to allow a 360/20 programmer to use COBOL instead of other languages such as RPG or BAL.

INPUT: The input language for DSI COBOL 20 is broken down into four standard divisions of COBOL. The Procedure Division allows a well rounded, although basic, subset of standard COBOL statements. These include the ADD, SUBTRACT, MULTIPLY, and DIVIDE arithmetic statements, the OPEN, CLOSE, READ, WRITE, START, REWRITE, and SEEK statements for files, the ACCEPT and DISPLAY statements, and the ALTER, PERFORM, MOVE, GO TO, AND EXIT control statements.

Most of these statements have been simplified by reducing the number of options available. The MOVE statement, for instance, provides for a single MOVE of an elementary or group item, but does not allow different MOVE's to be chained either implicitly or explicitly. Similarly, the IF statement allows for only a single test rather than allowing indefinite nesting of IF's. This means that the programmer may have to write a few more statements, but it does not prevent him from achieving the desired results.

The Data Division provides for labels to be STANDARD or OMITTED, and includes the BLOCK CONTAINS clause for record blocking and the OCCURS clause for subscripting. It also provides for redefining the data names so that parts of a file can be referred to in different ways, and allows for literal or figurative constants to define the values of particular fields. The PICTURE clause is fully implemented.

Computations may be performed using unpacked (DISPLAY) or packed decimal (COMPUTATIONAL-3) items only. This restriction is based on the fact that the Model 20 normally operates in packed decimal mode rather than in the binary, floating-point, or other representations available on the larger models of the IBM System/360. In practice, the restriction is probably of little importance, since most COBOL programs have traditionally been written using COMPUTATIONAL-3.

The Environment Division defines the hardware configuration available at both the source-computer and object-computer level. A 4K or larger Model 20 can be the object ➤

DSI COBOL 20
Decision Systems, Inc.

▷ DIS COBOL 20 is designed to be "machine independent," so that programs written in this language can readily be transferred from the Model 20 to other computers. DSI claims that the language is fully upward compatible with other COBOL versions, and programs written in DSI COBOL 20 have been successfully compiled on larger models of the IBM System/360 and on the RCA Spectra 70 computers.

If your IBM 360/20 installation is about to be (or is already) overloaded or over-budget, a thorough examination of the efficiency of programs which can be produced by DSI COBOL 20 is in order. The Model 20 is a severely limited system in both core memory size and operating speed, and these limitations become critical when the installation is heavily utilized.

To evaluate DSI COBOL 20's use of core, it is necessary to examine the object code produced by the compiler. One good test is to have a few of your RPG programs written in DSI COBOL 20 and compare the core utilization directly from the object-program listings. Indications are that up to a 40% saving may be achieved; but it is always better to use your own programs for this sort of evaluation.

To evaluate the throughput, actual tests using your own programs are again recommended. DSI stresses the efficiency of object-program performance and provides a series of guides indicating how to save core or increase operating speed, so the indications are that good performance can be achieved in most applications. Nonetheless, there is no substitute for tests using your own programs.

If evaluation of DSI COBOL 20 indicates that it can appreciably improve the situation at your IBM 360/20 installation, a cost-effectiveness analysis of the package should be undertaken. The total dollar commitment is about $5,000, with monthly leasing plans reducing this to about $200 a month. Management should look for at least an equivalent saving in return for this investment, taking due account of the present workload and of additional applications which might reasonably be put on the Model 20 once COBOL is in use.

In summary, DSI COBOL 20 can significantly increase the usefulness and effectiveness of a tape- or disk-oriented IBM 360/20 without involving a large expenditure. Users contacted by DATAPRO were highly impressed by DSI COBOL 20, and reported that the system lives up to all claims made by the vendor. □

➤ computer, but the source computer must be at least an 8K (tape) or 16K (disk) system. Source code input is via punched cards.

With Version 2, data files on disk are supported; files can be organized for sequential, indexed sequential, or direct access.

The Environment Division is the only one which normally needs to be changed if the COBOL program is to be compiled using one of the senior COBOL compilers provided for the' IBM System/360. Normally, such changes are merely to the source computer, object computer, and file control sections—changing, for instance, the assignment of the MFCM channels to individual I/O units, etc.

OUTPUT: The output of the compiler consists primarily of a text and printer sections. The text section, normally on punched cards, is ready for loading and execution. The printer output consists of a source program listing, a Data Division storage map, all subroutines which are called, the generated code of the full program, and a memory map.

The compiler distinguishes between syntactical fatal errors, non-syntactical fatal errors, and warning errors. In the case of a syntactical fatal error, scanning of the program will resume to check whether or not other program errors can be found during the original compilation attempt. Currently the compiler recognizes 9 syntactical fatal errors, 37 other fatal errors, and 20 warning errors.

On an 8K (tape) or 16K (disk) source computer, DSI COBOL 20 is restricted to 110 data names, 50 paragraph names, and 35 paragraph name look-aheads at any given time. Each additional 4K bytes of memory allow an additional 300 data names and 50 paragraph names to be used. The maximum number of paragraph names permitted is 127.

HARDWARE REQUIREMENTS: Version 1 (tape) operates on an IBM System/360 Model 20 with at least 8K bytes of core storage, card input and output equipment, printer, and at least two magnetic tape drives. Version 2 (disk) requires 16K bytes of core and replaces the two tape drives with one 2311 Model 11 or 12 Disk Drive. A tape drive can replace the card output in either version. The card input and output can be via either a separate reader and punch or the MFCM. All standard I/O units (except disk with Version 1) and additional core are supported.

SOFTWARE REQUIREMENTS: No additional software is required. DSI COBOL 20 is compatible with he standard IBM software. The object program can use the absolute, CIPL, or executive loaders as desired. While the output itself is normally on cards, it is suitable for loading via the IBM CMAINT program onto a systems tape. Output can be to tape if desired.

PRICING: DSI COBOL 20 (either version) may be purchased for $5,000 for a single installation; additional installations cost $2,000 each. It may also be leased for $200 a month for the life of the IBM 360/20 upon which it is installed. A 30-day free trial is available on request.

If a company buys or leases the tape version and later adds disk capability to its Model 20, DSI will furnish the disk version of COBOL 20 without additional charge.

SUPPORT: The purchase price includes the installation of DSI COBOL 20, training of personnel, and maintenance of the program.

INITIAL DELIVERY: Tape version—November 1969.
Disk version—August 1970.

CURRENT USERS: Over 40 of both versions. ■

SAFEGUARD
Digital Solutions

MANAGEMENT SUMMARY

There are two ways to protect a secret: lock it away, or encode it. The important disadvantage of merely locking it away is that a secret is *data*, and data can be "stolen" without removal – the thief need only copy it. On the other hand, well-encrypted data is inherently secure for as long as the encrypting key is held secret, and the added expense of vault security is a mere extra precaution that can be omitted in many cases.

SAFEGUARD, from Digital Solutions, is an encrypting package that is available to run on any System/360 or 370 computer, from Model 25 up, and is guaranteed for life against any defects or unwanted code-breaking. It has been estimated that it would take 10^{24} years on a System/360 Model 50 to break the code created through the use of randomly chosen keys.

SAFEGUARD uses no operating system-supplied functions and performs no I/O operations of its own. It is thus totally independent of the user's operating system. It is written in Assembler language for maximum efficiency, and uses only about 1K bytes of main storage, including all tables.

The user alone is responsible for data security, and it is the user who supplies a 16-character key that is used to develop the tables used to map data from one form to another. All data formats can be so treated (binary, character, floating-point, packed decimal), so any data file, load module, or object module can be protected.

Current production programs can be included under SAFEGUARD's protection quite simply. For instance, once a master file has been encrypted, only three subroutine calls need be inserted into a program that accesses that file. The actual encryption is specified by means of standard subroutine calls, specifying the data items to be mapped and their lengths.

A user's key both encrypts and decrypts his files (this property is known as *isomorphism*); but since each user is supplied with different algorithms in his package, the same key does not produce identical results for different users.

Digital Solutions has been in existence since 1969. Key personnel in the firm are five graduate computer scientists. Founded primarily as a consulting firm, the company only recently began to market SAFEGUARD as well as diverse application packages that it has developed, all for a one-time charge. Distinct from Digital Solutions but comprised of the same key personnel is Computer Linguistics, founded in 1971 expressly as a software house. The two firms occupy the same office and share facilities.

USER REACTION

At this writing there are only four users of SAFEGUARD that are known to Datapro – and, security-conscious as they are, only two were prepared to comment about the package. One of the two, though, called the license fee for the package the best $250 he'd ever spent. Both users verify that the package was installed with the utmost simplicity, is just as simple to use, and operates very efficiently in the 1K bytes of main storage it occupies. Both also report that they have already received improved new object decks from the vendor.

Datapro feels that the price of SAFEGUARD is low enough, and the potential value high enough, so that computer users concerned about data security should give it a try. However, the 30-day trial period should be fully exploited and the program tested thoroughly. □

SAFEGUARD is a 1K-byte encryption/decryption package for only $250. It operates on any System/360 or 370 computer, from Model 25 up, without performing any I/O of its own. How much is your data worth to others?

CHARACTERISTICS

SUPPLIER: Digital Solutions, 24 Aviation Road, Albany, New York 12205. Telephone (518) 458-1861. Digital Solutions has a mail reply box at Box 424, Troy, New York 12180.

BASIC FUNCTION: Data security on any IBM System/370 or any System/360 from Model 25 up, using encryption.

OPERATION: SAFEGUARD is installed by simply link-editing the object deck that Digital Solutions supplies (source code is not supplied), following the instructions that are also supplied by the vendor. After that, just an hour or so need be spent with the system's operators to acquaint them with the use of SAFEGUARD (how to supply the key, etc.).

In operation, the package uses two isomorphic pseudo-random number generators to produce the encryption/decryption tables. Each generator is "seeded" by the user-supplied key, with each unique key producing a unique mapping of the data. The generators are extremely sensitive to the seed, and a slight change in the key radically alters the data mapping. The term "isomorphic" means that the same key is used to encrypt and decrypt a given data set. The generators are pseudo-random, of course, since a finite computer cannot generate a set of truly random numbers. Each copy of SAFEGUARD contains different generators, so a given key produces different results for different users.

PERFORMANCE: SAFEGUARD is said by its users to have almost no overhead associated with its use, and one user says that it is faster than user-coded translation tables. The package occupies 1K bytes of storage and is intended to be main-storage-resident.

HARDWARE/SOFTWARE REQUIREMENTS: Any IBM System/360 or 370 computer capable of running under a version of DOS, DOS/VS, OS, or OS/VS (i.e., System/360 Model 22 and larger-numbered models and all System/370's). Main memory required is 1000 bytes.

PRICING: A one-time license charge of $250 per CPU entitles the user to unlimited use, new release decks, and maintenance. Documentation and installation instructions are also provided. Quantity discounts are available.

INITIAL DELIVERY: August 1973.

CURRENT USERS: 4 as of January 1, 1974. ■

1130/SORT
DNA Systems, Inc.

MANAGEMENT SUMMARY

1130/SORT is a powerful sorting program for the 1130 that can also be used on the IBM 1800 and several other similar machines. A number of languages are supported, including COBOL, FORTRAN, and Assembler. As the most cost-effective sort capability available for the 1130, this package should be examined by any 1130 user with a commercial programming requirement.

The IBM 1130 has built a following that can only be compared with that of its predecessor, the 1620, and perhaps the PDP-8. Intended as a small computer system for hands-on scientific processing, the 1130's users have employed it for just about every application you can name, including a wide range of commercial jobs. Strangely, in spite of its involuted data codes and formats, the 1130 elicits a general response of being a "fun" machine to work with.

IBM does not provide a standard sort routine for the 1130. Several "prior use" (Type 3 and 4) routines are available, but they are typically restricted in the file and data types they can accommodate. The DNA 1130/SORT program is more of a general-purpose package, accommodating virtually every type of file that is normally generated by an IBM 1130 installation except ISAM files generated by the RPG. (DNA Systems also provides a variety of other 1130 programs, including a Full FORTRAN IV compiler and CYTOS, a conversational operating system with terminal capability, powerful text editor, and a disk source library facility.)

Sorting requirements for an 1130 installation are not the same, in general, as for a larger commercially-oriented installation. Typically, records are relatively short; DNA states that 16- to 20-word records are typical for its customers. In addition, data files are frequently created and updated directly rather than by an update associated with processing the file. An example of such a process would be the analysis of a data base created from the responses to questionnaires.

Basically, 1130/SORT allows an IBM 1130 user to sort up to seven files contained on up to five full disk cartridges of data. Any number of keys can be identified, each in ascending or descending order. All of the multitudinous standard 1130 data types can be accommodated as keys. All alphanumeric keys must be in EBCDIC, which is conventional for FORTRAN or RPG programmers. However, the many data codes used in the 1130 give rise to the possibility of non-EBCDIC data files, particularly if assembly language is employed. Sorting of non-EBCDIC alphanumeric files could result in some peculiar ordering.

The sort can be performed using the input disk area as the output (wiping out the source data), or a separate output area can be used. Provisions are made for bypassing readback checking if desired. No provision is made for establishing checkpoints for restarting in case of a malfunction. Each installation will have to make its own decisions about the importance and recoverability of the source data. Sort times can get rather long. If the source ▷

CHARACTERISTICS

SUPPLIER: DNA Systems, Inc., 1258 South Washington, P.O. Box 1424, Saginaw, Michigan 48605. Telephone (517) 793-0185.

BASIC FUNCTION: To provide a standard utility for sorting data files on an IBM 1130 computer system; virtually all of the many 1130 data formats can be accommodated. Sorting is disk-based.

OPERATION: The 1130/SORT is composed of four phases.

The first phase initializes the program and performs an edit check of the input parameters to ensure that a valid set of specifications has been formulated. Failure of any specification causes an immediate abort.

The second phase creates the initial strings by an in-core sort. All of the memory except that reserved for the Disk Monitor System and for user's common is used, thus maximizing the string length.

After the strings have been created, the third phase is entered, which merges pairs of strings to form the ordered file. Data blocks are recorded whenever space is available in this phase, and a block directory is maintained that shows proper ordering.

Phase four shuffles the data blocks so that they are in correct physical order.

Read-back checking, a hardware feature where data is read following a write operation and parity is checked, is optional in phases two and three. If skipped in these phases, then every block is read and checked in phase four; if employed, then only those data blocks that need relocation are physically move in phase four.

Parameters to the sort are input via cards in a more-or-less free-form format. A total of 11 different specifications can be input, but only four or five are obligatory.

There are several very useful standard provisions in the specifications.

The user can specify that the same or another disk file be used as the output area. If a separate output file space is used, a copy of the source data is maintained. Files are identified by file name and cartridge ID. The location of a file on a cartridge is picked up from the table maintained by the Disk Monitor.

FORTRAN-type files do not contain header records. (The same file with DNA header records is a DCS file.) With 1130/SORT, any number of records at the beginning of a FORTRAN file can be designated as header records and passed directly to the output file without getting involved ►

▷ data is still available, at least the operation can be restarted from the beginning if a malfunction occurs. Using the same area for both input and output does allow one-disk 1130 users to sort rather long files, but the technique must be balanced against the possibility of catastrophe. The 1130/SORT program can use additional disks for output, with an overall limit of five cartridges for both input and/or output.

If you have much occasion to sort, then the modest $495 price ($675 with all options) can be quickly justified. The IBM COBOL compiler ($75/month) includes a rudimentary "Tag" sort capability that in no way obviates the usefulness of a full-service sort such as 1130/SORT; the Tag sort produces an ISAM-like index file with associated pointers that locate individual data records that are physically in an unsorted original sequence.

DNA has delivered more than 450 copies of 1130/SORT since August 1970, and users contacted by Datapro report highly satisfactory results from the package. □

➤ in the sort. If you decide to add header records to your FORTRAN files because 1130/SORT can handle them, be advised that special provisions will have to be made in your programs, because otherwise your program will interpret them as data records.

The LINK specification allows automatic transfer to a program when the sort is completed. A common area can be saved and restored after the sort if desired.

An unusual specification, SEQDATA, allows a group of new records to be added to the end of a file sorted independently, and then merged with the main file.

Comments can be interspersed throughout the sort specifications, a boon to good documentation.

Three extra-cost options–Spec-File, Roll-Out, and COBOL Interface–add useful capabilities to 1130/SORT.

With the Spec-File option, sort specifications can be cataloged. A special subroutine is furnished for entering the Sort program. Upon completion of the sort, control is transferred to another program or to the Monitor. A user common area can be maintained if desired. This feature allows automatic transition from one program to another with a sort between.

The Roll-Out option allows the use of the 1130/SORT program as a subroutine within a program. This feature could be particularly useful in the area of data analysis; records could be ordered and processed according to multiple key factors. This technique is particularly useful for reducing the amount of storage required to contain a program, but is time-consuming. Basically, the Roll-Out option transfers all of core to disk, loads and runs the sort, and reloads the previous program with execution beginning where it left off.

The COBOL interface option fully supports all standard COBOL data formats, including multi-volume files and spanned records. DNA 1130/SORT with the COBOL option can be "called" by a COBOL applications program.

DATA FILES: The file to be sorted must be contained on up to seven cartridges and can be up to 1592 sectors (509,440 words) long per volume. If only one disk file is present in the system, a maximum of 430,000 records can be sorted. With multiple disk files, the maximum number rises to 3,566,080 records. Files can be located in the user's area, in the fixed area, or in working storage. COBOL, FORTRAN, DCS, or RPG (except ISAM) file formats can be accommodated.

SORT KEYS: There is essentially no limit on the number or size of sort keys. Sorts can be made in ascending or descending order.

Virtually any standard IBM 1130 data format can be used for the sort keys, including binary; standard or extended precision floating-point; A1, A2, A3, A4, A6, DI and D2 character formats; and RPG A2 and packed decimal. The designations represent various data formats available to the 1130 programmer via COBOL, FORTRAN, RPG, and subroutine packages such as Commercial and IDEAL. Keys can be signed or unsigned. Binary keys do not need to start or end on a work boundary and can extend across words up to 32 bits (64 bits with the COBOL Interface option). Alphanumeric fields also need not begin on work boundaries.

Particular care must be taken in specifying alphanumeric keys for FORTRAN and DCS files because DNA has chosen to retain the peculiar file structure of these files in the specifications for the sort keys. The difficulty arises in these files because the sense of the data items and the order of the records are opposite; i.e., a record is placed in a lower core location than the record it follows. Thus, data stored six alphanumeric characters per three-word extended "real" variable is arranged very differently in memory than the same data stored four characters to a two-word standard-precision variable. In practice this creates no appreciable difficulty, but it could if the programmer gets careless.

The collating sequence for alphanumeric keys is standard EBCDIC; i.e., blank, special symbols, alphabetic, and numeric from low to high.

PERFORMANCE: DNA, in its manual for the 1130/SORT program, provides a graph of worst-case times. Illustrative times from this graph are:

- 3.1 minutes to sort 4,800 16-word records (240 sectors) with 16 binary keys.
- 23 minutes to sort 20,900 16-word records (1,080 sectors) with 16 binary keys.

The above times were based on an 8K 1130 with the 3.6-microsecond memory and with read-back checking used during string creation and merging. The file was sorted in place, and completely random data was used. Typical times are about one-half those shown above.

HARDWARE/SOFTWARE REQUIREMENTS: The 1130/SORT program will run on any valid 1130 disk configuration under the Disk Monitor Version 2. This means a minimum of 8K words of core.

Equivalent IBM 1800, Digital Scientific Corporation META 4, or General Automation 1830 systems can also run 1130/SORT.

PRICING: The 1130/SORT is offered on a perpetual license basis. The basic package costs $495. The Spec-File and Roll-Out options are $45 each. The COBOL interface is $90. Support for the GA 1830 or IBM 1800 costs an additional $245. (No additional charge is made for the DSC META 4, because an emulator is available that allows 1130/SORT to run in 1130 native mode).

FIRST DELIVERY: August 1970.

CURRENT USERS: More than 450. ■

DYL-250 and DYL-260
Dylakor Computer Systems, Inc.

MANAGEMENT SUMMARY

DYL-250 and DYL-260 are a pair of related packages from Dylakor Software Systems, Inc. DYL-250 is an extended utility program that costs the user only $1 per day in rental and has many uses. DYL-260, so named because its cost to the user on a monthly license is $2.60 per day, is a newer release oriented toward report production. Both packages are also available on a permanent license basis, in which cash multiple-copy discounts are granted.

DYL-250 runs on any System/360 or 370 OS or DOS system having at least 65K bytes of main memory and a tape drive or card reader. Programmers can use DYL-250 to reblock, edit, translate format-to-format, and create and/or access ISAM files. There are numerous other applications for DYL-250, and some users say that their imagination is the only limit upon its uses.

DYL-260 is actually three systems in one: a Report Writer, a Data Management System, and an Extended Utility package. The Report Writer features completely automatic composing and handling of all typical report requirements. The Data Management System contains all the functions necessary to implement many applications: move, compare, add, subtract, multiply, divide, indexing, branching, and more. The Utility handles file-to-file conversions and file creations. DYL-260 can read four files and write four files plus a report in a single pass, accessing sequential, ISAM and VSAM files and fixed, variable, variable-spanned, or undefined records.

Both systems are easy to install and execute their functions in a one-step load-and-go process.

USER REACTION

Both DYL-250 and DYL-260 were highly rated by the 12 users who judged them in Datapro's 1974 survey of proprietary software users, and both were accordingly named to the 1974 Software Honor Roll. For this report, we conducted telephone interviews with five more users of the Dylakor systems. Their ratings were:

	Excellent	Good	Fair	Poor	WA*
Overall satisfaction	5	0	0	0	4.0
Throughput/efficiency	5	0	0	0	4.0
Ease of installation	5	0	0	0	4.0
Documentation	1	2	2	0	2.8
Vendor support	4	1	0	0	3.8
Training	0	2	0	0	3.0

*Weighted Average on a scale of 4.0 for Excellent.

DYL-250 is an extended utility for such tasks as file maintenance, label printing, etc. DYL-260 is a fast-response report writing and data management system. Both of these highly regarded packages can be used on IBM System/360 and 370 computers under DOS, DOS/VS, OS, or OS/VS.

CHARACTERISTICS

SUPPLIER: Dylakor Software Systems, Inc., 16255 Ventura Blvd., Encino, California 91436. Telephone (213) 995-0151.

BASIC FUNCTION: DYL-250 performs file-to-file, record selection, and data manipulation tasks that standard utilities cannot handle; it can also handle common utility tasks in a simplified manner. It eliminates special programming to correct most file problems.

DYL-260 is a flexible, user-oriented data retrieval and report composer/writer system with additional data manipulation features for file creation and updating. It works with all file organization methods in common current use.

OPERATION: Control, or data selection/manipulation, and report composition parameters are entered on a DYL-250 or DYL-260 parameter sheet. The sheet is a single 8½-by-11-inch page on which parameters are entered in a fixed format, with all columns clearly defined as to their specific function. Parameters typically default to the most commonly used value, and thus the amount of user coding can often be minimized. Parameters are then keypunched and included with the appropriate JCL to be executed in a one-step load-and-go process.

DOS DYL-250 optionally permits conversational-mode entry of parameters via the console typewriter, and is also interruptible at any point during processing.

LANGUAGE: DYL-250 is written in Assembler language. The DOS version contains over 15,000 source statements in 20 phases, and is self-relocating to run in either a background or foreground partition. The OS version contains over 6,000 source statements in 8 modules and executes in a planned overlay structure.

DYL-260 is also written in Assembler language. The DOS and OS versions each contain more than 20,000 source statements, the DOS in 30 phases and the OS in a planned overlay structure.

INSTALLATION: DYL-250 and 260 are installed in the same way. For the DOS versions, object modules are provided on tape or cards and include all necessary JCL to link and catalog the system into the core image library. For the OS versions, an unloaded version of the load program is provided on tape, and the IBM utility IEHMOVE is used to load DYL-250 or 260 to disk. An example of the required JCL is provided upon installation.

FEATURES: DYL-250 can:

- Read and write sequential and ISAM files with fixed, variable, variable-spanned (OS only) or undefined formats on all commonly used I/O units;
- Perform random updating and updating in place of ISAM files;

DYL-250 and DYL-260
Dylakor Computer Systems, Inc.

The users had the following computer configurations: IBM 360/50–1, 370/135–2, and 370/145–2. Their operating systems were: DOS–2, DOS/VS–2, and OS/VS–1.

The systems performed as advertised immediately for four of the users. The fifth cited lack of user experience and desire for more complex output in noting that the system (DYL-260) ran eventually.

The DYL-250 and 260 users contacted by Datapro were uniformly enthusiastic about the packages, no matter how they were using them. Satisfaction with the vendor earned a uniform rating of Excellent, with users reporting that newsletters and improvements have been frequent and useful, and that Dylakor's responses to telephone inquiries (except on weekends) are prompt and helpful. As to the comparatively low ratings which the users assigned to documentation, Dylakor points out that new documentation on DYL-260 is being prepared and should be released in July 1975.

Datapro found DYL-250 being used on System/360 and 370 DOS and OS systems for such varied purposes as trouble shooting, data compression to save disk space, record selection, data manipulation, file modification, test data generation, and preparation of simple reports. DYL-260 was used primarily as a retrieval and reporting system. It handled diverse tasks in the marketing, personnel, and auditing areas. The system's load-and-go operational mode was cited as a major time-saver in handling the unique requests from each of these departments. □

- Print–files or selected portions thereof in character or hexadecimal format, concurrently, with file-to-file operations, with titling and page ejection automatically handled;
- Extract–any record or group of records;
- Reformat–any record field or type of record (and can also write multiple records from one record or create one record from multiple input records);
- Add, subtract, multiply, or divide fields;
- Translate–via user-defined translate tables (and can translate non-IBM tapes to 360/370-readable tapes); and
- Compress and re-expand previously compressed records.

DYL-260 is a report-oriented program with the following features:

- Report functions–regular or cumulative totaling, up to 7 levels of control breaking plus detail line, formula handling; output field editing; output size determination; heading, footing, and page ejection; specified rounding of fields; optional print suppression of fields with unchanged values; and multi-line capability;
- Automatic composition–positioning and spacing of column headings and detail lines are determined by the system or set by the user;
- Data selection, conversion, and reformat capabilities;
- Random ISAM update;
- Translation;
- Arithmetic–data in zoned or packed decimal or binary format can be added, subtracted, multiplied, or divided by data in the same or a different format;
- File copy, print, compare, correction;
- Test file creation;
- Merging;
- User exit routines; and
- Extensive error analysis and correction functions.

Optionally, DYL-260 can be equipped to sort data or to compress and restore data.

I/O SUPPORTED: DYL-250 reads one input file, or, if updating, one master file and one transaction card or card image file. Records can be written to one or two files, known as "accepted" and "rejected" files. Additionally, either or both files can be printed simultaneously.

DYL-260 supports one to four input files and one to four output files. Both programs support card units, printers, RJE terminals, 7- and 9-track tape drives, 2311, 2314, 3330 and 3340 disk drives, and CRT terminals.

HARDWARE/SOFTWARE REQUIREMENTS: DYL-250 requires a System/360 or 370 operating under DOS, DOS/VS, DOS with any spooling system, or any version of OS (including TSO), with or without HASP. The system must have at least 65K bytes of main memory, but minimum DOS and OS partition sizes are, respectively, 22K and 38K. (The average partition size figures are about 29K and 58K). A card reader or tape unit is required.

DYL-260 runs on all System/360 or 370 computers with at least 65K bytes of main memory, including virtual storage and HASP systems.

PRICING: DYL-250 is available at $2,950 per site for perpetual license (purchase), with multiple-site discounts available. It can also be obtained on a month-to-month rental basis for $31 per month.

DYL-260 can be purchased for $8,000, with multiple-site discounts obtainable. Its monthly rental is $80.60, with additional charges of $5 per month per programmable remote terminal per site or $5 per portable copy. Either system can be inspected for a 30-day free trial.

There is no charge except handling for automatically distributed updates ($9.30 for DYL-250 and $15 for DYL-260). All maintenance is supplied without additional charge. Extra users' manuals cost $6 each for DYL-250 and $8 each for DYL-260. Extra parameter sheet pads cost $0.80 for a 25-sheet DYL-250 pad and $1.25 for a 20-sheet DYL-260 pad.

TRAINING: Normally DYL-250 or DYL-260 can be easily learned from the User's Manual, which includes a comprehensive cross-index and explained examples of filled-in parameter sheets. Additional DYL-260 training can be arranged at user sites for a fixed fee plus expenses. Regional seminars are also offered at a fixed charge per attendee. These are 2-day in-depth studies of DYL-260 with hands-on experience in a workshop environment.

INITIAL DELIVERY: DYL-250–April 1970; DYL-260–November 1972.

CURRENT USERS: DYL-250/260 had more than 800 users as of February 1, 1975. ■

Multi-DOS2
General Electronics

MANAGEMENT SUMMARY

System/360 and 370 DOS users can employ Multi-DOS2 to place up to nine different supervisors on the same SYSRES disk pack, callable by the operator from the console. This allows users to add a new supervisor to a system and test its operation without worrying about having to rebuild the entire system if it fails. The user can simply select an alternate supervisor and continue.

In another important use, Multi-DOS2 can allow a system to run special jobs, such as 1400 Series emulation or QTAM, with only those features in the supervisor that are absolutely required for the job. This can reduce the size of the supervisor, allowing more space for application programs.

General Electronics, the vendor, was founded in May 1965 to specialize in installation and maintenance of communications equipment. The firm entered the software market in February 1968 with CBLSHORT, a COBOL precompiler for DOS users. The firm also markets a 1400/360 Sort Interface program that allows the use of a System/360 sort program under CS/30 or CS/40 (1400 Series emulators for System/360 Models 30 and 40, respectively), and a Job Control Statement Editor program that can help enforce system standards.

The company's original Multi-DOS package was introduced in August 1971. The Multi-DOS2 version was initially installed in October 1972. Both versions are still actively marketed.

The difference between Multi-DOS and Multi-DOS2 is that the original package requires compilation for each supervisor that is loaded, whereas the new version simply lists the available choices on the console and requests the operator to enter a number for the desired supervisor. The other difference is price – Multi-DOS can be purchased for $50, and Multi-DOS2 for $100. Either is a small price to pay to gain expanded application testing or to avoid acquiring additional main memory just to accommodate an oversized supervisor.

USER REACTION

Two Model 40 DOS users responding to Datapro's recent survey of proprietary software users gave Multi-DOS perfect scores. This means that overall satisfaction, efficiency and throughput, ease of installation, documentation, and vendor service (free, by the way) were all rated as "excellent," and both users said that the package performed as advertised immediately and did not require modification. We think that's a pretty good score for a package costing so little and requiring only 200 bytes of core that is overlaid by the supervisor.

Multi-DOS2 allows an IBM System/360 or 370 DOS user to have up to nine different supervisors on the same SYSRES pack and makes all of them available to the operator from the console. Its extremely low cost makes it unusually easy to justify.

CHARACTERISTICS

SUPPLIER: General Electronics, P.O. Box 79, Lyons, Illinois 60534. Telephone: (312) 447-4515.

In the Far East: Orient Research Ltd., Third Floor, KLM Building, 2-18 Patpong Road, Bangkok 5, Thailand.

In New Zealand: Computer Broking Services, Ltd., P.O. Box 934, Wellington.

BASIC FUNCTION: Provides up to nine different DOS supervisors on one SYSRES disk pack on a System/360 or 370. This facilitates testing of new supervisors, with the ability to simply select an alternate supervisor and continue operation if the supervisor under test fails. Also, the resident supervisor size can be minimized by using a specially tailored supervisor with the minimum necessary features for each job, shift, or function.

Two versions are available. Multi-DOS requires recompilation for each supervisor loaded. Multi-DOS2 lists the available supervisors on the console for the operator's selection.

OPERATION: The RESADDR card is changed to the address of the device used as SYSRES, and the CONSOLE card is changed to the address of the console typewriter. The values of MSG1 through MSG9 are changed to those that the user wants to appear, staying within the 10-position limits. Then, Multi-DOS2 is compiled and executed.

The output of the run will be seven cards, with 1 through 7 in card column 80, which contain the program that will call the required supervisor. Alternate supervisors are compiled at this point, using the // OPTION deck.

The phase card as punched by the assembler must be changed from PHASE $$A$SUP1,+0,NOAUTO to PHASE $$A$SUPn,+0,NOAUTO, where n can be any number from 2 to 9. The new supervisor is cataloged as any other supervisor would be. If the new supervisor is not the same size as the standard system supervisor, the user must re-IPL after the catalog.

Any program that must be able to run under different supervisors must be cataloged under the largest one with which it will be run. This rule holds even for system control and other non relocatable programs.

To load a supervisor, the operator must place Multi-DOS2 and a SET DATA card in the reader, ready the reader, set the load switches to the reader address, and press Load. The program will type console messages and then ask the operator for the number of the supervisor that is to be loaded. The number will be the one designated by n above

▷ The package's ability to reduce the size requirements for the resident supervisor should make it of interest to all users of small DOS systems. Its ability to make the system available without risk to test new supervisors should make it worthwhile for many other DOS users as well. At the price, Multi-DOS2, with its console messages and operator facility, should be the obvious choice for those users concerned about resident supervisor sizes. It could allow third-shift operations requiring supervisor changes to be performed by less experienced operators. (Users running communications programs at night to save on line costs, take note.) □

➤ in the phase card statement. If the supervisor wasn't cataloged, a standard low-core message will be issued.

The procedure for Multi-DOS (original version) is very similar. In that version, the required supervisor is automatically loaded without operator intervention. The program to do this consists of only 3 cards and does not use the console.

PERFORMANCE: All indications are that Multi-DOS and Multi-DOS2 perform exactly as advertised.

HARDWARE/SOFTWARE REQUIREMENTS: Any IBM System/360 or 370 computer system using DOS.

PRICING: Multi-DOS can be purchased for $50, and Multi-DOS2 can be purchased for $100. Maintenance is free, and training, though usually not required, can be obtained from the vendor at a small additional cost.

FIRST DELIVERY: Multi-DOS–August 1971; Multi-DOS2–October 1972.

CURRENT USERS: Over 125 for both versions as of July 1973. ■

Intercomm
GTE Information Systems, Inc.

MANAGEMENT SUMMARY

The Intercomm system is designed to provide IBM System/360 or 370 users with on-line programs installed and operating in a matter of weeks rather than years (if developing a data communications monitor from scratch) or months (if using IBM's popular CICS–Report 70E-491-02). It is also designed to provide this advantage at a small fraction of the cost of a normal in-house development effort.

Its essential characteristics lie in the efficiency and flexibility of the system. Efficiency is related to the capability of the system to handle a number of messages at one time. The program may or may not manage to "eliminate the Wait light" as the advertising literature claims. It does include the message handling and program switching facilities needed to insure adequate response times under varying conditions. These functions are handled within Intercomm rather than by the IBM operating system task management routines, and this prevents a considerable amount of apparently unnecessary overhead as operating system programs are moved in and out of core storage.

Flexibility within the system is shown in two different ways. First, the programs are handled independently. That is, application programs can be written for handling messages without being related to particular terminals. Moreover, application programs can be written in reentrant or non-reentrant Assembly Language, COBOL, FORTRAN, or PL/I. This flexiblity is important because it enables different programmers to use the languages they prefer.

A second type of flexibility is in the accessing of files. The access methods used by Intercomm include all the standard access methods of OS and OS/VS. This again is important from the point of view of efficient operation. Data base management systems such as TOTAL (Report 70E-132-01) and IMS (Report 70E-491-01) are also supported.

A key consideration about Intercomm is that it is already installed and working in some 60 installations. The teleprocessing portion of the Intercomm system is tailored to each individual installation. However, the basic system is always the same. One of the most difficult things to determine, in any on-line design system, is the frequency of errors and the types of errors that will occur. The types of errors which occur are often related to the types of terminals used and other factors. By reviewing the history of the particular terminals involved, PMI should be able to advise users of these important system design parameters.

All in all, Intercomm is a well-designed and functional system that can significantly reduce the cost and time required to implement an effective data communications system. □

Intercomm is a data communications monitor that works within a single partition or region on a System/360 or 370 under OS or OS/VS and controls the handling of messages from terminals, the accessing of I/O files, and the routine utility operations of an on-line message system. It imposes no restrictions on access methods, number of terminals, or source languages for the user's programs.

CHARACTERISTICS

SUPPLIER: Programming Methods, a division of GTE Information Systems, Inc., 1301 Avenue of the Americas, New York, New York 10019. Telephone (212) 489-7200.

BASIC FUNCTION: Intercomm acts as a 3-way interface between messages from teleprocessing terminals, the application programs which determine how the messages are to be processed, and either OS or OS/VS. It controls the queueing, editing, and dispatching of the messages, the centralized input/output operations involved, and the rolling in and out of programs.

OPERATION: The Intercomm Monitor may be considered to consist of four main modules: a Teleprocessing Interface, a System Controller, a File Handler, and a Dispatcher. The Teleprocessing Interface module communicates with the terminals and develops the queues of messages. The System Controller module retrieves messages from the queues, calls the appropriate user application programs and passes the messages to them, and handles all inter-program messages between application programs. The File Handler module processes all of the I/O requests from the application program. The Dispatcher module handles the inter-program switching required for efficient multi-thread message processing.

The Teleprocessing Interface of Intercomm passes all of the communications processing between the CPU and the outlying terminals. The user's own programs do not communicate with the terminals except through the Intercomm system.

Intercomm provides a number of useful services, including the polling and receiving of messages from terminals, the addressing and sending of messages to terminals, the editing of input and output messages, and the handling of error situations and rerouting of messages when a terminal is down.

The Interface module is totally responsible for the transmission, receipt, and control of data from on-line local or remote teleprocessing terminals. The Teleprocessing Interface is tailored by the user through I/O GEN tables based on his own particular teleprocessing configuration. Normally, either BTAM, QTAM, or a special communications hardware front-end is provided to supply the linkage with the telecommunication terminals. The Interface is transparent to the user's application programs, facilitating changes or additions of new terminals.

The Intercomm File Handler module is used by the application programs themselves. The handling of input/output by a real-time program is complex and requires proper con-

➤

Intercomm
GTE Information Systems, Inc.

sideration of all the other input/output operations being processed on the same machine at the same time.

Sequential data sets can be accessed by means of GET and PUT functions, referring to logical records, or by READ and WRITE functions, referring to physical blocks. Where updating is required, an exclusive PUT or WRITE is used for safeguarding purposes. Indexed data sets that are handled randomly also use READ and WRITE functions to access physical blocks of direct data sets. Indexed sequential records use the GET and PUT commands.

All the standard types of I/O files included in the OS or OS/VS data management facilities are supported by Intercomm.

The File Handler is a series of service sub-modules. Specific features of the File Handler include: provision for deferred mounting of on-line data sets, provision of additional access method capabilities not currently available under COBOL, and the overlapping of input/output operations with on-line applications processing.

Support of data base management systems includes CINCOM's Total and IBM's IMS-2. All interface requirements of these systems are handled by Intercom.

The System Controller and Dispatcher are used to handle each message as it moves through the processing cycle. This cycle includes obtaining the message from a teleprocessing terminal, formating it for the application program, obtaining any required records from on-line files, executing the appropriate application program, returning the updated file records, formating the response, and dispatching it to the proper terminal.

Characteristically, the Intercomm system handles a number of different messages in parallel, using either reentrant or nonreentrant routines. For example, while the central processor is actively working on one message, other messages are generally in the process of being read, queued, edited, processed, or output at the same time. And all of these message handling functions are transparent to the user's application programs, which can therefore concentrate upon the required data processing functions.

The System Controller bypasses the OS or OS/VS functions of task management and transfer of control. Instead, it provides its own task management and controls the various messages through the operation. The standard operating system regains control only when there is no message available for further processing, or when all messages are awaiting the completion of I/O operations.

Intercomm restricts two messages from simultaneously attempting to update the same record within a data base, but otherwise it permits any number of messages to be directed to the data base in parallel.

Message priority is established at the program level. When a message is defined as requiring a particular type of processing, it is assigned a priority. The Dispatcher establishes separate queues for each priority level in the system. The tasks in the queues awaiting execution are dispatched on a first-in/first-out sequence within each priority level.

Distribution of work is handled by a system control table for each program/module under the System Controller. The table contains an entry for each application program and indicates whether or not each program is actively processing a message. The System Controller searches the table for a message application or task. It obtains a message from the process queue based on priority and activates the appropriate program to start processing the message.

Real-time support utilities provided with the Intercomm program include an Edit routine, used to prepare messages from teleprocessing terminals for processing by the application programs. This routine strips off the teleprocessing control characters and edits the messages into a predefined format.

The Output support utility module handles the operations in the opposite direction. Each application program in the system can use this module to output the data fields that are necessary for a report. The Output module will determine what report to format, obtain the data to go in the report, and handle the actual formating for the selected device. The module also inserts the necessary teleprocessing control characters before passing the message over to the Telecommunications Interface for transmission to the designated terminal.

Two of the utilities modules, Display and Change, provide remote terminal operators with the capability to display an individual file record for any type of file supported by Intercomm in a fixed character format on his terminal. He may then modify selected fields within the file record. These modules are similar to Edit and Output in that they are totally table-driven and no program modification is required to any fixed-format file. The only item required by the user for the display of record data (in binary, hexadecimal, or character form) is the Record Description Table entries defining the characteristics of the individual file records.

HARDWARE/SOFTWARE REQUIREMENTS: Intercomm runs on an IBM System/360 or 370 under OS/MFT, OS/MVT, OS/VS1, or OS/VS2. The Basic system requires 40,000 bytes of main memory. An additional 10,000 bytes are required for dynamic storage for queues, control blocks, buffers, I/O areas, etc.

PRICING: Intercomm is available for purchase or lease, with or without the Real-Time, Output, Display, and Charge utility modules. The purchase price of the basic system is $32,000; the corresponding lease price is $800 per month for 48 months on a full-payout basis. The additional charge for the utilities is $8,000 (resulting in a $40,000 full purchase price) or $200 per month.

Interfaces for the following software products are available at the indicated purchase prices: for IBM's IMS (DL/1), $4,500; for Cincom's Total, $3,500; for IBM's RJE, $2,500; for Comress' AMIGOS, $1,500. These interface features, as well as future enhancements to the package, can be obtained for $100/month (minimum 48 months).

INITIAL DELIVERY: June 1969.

CURRENT USERS: About 60. ■

Score
GTE Information Systems, Inc.

MANAGEMENT SUMMARY

Score is a useful programming tool that can greatly simplify the preparation of COBOL programs to handle information retrieval, report writing, file maintenance, data transcription, file conversion, and a variety of other commonplace data processing functions.

The Score system accepts request forms filled out by the user and generates custom-tailored COBOL source programs, which are then compiled and executed to perform the requested functions. The request form uses straightforward, English-like terminology and can be used by nontechnical personnel with comparatively little training.

Since its introduction in 1968, Score has been aggressively marketed and has become one of the most widely used proprietary software packages. The number of Score users now exceeds 300, and a Score Users' Group has been formed.

The primary features of Score III (introduced in January 1970) include user exits, provisions for multiple I/O files, and numerous retrieval and formatting options. The 80 user exits enable more sophisticated users to insert their own COBOL procedural coding at appropriate points in the Score-generated programs. This facility places virtually all the computational capabilities of the COBOL language at the Score user's disposal–though he will need to be an accomplished COBOL programmer to take full advantage of them.

An improved version called Score IV was introduced in March 1972. Among the new capabilities of Score IV are: free-form COBOL FD's from the card reader or source statement library; variable header information; up to 90 subtotals; edit masks that can be either automatically generated or programmer selected; computational logic available at detail and/or summary time for percentage calculations, cross-footing, tax equations, etc.; and automatic development of the linkage section and/or working storage for accessing non-COBOL files such as IMS, TOTAL, etc. (see Reports 70E-491-01 and 70E-132-01, respectively).

Score's principal uses to date have naturally been in the areas where it can yield the greatest savings in programming time and costs. One-time reports are first on the list, although the repeated execution and maintenance of these reports are also quite easily handled. Other uses include auditing of existing data processing functions, test data generation for new systems testing, file-to-file maintenance, and conversion of older programs (e.g., IBM 1401, UNIVAC 1004, IBM Assembler language) to COBOL. For programs that are to be run more than once, the Score generation process is not repeated, since the COBOL that ▷

Score is a multi-purpose COBOL program generator that enables technical or non-technical personnel to produce information retrieval and file management programs quickly. It can be used on just about any computer system that includes a COBOL compiler.

CHARACTERISTICS

SUPPLIER: Programming Methods, a division of GTE Information Systems, Inc., 1301 Avenue of the Americas, New York, New York 10019. Telephone (212) 489-7200.

BASIC FUNCTION: Score is designed to create COBOL programs to process data files on any of the standard computer media. The resulting COBOL programs can handle reporting, file maintenance, data transcription, and file creation functions in accordance with the user's specifications.

INPUT: Score input consists of file definition forms, which describe each of the files in an installation, and request forms, which define the selection criteria, computational procedures, and headings for specific programs.

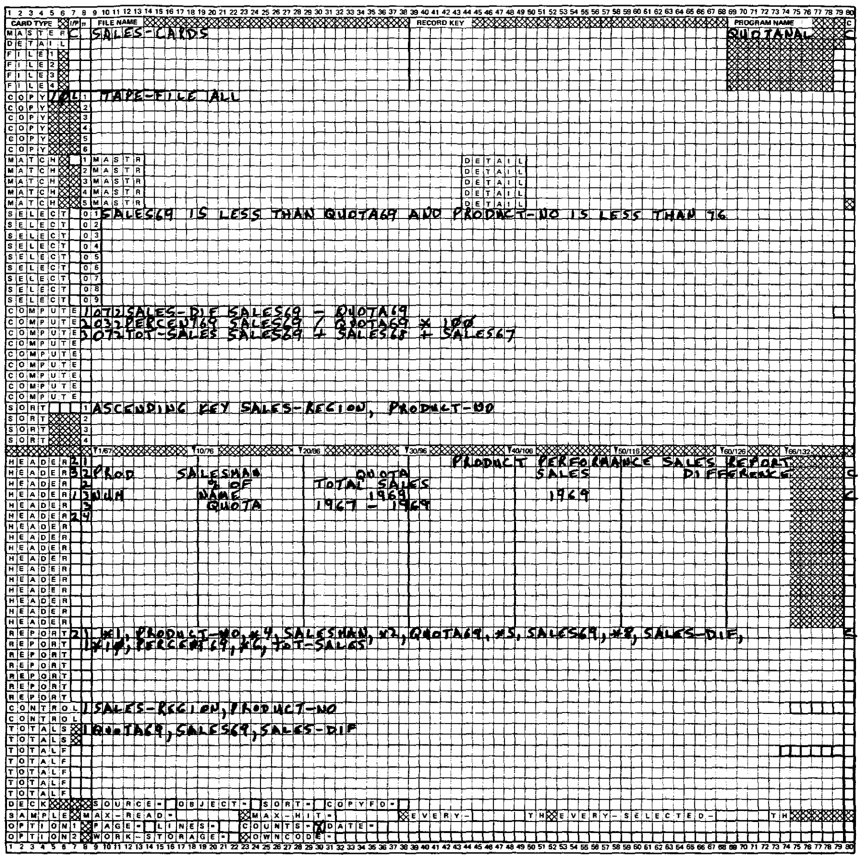
The file definition form needs to be filled out only once for each of the user's files. The definitions are then stored in the form of a card, disk, or tape library which can be maintained by Score or the system's COBOL "copy" function. The files may include fixed and/or variable length records. Each field within a record is defined in terms of a standard COBOL Picture. A single Score program can handle up to seven input and output files.

The request form is filled out for each program to be generated by Score. The operations available in a Score program include:

- Specification of the master and detail files to be used.
- Selection of the master-file records to be processed. Selection can be on the basis of matching or non-matching of keys, simple or compound conditions, results of computations, or sampling criteria (e.g., every tenth or every hundredth record could be selected).
- Definition of the control fields. Up to nine lines of print can be generated on any one control break; 15 separate totals can be accumulated and printed within each control break.
- Definition of the items in the report which are to be totalled.
- Definition of the computations to be performed. The standard operations of addition, subtraction, multiplication, division, and exponentiation can be specified. The computed results can be automatically rounded and edited by Score.
- Definition of the sorting required to produce each report.
- Definition of report headings.

There are 80 "user exits" provided to enable the user to expand Score's capabilities by inserting his own special ▶

Score
GTE Information Systems, Inc.



```
CARD TYPE   I/P n FILE NAME                          RECORD KEY                      PROGRAM NAME   C
MASTER      C     SALES-CARDS                                                        QUOTANAL       C
DETAIL
FILE1
FILE2
FILE3
FILE4
COPY        TØL 1 TAPE-FILE ALL
COPY            2
COPY            3
COPY            4
COPY            5
COPY            6
MATCH           1 MASTR                              DETAIL
MATCH           2 MASTR                              DETAIL
MATCH           3 MASTR                              DETAIL
MATCH           4 MASTR                              DETAIL
MATCH           5 MASTR                              DETAIL
SELECT         01 SALES69 IS LESS THAN QUOTA69 AND PRODUCT-NO IS LESS THAN 76
SELECT         02
SELECT         03
SELECT         04
SELECT         05
SELECT         06
SELECT         07
SELECT         08
SELECT         09
COMPUTE 1072 SALES-DIF SALES69 - QUOTA69
COMPUTE 2032 PERCENT69 SALES69 / QUOTA69 * 1ØØ
COMPUTE 3072 TOT-SALES SALES69 + SALES68 + SALES67
COMPUTE
COMPUTE
COMPUTE
COMPUTE
COMPUTE
COMPUTE
SORT            1 ASCENDING KEY SALES-REGION, PRODUCT-NO
SORT            2
SORT            3
SORT            4
            1/67      10/76      20/86      30/96      40/106      50/116      60/126      66/132
HEADER 21                                  PRODUCT PERFORMANCE SALES REPORT
HEADER 32 PROD     SALESMAN        QUOTA              SALES              DIFFERENCE     C
HEADER  2           % OF       TOTAL SALES
HEADER 13 NUM       NAME             1969             1969                              C
HEADER  3           QUOTA      1967 - 1969
HEADER 24
HEADER
HEADER
HEADER
HEADER
HEADER
HEADER
HEADER
HEADER
HEADER
HEADER
REPORT 21 *1, PRODUCT-NO, *4, SALESMAN, *2, QUOTA69, *5, SALES69, *8, SALES-DIF,     C
REPORT    1*1Ø, PERCENT69, *6, TOT-SALES
REPORT
REPORT
REPORT
REPORT
REPORT
CONTROL 1 SALES-REGION, PRODUCT-NO
CONTROL
TOTALS  1 QUOTA69, SALES69, SALES-DIF
TOTALS
TOTALF
TOTALF
TOTALF
TOTALF
TOTALF
DECK      SOURCE-□ OBJECT-□ SORT-□ COPYFO-□
SAMPLE    MAX-READ-      MAX-HIT-       EVERY-      TH  EVERY-SELECTED-      TH
OPTION1   PAGE-□ LINES-□  COUNTS-☒ DATE-
OPTION2   WORK-STORAGE-□  OWNCODE-
```

The 16 Score input specifications in this request led to the generation of a COBOL program with 349 source statements. The program, in turn, produced a sales report with seven columns of data, three levels of totals, and an output tape file. A truncated Score input form is shown here to save space. Up to 9 COPY statements, 20 SELECT statements, and 5 COMPUTE statements can be entered on the standard Score form.

▷ is produced can be compiled and stored in object form. Thus, the Score generation process does not need to be repeated unless the report requirements change. □

➤ coding along with the standard Score parameters. This user coding is written in COBOL and inserted into the Procedure Division of the resulting COBOL program by the Score generator. Working storage can also be added by the user.

Score-generated programs can maintain a count of the detail records comprising each control break, as well as grand totals of all records selected. Report pages are numbered automatically. Line counts are checked, and when a page is filled, the appropriate headings are automatically printed on the next page.

Score can be used to create and maintain card, disk, or tape files. Disk files can be either sequential or indexed sequential. Score can be instructed to reformat or reblock records, copy records or fields from other files, replace or delete existing fields, and insert new fields. In addition to data from input records, the user can specify that the output records shall contain literals or computed values.

OUTPUT: The output of the Score program generator is a COBOL source program. This, in turn, is compiled (using a standard COBOL compiler) and executed to produce the requested output files and/or reports. The user may request a source or object deck of the resulting COBOL program, or the program can be cataloged into a library.

PERFORMANCE: The generation, compilation, and execution of a Score-generated program typically takes from one to three minutes. Score itself usually adds only about 30 seconds to the normal COBOL compilation time. Thus, Score's overall performance is heavily dependent upon the efficiency of the COBOL compiler with which it is used.

HARDWARE/SOFTWARE REQUIREMENTS: Score IV runs on an IBM System/360 or 370 under DOS (47K) or OS (88K), Burroughs 2500 or 3500 (62.5K), Honeywell Series 6000, or CDC 6000 (50K words). Score III runs on the same systems as Score IV, but can also run on smaller configurations: IBM System 360 or 370 under DOS (32K), Honeywell Series 200 or 2000 under MOD 1 or OS/2000 (28K), UNIVAC Series 70 under TDOS, DOS, or TSOS (50K), UNIVAC 1108 (20K words), or an NCR Century 200 or larger (32K). In addition to the main memory partition sizes listed above, a card reader, printer, and two sequential work files are required. A card punch is optional, and tape or disk units can replace the reader and printer.

A minimum COBOL compiler is required to translate the output of Score into executable programs.

PRICING: The purchase price of Score III is $12,500 for a single system. Score IV costs $15,000. Multiple systems within a single corporation are discounted by 50 percent. If the user wishes, the purchase price can be paid in installments over a 1, 2, or 3-year period. The price includes on-site installation assistance and maintenance of the Score program in an error-free condition for one year. Additional maintenance may be purchased if desired for $500 per year.

INITIAL DELIVERY: Score I—August 1968; Score IV—April 1972.

CURRENT USERS: 240 Score III users and about 60 Score IV users. ■

Minicomm
GTE Information Systems, Inc.

MANAGEMENT SUMMARY

Minicomm has apparently grown up. Originally a CRT display control system for IBM 2260-type terminals operating in a local mode under System/360 DOS, it is now offered in a version similar to its original nature and in an enhanced version that can support local-only or local and remote CRT's under a multiprogramming DOS supervisor. In its most advanced version, Minicomm II for local/remote terminals, it is thus competitive with the DOS-Entry version of IBM's CICS (see Report 70E-491-02). Minicomm's chief advantages over the IBM communications monitor are its much lower core requirements and shorter implementation lead time. Moreover, September 1973 should see release of a virtual storage version of Minicomm.

Programming Methods, Inc. (PMI) is a division of GTE Information Systems, Inc. Minicomm was acquired by PMI in March 1972 from AIDS Computer Services of Stamford, Connecticut. The system was called DCS (Display Control System) by its creators.

Minicomm is used in on-line environments, where in-place editing of input data streams is done. Parameters are entered during the processing run to control a program's execution, while status inquiries to a data base can be made. This provides a human interface to an application program during its execution. Using Minicomm, the host-language (e.g., COBOL) program can make simplified, high-level references to CRT displays.

A particularly noteworthy feature of Minicomm deserves special attention. Minicomm I, for use in a batch-oriented system, actually simulates a basic multiprogramming environment in a non-MPS system. A core-sharing technique employing swapping is used to allow one major batch program to share main memory with the various CRT applications programs. Thus, with Minicomm I, significantly greater flexibility and capability can be provided for small, entry-level DOS systems without the overhead of a multiprogramming supervisor. (Minicomm II runs in a foreground partition of a multiprogramming DOS system.)

USER REACTION

Users of Minicomm contacted by the Datapro staff report that both versions of Minicomm are very efficiently coded, and that Minicomm II provides a response time about twice as fast as that of LOTUS, a no-charge IBM prior-use program that is the most similar competitive product. All Minicomm users praised the reliability of the system and stated that the claims of the vendor were very well fulfilled, with no program fixes required.

Datapro interviewed several Minicomm users, including one who has the package interfaced to SDI's Grasp (Report 70E-760-01) and another who has Minicomm interfaced to USI's ASAP (Report 70E-879-01). Users of all Minicomm versions were contacted, and they gave the package an overall good to excellent rating. ▷

Minicomm, originally a local display system monitor, has evolved into a multi-version package competitive with IBM's CICS DOS-Entry communications monitor. Non-multiprogramming Minicomm I is for local displays; Minicomm II is a multiprogramming version for handling local-only or remote/local units.

CHARACTERISTICS

SUPPLIER: Programming Methods, Inc., a division of GTE Information Systems Inc., 1301 Avenue of the Americas, New York, New York 10019. Telephone (212) 489-7200.

BASIC FUNCTION: Minicomm is a CRT monitor that handles display terminal networks for System/360 and 370 DOS installations. Three versions are available to operate in a local-display batch-oriented environment without multiprogramming (Minicomm I), in a local-display multiprogramming environment (Minicomm II, in one variety), and with local or remote displays in a multiprogramming environment (Minicomm II, in another variety).

All versions of the Minicomm monitor provide private library functions and an access method consisting of a set of user subroutines with exit and entry points for on-line interfacing between user application programs written in high-level host languages (COBOL, etc.) and a network of IBM 2260- or 3270-type terminals. Any type of conventional file organization or data base structure can be supported, including BOMP (IBM's Bill of Materials Processor, Report 70E-491-21), DBOMP (Data Base Organization and Maintenance Processor), etc.

OPERATION: Minicomm I is initiated by the DOS Job Controller from a JCL stream for batch operation, while Minicomm II is initiated in a foreground partition by a multiprogramming DOS supervisor. Minicomm II is self-relocating, thus eliminating the need to recatalog the display control system if partition boundaries are changed. For Minicomm I, 1350 bytes are added to the DOS Supervisor during a Sysgen to support the monitor, while 320 bytes are added to the Supervisor for Minicomm II.

Minicomm I and II automatically roll each respective application program into main memory whenever control is passed to a particular terminal. The programs are not written back out to the library, however, unless (in the case of Minicomm II only) the processing logic of a program actually causes the program to modify itself. In that case, Minicomm II can rewrite the program onto the library.

For programs in which user parameters must be retained for a given terminal between turns at the processor, a working-storage save area is used. Minicomm II has a High-Priority Program Feature that automatically causes a predesignated high-priority program to be swapped in whenever no other requests for service are pending. This feature plays the odds that a frequently used program (e.g., used by 1/3 or more of the terminals) will be available when needed.

For either Minicomm I or II, three disk files must be created, using the IBM Clear Disk Utility program: (1) a High Speed Program Library (FLIP) File with one cylinder for each user application program; (2) a Working Storage Save and Get (SAVE) File with one cylinder for each user application program; and (3) a High Integrity Data Capture (CAPTURE) File with enough cylinders to capture all the input records expected from the terminals. ►

Minicomm
GTE Information Systems, Inc.

▷ A user of the original AIDS DCS-1 (Which took a core snapshot and placed it on disk, then processed the CRT task) reports having upgraded two years ago to Minicomm II, which runs in its own 22K background partition with a 3K monitor.

Three other Minicomm users contacted chose one version or another in preference to CICS-DOSE for the reasons mentioned above. Another Minicomm II user rejected CICS-DOSE after a trial, and finds that he's further along in implementation with Minicomm after the same period. His program is running in 30K bytes (8.5K of which is the monitor), and supporting remote 3275 terminals under IBM's DBOMP. This user once had LOTUS, but that package wouldn't support System/370 hardware. The user particularly likes the way in which Minicomm permits writing programs almost as if they were batch jobs, yet, despite this transparency, uses advanced programming and hardware features available with the new IBM terminal equipment. □

➤ Other facilities provided with Minicomm include:

- A Warm Restart Programmer Aid that permits the suspension of Minicomm and allows individual CRT application-program maintenance to be performed without having to terminate and reinitiate the Minicomm system. (This facility should not be confused with warm restart capabilities in data communications monitors that assist in recovery following a system failure.)
- A CRT Image Simulator that accepts card-image records with comments and formats them into CRT screen images. This batch simulation program is very helpful in the design of CRT layouts for new applications.
- A Data Capture File that stores user input data records on a sequential disk file for audit trail and backup purposes. A batch program to retrieve records from this file is also included.
- An Error Recovery and Retry feature that includes a "CANCEL Trap" for all DOS CANCEL conditions. This feature shields Minicomm's operations and prevents the system from going down during on-line usage due to DOS CANCEL messages.
- Terminal independence for a conversational program environment where different types of terminals access the same application program. Block lengths of up to 7200 bytes of working storage can be handled by the GET/SAVE subroutines. A terminal ID number is associated with each block of working storage, and this ID is used by Minicomm as a security key for each user's version of an application program.
- High-level user language interfaces for COBOL, PL/1, RPG, and BAL that allow CRT applications programs written in these languages to address the displays with straightforward read and write statements.
- Capability to support terminals attached to a 360 or 370 via a 270X or ICA on leased or private telephone lines, overlapping I/O with processing and providing a disk queue to accomodate peak message processing.
- An IBM 3270 mapping facility for programmer convenience.
- Shadow jobs – the capability to effectively spool, and perform as low-priority jobs within the on-line partition, CRT hard-copy printing, optical scanning (IBM 1287), CalComp plotter operations, IBM System/7, and user-written applications.
- High-speed overlay, to provide a virtual-like capability for users with, in some cases, as little as 4K or 8K bytes of main storage for applications larger than that.
- A screens file utility, which documents and stores screen images required for on-line applications.

Minicomm I can control up to 16 on-line CRT terminals, while Minicomm II can accommodate virtually any number of devices subject to reasonable response-time criteria. Up to 98 different applications can be loaded at once by either version of Minicomm, with some terminals having multiple operational programs.

PERFORMANCE: On a 32K-byte System/360 Model 30 with IBM 3270 displays, the typical response time for a 24K application program is just under 1 second for Minicomm I and 0.7 second for Minicomm II. In a 10-terminal network, users can expect delays of not more than about 10 seconds when all of the terminals are active simultaneously.

PRICING: Minicomm is available on either a direct "purchase" or 24-month full-payout lease. In either case, 12 months of maintenance are provided at no additional charge, with an annual maintenance rate of $500/year for subsequent years. Each Minicomm system is licensed for perpetual use on a single CPU. Additional copies of Minicomm are provided at a 25% discount from the first-copy price. Three man-days of system installation and education are provided.

	Single Payment	Monthly Payment*
Minicomm I (Batch-mode operation)	$ 8,500	$425
Minicomm II (Multiprogramming operation)**	10,500	525

*24-month full-payout lease, totalling 120% of the single-payment charge.

**If the user requests, Minicomm II can be delivered in the local-only variety at Minicomm I prices. Upgrade to local/remote Minicomm II would subsequently be billed at the current price differential.

HARDWARE/SOFTWARE REQUIREMENTS: Minicomm I or II runs under DOS on an IBM System/360 or 370 with at least 24K bytes of main memory. Minicomm I is imbedded in the DOS Supervisor and requires 1350 bytes plus 2 bytes per CRT and 2 bytes per application; a typical Minicomm I system with 10 CRT's and 20 application programs will fit in an 8K DOS Supervisor. Minicomm II requires a multiprogramming DOS supervisor and occupies 2.5K bytes for a 480-character screen CRT or 4K for a 1920-character CRT, or 4.5K to 8.5K in remote applications; partition size can be as large as 140K bytes. About 320 bytes of the DOS supervisor are also required for Minicomm II. If ASCII terminals (generally those other than the IBM 3270) are used, an additional memory requirement of about 1K bytes is needed for a code translation table, etc.

INITIAL INSTALLATION: Minicomm I – June 1970; Minicomm II – April 1971.

CURRENT USERS: Over 20 for both versions. ■

UCANDU
Gulf Oil Computer Sciences, Inc.

MANAGEMENT SUMMARY

The free IBM-supplied dump and copy software does just that: dump and copy, and nothing more. Gulf Oil Corporation, a major IBM System/360 OS user, decided it needed an OS dump/copy utility with additional facilities, and UCANDU was born. The program was developed by Gulf Oil of Canada, Ltd., a Gulf Oil Corporation subsidiary.

Besides making it easy to list, dump, or copy OS data files, UCANDU has highlights that enable its use as a versatile tool for applications programmers, systems programmers, and operations personnel. Among the UCANDU highlights are device independence, free-form parameter-driven coding, ability to selectively dump or copy files based on record content or position, data selection capability at the byte level, selectability of random records, record search capability, default parameters to simplify coding, formatted dumps and listings to make readings easier and more presentable, and ability to handle both sequential and indexed sequential I/O.

UCANDU can have a multiplicity of applications, such as auditing, data set backup and protection, file sampling, selective file dumping or listing, source program merging, test file generation, and verification of program output. Many of these seemingly simple tasks cannot be handled by IBM's free dump/copy utility; examples include creating a test file from a live file, or merging source programs (which, incidentally, is a program librarian function).

USER REACTION

It is in the areas of function and versatility that UCANDU is praised by its users. They are using it in order to take advantage of its added facilities; all seem fully cognizant that UCANDU offers no processing speed advantage over the IBM programs, except when copying an ISAM file. For this operation, UCANDU is about twice as fast as IBM's IEBISAM.

Major manufacturers of automobiles and heavy equipment responding to Datapro's 1973 Survey of proprietary software users reported good or excellent experience with UCANDU on System/370 Models 145, 155, and 165. Vendor service was described as either not required or ▷

UCANDU is an OS dump/copy utility for the IBM System/360 and 370 that offers sufficient additional flexibility to make it of considerable use to applications and systems programmers and operations personnel.

CHARACTERISTICS

SUPPLIER: Gulf Computer Sciences, Inc., P.O. Box 2100, Houston, Texas 77001, telephone (713) 228-7040; Gulf Computer Sciences, Inc., P.O. Box 1166, Pittsburgh, Pennsylvania 15230, telephone (412) 391-8677; or Gulf Computer Sciences, Inc., P.O. Box 81608, San Diego, California 92138, telephone (714) 453-8211. The package is a proprietary program of Gulf Oil Canada, Ltd., a Gulf Oil Corporation subsidiary.

BASIC FUNCTION: UCANDU is a generalized utility for selective listing, dumping, and copying of OS files on IBM System/360 and 370 computers.

Additional uses to which UCANDU can be applied are:

- Selective dumping or listing of test files–the user can elect to dump or list only those records necessary to properly check a program.
- Generation of test files–records can be extracted from a large file to provide a sampling of all record types, and old test files can be modified to check out new program functions.
- Data set backup and reorganization–data sets can be copied and/or dumped, with automatic default options in case file characteristics are unknown; and ISAM files can be reorganized or backed up, with prime and overflow area statistics provided automatically.
- Verification of production–key fields can be sampled for invalid or illogical conditions, and files can be scanned for incorrect output.
- Source program merging–source program libraries can be built by selectively merging data sets, a program librarian function.
- Auditing–records can be copied or listed from a file based on record content and/or position; numbers of records can be reported and/or controlled; random records can be selected.
- File sampling–records that conform to specified criteria can be listed; criteria parameters can be logically ANDed and ORed; data can be selected from mixed-format files.

OPERATION: After it has been link-edited, UCANDU is invoked as a normal OS job, with the JCL followed by as many parameter cards as are necessary. One invocation of UCANDU can handle any number of functions on any ➤

UCANDU
Gulf Oil Computer Sciences, Inc.

▷ excellent. UCANDU was reported to perform as advertised immediately, without any need for modification. Earning special praise was the package's documentation, which includes self-teaching manuals. □

➤ number of files. There are four general types of parameter cards:

- Action Director cards describe functions to be performed; e.g., DUMP, LIST, COPY, and parameters such as HEX, NOHEX, ALL, and LIMIT.
- Data Selection Criteria cards describe individual fields within a file's logical records. Also, relational operators such as EQ, GT, LT, and NE can be used.
- Logical Connector cards logically AND or OR Data Selection Criteria cards together. Since AND is an automatic default, the OR must be used where needed while AND statements serve merely as documentation.
- Comment cards are identified by an asterisk in the first character position. They are printed with parameter cards in the order introduced.

As an example of the simplicity of UCANDU's JCL, the following will cause all records in a file except records 50 to 100 to be dumped (i.e., dump records 1 to 49 and 101 to end of file):

```
//SYSIN    DD*
    DUMP,ALL EXCLUDE,FROM=50,TO=100
/*
```

Self-monitoring facilities check at 10-minute intervals for unusual or unintentional "wait" conditions. If such a condition is encountered, a message is sent to the operator. If the condition persists for 30 minutes, a utility-controlled ABEND results, preventing unproductive monopolization of machine resources. ABEND dumps are automatically directed to the SYSPRINT data set when that data set is directed to the SYSOUT data set, but the dump is not automatic if SYSPRINT is a tape device.

PERFORMANCE: On a straightforward dump or copy, UCANDU operates at about the same processing speed as the IBM-supplied utilities. The performance edge offered by UCANDU is in function and versatility, not speed. For example, UCANDU control cards are easier to work with, and the package can provide functions such as creating test files from a live file, which the IBM utilities cannot.

HARDWARE/SOFTWARE REQUIREMENTS: UCANDU requires an IBM System/360 or 370 OS configuration. The program is distributed on a mini-reel of magnetic tape. About 48K bytes of main memory is required, plus buffer areas for data sets.

PRICING: After a 30-day free trial period, UCANDU can be purchased on a perpetual license for $1,250 or rented for $50 per month with a 30-day notice cancellation privilege. (The first nine months' lease charges can be applied toward purchase of a perpetual license.) Charges for additional installations are determined on an individual basis, but usually amount to only a $450 one-time charge per installation.

Ten users' manuals per installation are provided, with additional copies available at extra cost. First-year maintenance is free; it's usually a simple matter of receiving a new mini-reel from Gulf. No training is provided, but the manuals are of the self-teaching type.

INITIAL DELIVERY: UCANDU was first used internally in November 1969. The first commercial installation was in October 1972.

CURRENT USERS: Approximately 25 companies are using 28 copies of the program. ■

Testmaster/COBOL
Hoskyns Systems Research Inc.

MANAGEMENT SUMMARY

Modular programming is a clever idea. Separation of processing elements into neat packages offers many advantages for parcelling out the different pieces among several programmers, for establishment and adherance to programming standards, and for clarifying program logic. True, high modularity is not as efficient as the best monolithic programming; the question you must resolve is whether or not you are getting the best programming now. (Incidentally, most IBM programmers are probably doing modular programming to some extent because the linkage facilities provided in the standard software make it easy and natural.)

Hoskyns Systems Research Inc. takes the attitude that modular COBOL programming is a good thing. To support this view, Testmaster was developed to permit testing of the subprogram building blocks independently, without having to worry about input/output.

To gain a perspective for what Testmaster can do, first imagine inserting print statements throughout a program for printing out intermediate results so that each step of the processing procedure can be checked. Next, imagine the capability for collecting these output statements into a separate module so they can be easily removed when it's time to put the production version of the program together. Add the capability to assign values to data items (and thus remove the need for input/output operations), and you have in a nutshell what Testmaster can do.

On the surface, these capabilities seem rather simple to implement. And indeed, a limited version of this capability wouldn't be hard to crank out in your own shop for each programming task. But Testmaster provides a generalized set of facilities ready to go.

The price may seem steep ($7,500 or $12,000, depending on the version); and in view of the fact that the package only provides for testing procedural coding, not file and record handling, it *is* steep. Nonetheless, Testmaster provides a rational way to test your COBOL program modules modularly, and at a price that is lower than the cost of writing it yourself.

The chief advantage of testing procedural coding with Testmaster is that it conveniently permits setting up multiple tests of a module using a range of values for variables. If complete programs were tested in this manner, the resulting output would be very difficult to interpret due to its magnitude. The output of the tests is geared to easy reading; variables that have changed in value are printed out using the actual data names as

Designed to facilitate independent testing of IBM System/360 or 370 COBOL program modules, Testmaster replaces input/output operations with direct assignment of values to input variables. By breaking down testing operations to module size, Testmaster allows comprehensive testing while keeping the test output to a manageable size.

CHARACTERISTICS

SUPPLIER: Hoskyns Systems Research, Inc., 600 Third Avenue, New York, New York 10016. Telephone (212) 687-8090. European marketing is handled by Hoskyns Systems, Ltd., 91 Faringdon Road, London, EC 4, England.

BASIC FUNCTION: To test COBOL procedural subprograms individually without the use of input or output files. The package runs on an IBM System/360 or 370 under DOS or OS (or their VS counterparts).

OPERATION: Testmaster consists of three parts: a language for stating testing operations, a compiler for translating these statements into executable code, and a testing supervisor that operates at execute time to control execution/testing of the COBOL subprograms. Effectively, a module is generated that acts as the main-line program and I/O subprograms to permit passing data values to and from the tested subprogram without worrying about I/O coding or program coding not yet completed. Calls to subprograms not included in a testing run can be simulated by specifying the data values returned to the calling subprogram.

TEST LANGUAGE: The structure and syntax of the language simulates that of COBOL itself. There are three divisions: PARAMETER, TEST, and SIMULATION..

The PARAMETER Division corresponds exactly with the LINKAGE Section of the DATA Division of an IBM COBOL subprogram. In fact, a COPY option is provided that allows the LINKAGE Section to be picked up directly from a cataloged subprogram. Note that VALUE statements and REPORT PICTURE specifications are not permitted.

The TEST Division consists of a series of value assignment statements, simulation statements, test execution statements, and option statements.

Values are assigned to data items identified in the PARAMETER Division through a series of "data-name VALUE decimal-value" statements, one per data name. Assignment of values to a group variable is simple. Multiple values can be assigned to any data name to cause a series of iterative tests.

SIMULATE statements are required to reference a paragraph in the SIMULATION Division when calls to subprograms not included in the testing run are imbedded in the subprograms being tested.

Testmaster/COBOL
Hoskyns Systems Research Inc.

identifiers. Each test is numbered, and the input values used for each test can be printed if desired. For extended tests, printing the input assignments is probably a good idea; otherwise, it can become very difficult to identify which set of input values gave which results.

Hoskyns Systems Research Inc. is a U.S. branch of an English firm, Hoskyns Group Limited. This firm has several subsidiaries devoted to various aspects of the information processing industry. Hoskyns Systems Research Inc. provides services for medium-to-large IBM System/360 or 370 users oriented toward increasing system and programming efficiency. Other Hoskyns software products currently available are Linkmaster, designed to facilitate modular programming, and Tablemaster, a decision-table processor. Only Testmaster is being heavily marketed at this time. □

The TEST . . . USING . . . is identical to the COBOL CALL . . . USING . . . statement. It specifies the first subprogram to be entered in the particular test. Program flow to other subprograms is controlled by the subprogram coding itself.

Each TEST statement represents one test or one related series of tests. Multiple TEST statements can be included in one test run specifying the same subprogram with different sets of data values or different subprograms to be tested.

If multiple values are assigned to a variable or multiple sets of values to a group variable, each combination of values creates one test case. Thus, if 4 values were assigned to one variable and 5 values to another, then 20 different tests would be executed.

The SIMULATION Division permits assigning values to data names which are the expected results to be returned from a subprogram called from a subprogram being tested. Multiple sets of values can be assigned, permitting iterative tests to be made in the same run. The simulation facility allows subprograms to be tested in isolation or allows a subprogram to be tested prior to the completion of programming of supporting subprograms.

PROCESSING OPTIONS AND TEST OUTPUT: The basic output for a successfully executed test is the value of any variable that was identified in the TEST paragraph and that has changed in value from the original assignment. These results are printed in clear alphanumeric form. Each test is identified with a number, sequentially assigned. In addition, the running time for the test execution is printed. If the test is not completed because of a program check or because execution time exceeded a preset value, diagnostic information is printed. This information consists of the Program Status Words (Registers 6 and 7) along with the contents of Registers 0 through 5 and 9 through 13. In addition, the contents of the data area assigned according to the PARAMETER Division is printed. The data area is printed in hexadecimal and alphanumeric form; the other items are printed in hexadecimal.

Several options extend the operating capability of Testmaster. Ordinarily, if a program check is encountered during one of the tests specified in a TEST paragraph, that paragraph is abandoned following output of diagnostic information; optionally, execution of other tests in the paragraph can be forced. Through the LIST option, input parameters to each test can be printed before the results. In any kind of extended testing, this is a valuable option, both to verify input values and to serve as a place finder. If desired, results can be passed from one test to another in the same run. This is helpful where intermediate results are difficult to predict. If you like to look at core dumps, the full contents of the partition can be output following a program check.

The SEARCH option allows variables defined in the PARAMETER Division but not called out in the TEST paragraph to be monitored for changes in value; this allows checking for illegal overwriting. Under OS, a programmer can mask out any or all of the arithmetic program checks and prevent abandonment of a test due to this cause. A COPY option permits including Testmaster source input from cataloged data sets; the most frequent use of this option is to pull the PARAMETER Division directly from the LINKAGE Section of the subprogram being tested. Descriptive comments can also be inserted anywhere in the test data deck.

Intelligent use of the basic capabilities and options allows the creation of a printed test output report that is very easy to analyze.

PERFORMANCE: Hoskyns states that the Testmaster compiler works at about 4 statements per second on an IBM Model 40 when output is to disk, and that the test execution is totally bound by the device used to record output.

HARDWARE REQUIREMENTS: Testmaster will operate on any IBM System/360 or 370, Model 25 or larger; it will run under DOS or OS (PCP, MFT, or MVT), or their VS counterparts. The minimum partition size is 24K bytes. This allows about 10K for the subprograms being tested; additional main memory can be employed to test larger subprograms. Library space required to hold Testmaster is about nine 2314 tracks or seventeen 2311 tracks. Testmaster uses only system files; no other input or output is required.

SOFTWARE REQUIREMENTS: Testmaster is supplied in object code form, and system generation requirements are confined to establishing the linkage editing to the appropriate library.

PRICING: Testmaster is available for a 2-year license term only, with one-time payments of $7,500 (DOS) or $12,000 (OS). Time payments can be negotiated. Subsequent 2-year contract renewals are $3,000 and $1,875 for OS and DOS, respectively. One day of training and free maintenance are included.

INITIAL DELIVERY: 1969.

CURRENT USERS: About 75 worldwide. About 30 of these users are located in the U.S. ■

IMS
IBM Corporation

MANAGEMENT SUMMARY

IBM's Information Management System is the patriarch of a variety of data base management systems available from IBM and other sources in the computer industry. Although many of the non-IBM systems have enjoyed considerable success, and some have demonstrated throughput and response-time performance superior to that of IMS-2, the currently-available real memory version, none has the range of flexibility and established interfaces to ancillary software products that the IMS Program Products and related subsets have. The flexibility of IMS leads directly to much of the confusion that currently surrounds the product. IMS is a complex system, with many subtle attributes capable of causing considerable impact upon organizations implementing it.

At the center of this potential disturbance is the apparently innocuous data base: a nonredundant collection of interrelated data items processable by one or more applications. But to quote Keane Associates, Inc., a New England firm specializing in independent consulting for IMS/360 implementation: "Installing the IMS centralized data base management system is like opening Pandora's Box as far as corporate policies and organization are concerned." This management aspect of data base implementation is discussed in depth below.

Briefly described in the most simplistic way that still retains accuracy, IMS is concerned with the efficient organization and structure of data on physical direct-access devices. IMS provides a means for physical-level access to the data, and sets up an interface between the user's application program and the operating system's data management and communications management facilities.

The fundamental functional concept underlying IMS's technique for data base management is fairly simple: The user employs utility programs to describe the structure of the system from two viewpoints; stored data structure as seen by the system, and logical data structure as seen by an application. Stored data is described once only, but many descriptions of the logical data can exist. The logical data base descriptions are external to the application programs and exist as stored data that the system references when processing access requests to the data base from application programs. Thus, IMS maps logical data descriptions into the physical data descriptions it alone can use, using logical data names supplied by the application. It also determines an access strategy and performs the requested function against the stored data. The entire range of system attributes and functions rely upon this basic concept.

IMS was originally developed as a joint venture between IBM and the North American Rockwell Company in the mid-1960's. At that time, the classic concept of special-purpose file processing programs with one or more individual files for each program was the accepted way of doing business. In an effort to centralize file maintenance, eliminate the storing and maintenance of redundant data, ▷

IMS, IBM's principal data base management system, is now available in several versions that permit operation in either real or virtual storage mode and in either batch or on-line environments. This report describes and analyzes IMS-2, IMS/VS, and two IMS subsets for use on smaller systems: DL/1 DOS/VS and VANDL-1.

CHARACTERISTICS

SUPPLIER: IBM Corporation, 1133 Westchester Avenue, White Plains, New York 10604. Telephone (914) 696-1900.

BASIC FUNCTION: Provides facilities for generation and accessing of a data base that permits automatic cross-referencing among data records. IMS/VS does so under the OS/VS1 or OS/VS2 operating system, and it is the basic, or data base (DB) portion of IMS/VS that provides these facilities. IMS-2 is analogous to IMS/VS and operates under OS/MFT or OS/MVT or, with fixed pages, under OS/VS1 or OS/VS2. IMS/VS and IMS-2 offer on-line message processing in their data base/data communications (DB/DC) optional forms when the Data Communications Feature is used. With this feature, both offer high-level on-line inquiry with IQF (Interactive Query Facility) and batch inquiry using GIS (General Information System). A data language, whose function is to replace user I/O coding with simpler commands to IMS, is provided.

Providing the same basic data base facilities under DOS/VS is DL/1 DOS/VS; however, its only available data communications extension is linkage to CICS/DOS/VS (Report 70E-491-02). IMS/VS can likewise link to CICS/VS, and IMS-2 to CICS-OS Standard, but they must then forego their own data communications features, and, more significantly, IQF.

Providing DB facilities under DOS or DOS/VS (in virtual mode) is VANDL-1, a programming RPQ from IBM that is currently available. VANDL-1 can link to CICS (Entry level or DOS Standard) to provide teleprocessing facilities.

All of the foregoing programs are written in Assembly language and offer their DB facilities to users of COBOL, PL/1, and Assembly language. VANDL-1 is an upward-compatible subset of DL/1 DOS/VS, which is itself an upward-compatible subset of IMS/VS.

Throughout the Characteristics description that follows, readers are asked to recall the relationships between IMS/VS, IMS-2, the IMS Data Communication Features, IQF, DL/1 DOS/VS, VANDL-1, and CICS. The accompanying table clarifies these relationships. Also, unless specifically stated, "IMS" will be taken to mean the data base facilities of four programs (IMS/VS, IMS-2, DL/1 DOS/VS, and VANDL-1), or, alternatively, the combined facilities of DB/DC IMS/VS and IMS-2.

OPERATION: IMS operation requires that the user perform the following steps (presented in the typical development sequence, although considerable overlapping normally takes place):

- The applications must be defined in terms of functional requirements and types of data to be processed. Individual applications programs must then be written in COBOL, PL/1, or Assembly language to perform these applications.
- All of the individual data requirements for each application must be coordinated into an overall data base contents requirement. This task is most appropriately handled by an individual serving as a Data Base ➤

isolate individual programs from each other and from the data (data independence), ease the often complex handling of variable-length records, and provide a framework for applications program development, NAR and IBM produced IMS-1. The system was subsequently released as a Program Product with full IBM support in September of 1968. Enhancements have resulted in the more recent IMS-2 product.

A serendipitous benefit of the centralized data base approach is what IBM promotes as "data integrity." Data integrity in IMS is merely the blessing of having the management and planning staffs all "reading off the same scorecard"; i.e., one source of figures is used by everyone. Surprisingly, to the uninitiated, this basic coordination is very difficult to achieve in most large corporations.

Although IMS passed through a number of intermediate releases before arriving at Version 2, the major distinctions of IMS-2 from IMS-1 can be briefly summarized as follows:

- Two hierarchical direct data organizations–HDAM and HIDAM–were added to those already available.
- The Interactive Query Facility (IQF) was made available. (Actual release of the facility did not occur simultaneously with IMS-2, but IQF can be used only with IMS-2.)
- IMS-2 was packaged differently from IMS-1, with data communications handled as an optional feature.
- Performance of IMS-2 is significantly improved over that of IMS-1.

Early in 1973 the next step in IMS development occurred when IBM announced IMS/VS (IMS for virtual storage) for availability under OS/VS1 and OS/VS2 Release 1 in February 1974 and under OS/VS2 with multiprocessing support in July 1974. New functional capabilities were added and operational flexibility was increased. The IMS/VS improvements can be summarized as follows:

- Data base changes can be "backed out" in the event of an abnormal termination of a message or batch-message program.
- Batch checkpoint/restart facilities are added for batch and batch-message processing.
- New program isolation allows the same segment types within data bases to be concurrently updated by multiple-message and batch-message programs.
- Improved multiple indexing allows access to root and dependent segments within a data base, allowing any field within most segments within a data base on disk storage to be indexed.
- VSAM support is provided for disk storage of data; VSAM data sets and single-segment HISAM data bases will share a common stored record format.

Administrator. Based upon the processing logic of the various applications programs, the administrator then selects one of the hierarchical IMS structures for the physical storage of the data.

- **Communications and other on-line retrieval requirements must be evaluated, both in terms of functional (application) needs and in terms of the network design criteria (terminal types and quantities) that may be present. If communication-type requirements or interactive query requirements are present, a choice must be made of two data communications monitors (the Teleprocessing Option of IMS and CICS) and/or two high-level inquiry methods (IQF or GIS). Note, however, that CICS and IQF cannot both be selected: IQF runs only with the IMS Data Communications feature.**
- **At this point, the Data Base Administrator must put together the IMS modules required to support his installation. The complexity of this task can be considerable, particularly if data communications is involved. The IMS modules include the resident nucleus, control facility system tasks, BTAM, GAM, OSAM, or ISAM re-entrant modules, BTAM or BSAM device support, terminal handlers, control blocks, buffer pools, and certain OS requirements.**
- **In conjunction with the selection of an appropriate data storage structure and data communications or inquiry method, the Data Base Administrator then sets out to create one or more "logical" data bases that redefine the actual physical data base. This process establishes logical relationships in the form of pointers and/or chains that facilitate user-program access to the data. Although wide flexibility is provided within IMS to create valid logical data bases that tie in various data elements or segments, inappropriate design at this point can result in severe overhead problems.**
- **The actual data base is created by a special-purpose off-line IMS utility program.**
- **Finally, the user is ready to incorporate DL/1 access language into his own programs and process his IMS data base applications. This step basically consists of replacing the standard I/O syntax in each application program with CALL statements containing parameters pertinent to IMS.**

The above sequence of operational steps describes a classical, theoretical implementation of the full-scale IMS management information system environment, including data communications capability. In common practice, however, most IMS applications evolve incrementally, with only a basic core of applications and a partially developed data base to start with. Also, any number of responsibilities assigned to the Data Base Administrator above can be handled by systems programmers at the user's option.

The data base environment within IMS provides for a "separation" between the data and each user program. This concept of data independence implies that changes to the data base–such as the inclusion of new fields, changes in record length or description, and physical reorganization into new structures or different device types–need not be accompanied by corresponding changes in what is likely to be a large number of individual application programs that access the data base.

In order to achieve data independence, a common symbolic program linkage and data base description is used that supports five primary types of data-base access or I/O operations:

- **Retrieve a unique segment (GET UNIQUE).**
- **Retrieve the next sequential segment (GET NEXT).**
- **Replace the data in an existing segment (REPLACE).**

IMS
IBM Corporation

Relationships Between the IMS Versions and Operating Systems and Communications Programs

Program ID	Operating System (and mode)				Interface for:
	DOS	DOS/VS	OS	OS/VS	
IMS Version 2 5734-XX6	NA	NA	yes	real	DC feature or CICS Version 1 or 2
IMS/VS 5740-XX2	NA	NA	NA	virtual	DC feature or CICS/OS/VS (for CICS V1 or V2 in real mode)
DL/1 DOS/VS 5746-XX1	NA	virtual	NA	NA	CICS/DOS/VS (or CICS DOS or CICS DOSE in real mode)
VANDL-1 5799-AEY	yes	virtual	NA	NA	CICS DOS or CICS DOSE (real mode)

- The IMS control region and dependent regions can be executed in virtual mode.
- Variable-length data segments can be dynamically changed.
- Support for remote computers includes the System/3 Model 10 and System/7, both on nonswitched multipoint lines.

IMS/VS communications supports all devices supported under the communications feature of IMS-2 except for the 2260, 2265, and 1030 terminals. Also, the IMS/VS control region must operate in real mode when the 7770 Model 3 Audio Response Unit is used.

DOS and DOS/VS users are not left out of the IMS picture. In October 1972, IBM announced DL/1 DOS/VS for delivery in November 1973. This program—not to be confused with DL/1 (Data Language/1), the command language used with IMS—provides the DOS/VS user with data base capabilities that are a compatible subset of those found in the DB portion of IMS/VS. For the DOS/VS user interested in a DB/DC system, DL/1 DOS/VS will offer an interface to CICS/DOS/VS (Customer Information Control System, report 70E-491-02).

Finally, the DOS user who can't wait until the end of 1973 for a data base system can use VANDL-1 (Vancouver Data Language-1), a currently available upward-compatible subset of DL/1 DOS/VS. VANDL-1 runs under either DOS or in a virtual mode under DOS/VS; it can be extended to provide DC facilities, using CICS, as well. It is a programming RPQ that IBM encourages for interim use. The relationships of the IBM IMS systems to the various operating systems, and the interfaces each can provide for growth to a DB/DC system, are outlined in the accompanying table.

THE STRUCTURE OF IMS

As it currently stands, IMS-2 and IMS/VS each come in two basic flavors: Data Base (DB) system only, and Data Base system with Data Communications feature (DB/DC). While the DB system handles input job streams by batch

- Delete the data in an existing segment (DELETE).
- Insert a new segment (INSERT).

These operations can be performed on one or more hierarchically related segments or data elements in a logical data base record. Each of the operations is invoked by a Data Language/1 (DL/1) command. (The appropriate DL/1 command is indicated in parentheses above.) DL/1 works with a data base description (DBD) produced by the off-line IMS utility program that creates the data base. The DBD provides the "mapping" from the logical structure of the data base (as viewed by the application program) to the physical structure of the data base (as kept on a storage device by OS). The logical data structure of IMS is based upon segments: an IMS data base consists of 1 to n data base records; a data base record consists of 1 to n segments of up to 255 segment types and up to 15 segment levels. There is one root segment per data base record and 0 to n occurrences of dependent segments per parent.

Four primary physical data organizations are provided in IMS:

- Hierarchical Sequential Access Method (HSAM)—an extension of basic serial tape and disk file processing (SAM). This method offers limited data independence and no interrelatability of the data base through "pointers." In order to insert a data base record, the data base must be copied up to that point, the new record written, and the rest of the data base copied. Each record is physically present in the serial order in which it logically appears in the data base.
- Hierarchical Indexed Sequential Access Method (HISAM)—provides an imbedded hierarchy of ISAM-like data sets that are related by sets of symbolic pointers or keys. The distinguishing aspect of HISAM (or HSAM), as opposed to the hierarchical direct methods described below, is that all segments in a physical data base record are "related by physical juxtaposition." For HISAM, this means that a direct-access relationship is established between all of the physical blocks containing the segments belonging to a given data base record. An Overflow Sequential Access Method (OSAM) physically contains the segments that cannot fit in the HISAM logical record. OSAM is based upon standard OS physical data sets and combines the best features of both BSAM and BDAM: concurrent sequential and direct access for retrieval, in-place updating as well as addition at the end of a data set, data set "end" recognition, secondary extent definition for data sets, etc. IMS/VS also provides support for certain VSAM data sets on disk; that is, VSAM

IMS
IBM Corporation

scheduling, the DB/DC system is transaction-oriented and schedules work based upon input messages.

In the DB batch environment, all necessary IMS modules are combined with each user's program in their individual partitions. The DB/DC system, on the other hand, concurrently supports both batch and on-line applications and provides an independent control partition or region for the IMS modules that is separate from the message processing/batch applications processing partitions or regions, each of which contains only the user's application program.

IMS is a "host language" system, which means that no direct application code is provided by IBM. The user writes his own applications programs in COBOL, PL/1, or Assembly Language, for which interface modules are provided. These application programs access the data base through Data Language/1 (DL/1) commands that replace what would otherwise be complex procedural I/O coding and the file descriptions present in applications programs. Thus, most applications programs written for an IMS environment tend to be smaller than similar non-IMS programs that perform the same function. For COBOL, as an example, the source language programs tend to be one-fourth to one-third smaller, and the corresponding object programs up to one-fourth shorter, with a consequent reduction in application program development effort.

Each IMS "physical" data base is organized in one of four hierarchical or tree-type structures. The user views the physical data through a "logical" data base description that allows sophisticated security precautions and data relationships to be expressed. All user interfacing is done at the "segment" level, which is the basic IMS data element. The logical data base represents one of IMS's best assets, as it facilitates data independence and simplifies applications program logic. Data stored in multiple separate physical data bases can be treated as if it were logically part of one data base. Because of the logical data base concept and DL/1, existing IMS applications programs can be insensitive to the physical reorganization of data on the storage devices, the addition of new applications or data, changes or new developments to OS or VS access methods, and changes in device types or terminal devices.

A full complement of IMS utility programs is provided to describe the data base structure, create the data base, reorganize the data base, recover and reconstruct data (checkpoint/restart capability), specify security control, and analyze the system workload.

One interesting aspect of IMS is its potential for management information system (MIS) development. MIS and IMS get confused with each other for obvious reasons; but an MIS system can be thought of as a collection of management information application programs depending upon an IMS data base. The extent to which these application programs satisfy management's information needs determines the extent to which the MIS can be considered successful. Often the installation of IMS is the preliminary step in the implementation of a full-scale MIS.

data sets and single-segment HISAM data bases will share a common stored record format. HISAM does not yield particularly good results in an on-line environment.

- Hierarchical Direct Access Method (HDAM)–stores data in a physical tree structure with all segments in a physical data base record related by direct addresses. Segments can be interrelated to each other as physical twins (multiple occurrences of the same segment type under a given parent), physical parents (segment immediately above), or physical children (first and last occurrence of each segment type immediately subordinate) through chains of pointers. HDAM uses OSAM as a base for data storage and provides very effective access to dependent segments–especially in teleprocessing environments–at some overhead cost in terms of data base size.

- Hierarchical Indexed Direct Access Method (HIDAM) –provides an ISAM index to data physically stored in OSAM format. The ISAM index contains the key of a root segment and a direct address to the root segment, while the actual storage of data is done in OSAM data sets. Because the data base index and the actual data base are kept on two separate data sets, reorganization of the index separately from the data is facilitated. HIDAM is the most generally appropriate and most often used data organization method for IMS applications.

In addition to the above data structures and access methods, the basic batch-oriented version of IMS (also called "DL/1 Data Base" or the DB system), can be augmented with data communications capability to produce a transaction-driven system. The DB system is prerequisite to the DC Feature ("IMS teleprocessing"). The resulting full-scale IMS is known as the DB/DC system, and can handle both batch and on-line operations concurrently. A DB/DC system can have a wide variety of physical terminals (more than a dozen types), each of which can have one or more logical or symbolic names. Individual security parameters can be associated with each terminal's logical name. Among other facilities in a DB/DC system are the following:

- Master Terminal for network control, message scheduling, and interrogating or altering the IMS processing functions.
- Input message processing for transactions, terminal-to-terminal messages, or system interface messages.
- System log with a queue of all I/O messages on disc (used for restart).
- Standardized message editing with the ability to add user-defined editing routines.
- Conversational processing capability for terminal access to applications programs.
- Video Terminal Paging feature.
- System security both by terminal and by password.
- Terminal Test Mode to debug on-line IMS applications.
- Message scheduling, including up to 15 priority levels, message class, and dynamic message re-prioritization parameters.

As an alternative to the IMS-Teleprocessing option, a DB/DC system can be put together using the Customer Information Control System. CICS generally provides similar functional capabilities with lower overhead in some environments. CICS was designed for relatively short program modules of about 2K to 6K bytes, while the IMS Teleprocessing option is better suited to 20K-byte modules

IMS
IBM Corporation

USER REACTIONS

Users of IMS-2 contacted by Datapro report a variety of reactions. One of the most pervasive was that local IBM representatives do not seem to understand IMS-2 thoroughly, and this sentiment has been echoed within IBM itself.

Another finding among users is that one of the least understood and most sensitive aspects of IMS-2 usage is that of selecting an appropriate hierarchical data organization. By survey, Datapro observed that fewer than 5% of the IMS-2 users have selected HSAM. Many of these had been IMS-1 users or are installations that converted from serial tape processing; this group is generally receiving the least of IMS's potential benefits. Of the remainder, about 10% are working strictly with either HISAM or HDAM; about 50% are exclusively using HIDAM; and the last 25% are using a combination of HISAM and HIDAM for their data bases. While it is generally agreed that HIDAM is perhaps the most generally effective of the IMS-2 data base organizations, it is by no means clear what the best organization is for a particular user's situation. Indeed, this is one way in which the TOTAL data base management system (Report 70E-132-01) differs from IMS-2: fewer data organization choices, using an overall "best fit" approach.

Some customers of IBM's IMS, as have CICS users, have found it wise to obtain outside consulting help. One firm experienced with both IMS and CICS is Automated Concepts Incorporated, New York, N.Y.

CORPORATE IMPLICATIONS

Perhaps more important than any of the computer-related considerations of a data base management system are the overall corporate implications. As an example, consider the case where different departments access the same data in the data base. Which department is to be responsible for gathering the update information, editing it, and entering it as maintenance transactions to the IMS data base? Further, if the department currently responsible for the legwork and procedures involved in updating some of the common elements in the data base receives a new charter eliminating that department's need to access that data, can it drop the update responsibility? Who would then inherit the update manpower and/or responsibility? Since there may not be a change in the net overall corporate responsibilities, it would seem that no net manpower change should be effected; but individual departments can be dramatically impacted. And, of course, there is a matter of security. Who gets to see the payroll information, since it may possibly be on the same physical storage device as the corporate production data?

There are direct solutions to some of these problems, but not to others. Top corporate involvement in the design and subsequent care and feeding of the data base is a key element to success with IMS. One approach that has gained almost complete acceptance among IMS-2 users is to establish a Data Base Administrator function. This position calls for a sort of "referee" to control the contents of the data base, take charge of enforcing

or larger. The CICS/IMS interface has been available since early 1972.

Also available for full DB/DC IMS systems is the Interactive Query Facility (IQF). IQF is a basic query language that offers the capability for on-line retrieval and display of data in an IMS data base. IQF consists of retrieval phrases that define, delete, list, sort, count, total, limit, and query the data base; the "when" qualifier to establish criteria for data selection; and a basic complement of relational (EQ, NE, LT, GT, LE, GE), logical (AND, OR), and arithmetic (+, -, /, *) operators. Other handy features of IQF include null words (e.g., THE, OF, FOR) set up by "define" phrases, literal and numeric constants, segment synonyms, etc.

IQF is handled like a standard applications program under IMS. A batch utility program is used to create and maintain the following data bases associated with the use of IQF:

- System Data Base for rapid resolution of data base field names so that each IQF query does not have to specify individual segment names.
- Phrase Data Base that contains predefined phrases and null words set up by the user to tailor IQF for a specific set of applications.
- Two IQF Index (QINDEX) Data Bases that index the non-key data in the IMS data base(s). Typically a "large" and "small" key-field data base is created. The QUINDEX data bases are also used to streamline on-line retrieval requests.

The IQF utility is run when putting the IMS system together and for subsequent index creation or modification. IQF can be used only with the IMS DC Feature. It is not supported by CICS. IQF can be used in conjunction with the more powerful query capability of the Generalized Information System (GIS/2)—a full-scale information system—although no direct relationship between IQF and GIS/2 exits.

GIS/2 with the DL/1 Query Support Feature can be used to produce tailored processing modules that permit batch operations upon IMS data bases as well as a variety of other data file types. GIS/2 is actually a sort of "super RPG" that accepts report format and query selection criteria as input and produces an object deck plus Job Control Language (JCL) as output. These GIS/2 object modules can be executed under either the DB or DB/DC environment through local control as batch programs only. Full GIS/2 query capabilities (except LIST RECORD and HOLD RECORD) are supported for IMS data bases.

A Bill Processor Bridge System is also available that converts Bill of Material Processor (BMP 360A-ME-06X) and Data Base Organization and Maintenance Processor (DBOMP 5736-XX4) files into IMS-2 data bases.

HARDWARE/SOFTWARE REQUIREMENTS: IMS/VS runs on System/370 computers under OS/VS1 or OS/VS2. The minimum machine requirements are a System/370 Model 145 or larger CPU with main storage of 384K (OS/VS1), 512K (OS/VS2, Release 1), or 768K (OS/VS2, Release 2) for a data base (DB) only system. With the Data Communications feature added (DB/DC system), these systems would require respectively, 512K, 768K, or 1,024K bytes of main storage. Within those storage sizes, the practical minimum address spaces IBM recommends are consistent: 90K plus the size of the user's application for a DB system and 200K for a DB/DC system. Estimates on how much of the address space must be fixed in main storage are presently unobtainable. Also required are an OS/VS1 or OS/VS2 system console, one magnetic tape unit, and, for a DB/DC system master terminal, either a 2740 Model 1 or 1050 Data Communications System with a 1052 Printer-Keyboard. Finally, a DB system needs 125 cylinders of 2314-type or 83 cylinders of 3330-type disk space, and a DB/DC system requires 225 cylinders on a 2314-type or 158 on a 3330-type disk.

IMS
IBM Corporation

▷ adherence to system standards, and steer potential intraorganizational conflicts over the data base to the proper authorities for resolution. This staff function must be neutral to all parties and technically qualified to give advice to management. The Administrator has many responsibilities akin to those of the system programmers, and one of the most important of these is "tuning" the system for maximum performance. This can involve reorganization, creation, or modification of logical data base descriptions, etc.

An important trade-off that must be made by the designers of the data base and subsequently reviewed by the Data Base Administrator during the tuning process is that of DL/1 programming complexity and data organization method (HIDAM, HSAM, etc.) against processing time, size of disk storage requirements for the data base, and main memory requirements for the IMS system modules. Adjustments will be called for as the data base develops, and tuning should be an on-going process with periodic reviews.

Finally, one piece of advice about IMS: don't underestimate the magnitude of the commitment necessary to implement the system. The cost is considerable in terms of computer hardware, manpower, time, and applications development effort, not to mention the direct rental charge of the IMS software. It should be pointed out, however, that a major percentage of the cost of implementing IMS is a front-end expense for development work, and this expense will be roughly related to the complexity of the application itself. A successfully implemented IMS system will produce a much more efficient data processing shop and yield significant long-term benefits for the entire corporation.

Any prospective OS data base user owes it to himself to investigate IMS-2 and/or IMS/VS carefully and compare it to the competitive products available from a number of sources. Likewise, DOS/VS (and DOS) users should examine the capabilities of VANDL-1, with the DOS/VS user keeping his eye on possible growth to DL/1 DOS/VS and then either to OS/VS1 and IMS/VS or to DL/1 DOS/VS and interfaced CICS/DOS/VS. Central data base management systems with communications capability are rapidly gaining acceptance, and reliable data management systems built around centralized data bases hold too much potential to be ignored by any large corporation much longer. □

➤ IMS-2 runs on System/360 or 370 computers under OS/MFT or MVT or their VS1 or VS2 counterparts. Under VS operating systems, it runs in fixed-page (real memory, or V=R) mode. A System/360 Model 40 or System/370 Model 145 or larger CPU is required, with main storage of 128K (OS/MFT), 256K (OS/MVT), or 512K (Model 65MP-multiprocessor) for a DB-only system, or 256K (OS/MFT) or 512K (OS/MVT or 65MP) for a DB/DC system. IBM states the minimum practical OS/MVT partition or MVT region sizes as 76K (MFT) or 81K (MVT or MP65) plus the size of the user's program for a DB-only system, and 175K (MFT) or 171K (MVT or MP65) for a DB/DC system. System/360 Model 67 must run in the Model 65 mode to use IMS-2. An operating system console and one magnetic tape unit for a DB system or two for a DB/DC system and 115 cylinders of 2314-type disk space for DB or 185 for DB/DC are also required. A DB/DC system also has the master terminal requirements mentioned above for IMS/VS DB/DC.

In addition to the indicated requirements, about 50K bytes are used to support IQF in an independent message processing region. The batch IQF Utility, however, requires at least 140K bytes of resident main memory. The IMS batch utilities each have various main memory requirements associated with them. If CICS is used instead of the IMS Data Communications (DC) feature for on-line processing, the CICS minimum resident main memory requirement is about 65K bytes. The optional use of GIS/2 imposes a minimum main memory application partition requirement of about 120K bytes.

DL/1 DOS/VS runs under DOS/VS on a System/370 Model 125 to 158 CPU with at least 98K bytes of main storage. Typically, a System/370 Model 145 would be used, however, as an estimate, about 60K of main storage should be allocated for basic operations (1 batch program), and about another 30K if teleprocessing using CICS/DOS is used. In addition to the DOS/VS requirements, additional 2314- or 3330-type disk storage can be used for the data base. In addition, two magnetic tape units, a card read punch, and a 1403-N1 or equivalent printer are required. The minimum linked DL/1 DOS/VS-CICS/VS system requires a System/370 Model 135 CPU with 147K bytes of main storage.

VANDL-1 runs on System/360 or 370 systems under DOS or DOS/VS. Under DOS (Release 26), it requires at least a 14K partition for the VANDL-1 module plus 8-10K additional in the user's batch area for two data base files or 8K for a CICS partition; under DOS/VS the same address space, but possibly less main storage, is required. VANDL-1 is known to release CICS very quickly.

PRICING: IMS-2, IMS/VS, and DL/1 DOS/VS and their supporting programs and features are standard IBM Program Products with full, centralized IBM Class A programming support. VANDL-1 is an IBM programming RPQ with Class B support.

Program	Number	Monthly License
IMS/VS	5740-XX2	$700
IMS/VS DC Feature	6001 or 6002	850
IQF for IMS/VS (DC) or IMS-2	6068, 6069, or 6070	300
IMS-2	5734-XX6	550
IMS-2 DC Feature	6022-6024	625
GIS-2	5734-XX1	450*
DL/1 DOS/VS	5746-XX1	300
VANDL-1	5799-AEY	350

*Numerous optional features available for GIS-2 can raise its total cost to as much as $1500 per month.

INITIAL DELIVERY: IMS has been delivered in several releases, and other releases are scheduled as follows:

Release Number	Distinguishing Characteristics	Date
2.0	Basic IMS-2	March 1971
2.1	3330 support	November 1971
2.2	Improved data communication service	August 1972
2.3	3270, Virtual Storage support	November 1972
1.0	IMS/VS for OS/VS1 and OS/VS2, Release 1	February 1974
2.0	IMS/VS for OS/VS2, Release 2	August 1974
1.0	DL/1 DOS/VS	December 1973
1.0	VANDL-1	November 1972

CURRENT USERS: More than 125 IMS-1 and 200 IMS-2 installations. (Note that these figures are based upon current market research information; IBM does not release official installation figures.) ■

CICS
IBM Corporation

MANAGEMENT SUMMARY

CICS (Customer Information Control System) is one of IBM's three primary data base/data communications (DB/DC) Program Products. (The others are IMS-2, Report 70E-491-01, and the Generalized Information System, Version 2). CICS shares the distinction of being among the first of IBM's Program Products unveiled after the unbundling announcement in 1969. Its name (occasionally pronounced "kicks") is actually an anachronism carried over from its original development by the public utilities industry group in IBM.

CICS is designed to simplify the on-line communications interface between user-written applications programs in COBOL, PL/1, or Assembler language and IBM's operating systems. Thus, CICS consists of a system of program modules providing services that largely free the applications programmer not only from concern with the communications environment, but also from concern with the operating system itself. As a comprehensive, general-purpose data communications monitor. CICS has the widest range of established data base interfaces and ancillary software product support of any such system in the industry.

CICS also provides a limited amount of inherent data base (DB) management capability. The native CICS DB component, however, unlike Data Language/1 (DL/1), its more advanced, top-of-the-line IMS-2 DB counterpart, is not available separately, but only in combination with the CICS data communications facilities as a joint DB/DC system.

The present CICS family of products is a wholly IBM-sponsored outgrowth of what was originally a joint development effort between IBM and the Commonwealth Edison Company of Chicago – a major public utility firm – in the mid-1960's for a Customer Information System (CIS). Version 1 of CICS was subsequently announced as a generalized Type II IBM program supporting Assembler language under OS in April 1968, but it was first delivered as a Program Product following unbundling about a year later (Actually, several dozen CICS pilot customers received no-charge versions of the then-Type II program just prior to unbundling.)

The initial release of CICS was designed for the customer inquiry environment of the public utilities industry, but was really a full-function data communications monitor easily made applicable to other industries as well. Subsequent DOS releases of CICS near the end of 1971, together with on-going enhancements, have greatly enlarged the industry applicability of CICS and extended its potential usefulness to virtually every type of on-line application.

Early in 1973, IBM announced DOS/VS and OS/VS versions of CICS which are upward-compatible with their predecessors. Real-memory versions of CICS will run in a virtual mode under control of VS operating systems, but ▷

CICS, one of IBM's three major data base/data communications Program Products, is the world's most widely accepted data communications monitor. It comes in a DOS Entry-level version, in full-scale DOS and OS Standard versions, and in new DOS/VS and OS/VS versions.

CHARACTERISTICS

SUPPLIER: IBM Corporation, 1133 Westchester Avenue, White Plains, New York 10604. Telephone (914) 696-1900.

BASIC FUNCTION: CICS is a general-purpose data communications monitor that operates in a single partition or region of an IBM System/360 or 370 under DOS or OS (or their VS counterparts) to control multiple on-line user terminals and applications. By consolidating the required communications interfaces and I/O and control functions, CICS isolates the user's applications programs from the communications environment and, to a considerable degree, from the operating system itself.

Written in Assembler language, CICS provides transaction processing support for data base management or file control programs written in Assembler, PL/1, or COBOL, thus allowing on-line applications to be developed without significantly greater difficulty than similar batch programs. In addition to supporting several external data base management structures (e.g., IMS's DL/1, DBOMP) CICS includes enough native data management capability to be considered a data base/data communications (DB/DC) system.

OPERATION: CICS operation requires that the user perform the following steps (presented in the typical development sequence, although considerable overlapping normally takes place):

- A CICS installation planning and project management team must be assembled to set up standards, schedules, and strategies for implementation of CICS. The suggested division of labor in such a group calls for a project manager with authority over three separate personnel functions: (1) applications program development programmers, (2) control of data preparation and testing, and (3) system programmers to handle CICS generation and interfacing. The size of the entire team will vary with the complexity of the installation, but will typically run from two persons in a small environment to more than a dozen individuals in a large-scale system. An important function carried out by the project manager is the establishment of an in-house training program.
- The potential applications for the CICS system are reviewed, and one or more of the simpler applications (e.g., inquiry, data collection, conversational "front end" to a batch program for data entry/validation, etc.) are selected for pilot CICS implementation.
- All of the individual data requirements for each application (not just the first handful of pilot applications) must be reviewed, and a determination made as to whether one or more "centralized" data bases will be required. This task is most appropriately handled by an individual serving as a Data Base Administrator in the data preparation and testing area. If a data base is required, then efforts should be understaken to set it up (see IMS, Report 70E-491-01). Although the simpler pilot applications may not caii for an "external" data base, the more long-range MIS plans probably ►

are not specifically designed to take advantage of virtual storage. Thus, their use could be highly inefficient. The new virtual-memory versions of CICS, however, are designed for VS operation. CICS/DOS/VS will be delivered in February 1974, and CICS/OS/VS in January 1974.

THE STRUCTURE OF CICS

The original CICS Version 1 was a data communications monitor that facilitated access by Assembly-language programs to the standard IBM data structures provided by the operating system (e.g., ISAM), plus access to somewhat enhanced structures (e.g., ISAM incorporating 2-level record segmentation and file chaining). User-program interfacing for COBOL or PL/1, as well as support for other than a few basic types of terminal devices, was provided by an optional Language and Terminal (L&T) feature.

Subsequent CICS developments not only incorporated the L&T feature as a standard component in CICS, but also acknowledged the potential for further efficiencies to be gained in an on-line environment by adding certain file management capabilities to CICS that historically had not been needed for batch-oriented systems. Very significantly, these facilities turned out to be precisely those fundamental criteria that identify a data base management system: a shared data base manager, a degree of data independence, and something more than the standard operating system facilities to interrelate physical data structures. In particular, these developments have included both embedded data management facilities (File Browse/Generic Search, Exclusive Record Access, Dynamic File open/close) and external data management interfaces (DL/1, DBOMP).

Significant developments continued with the VS versions of CICS. Transactions can now be invoiced by using one to four characters or a single terminal function or attention key, instead of exactly four characters. User VSAM data base file support has been added, with several new file management functions included. Flexibility of use has been advanced by new message switching and routing facilities, terminal paging (storing more output than can be displayed at one time, with the pages retrievable in any order by the terminal), and terminal device independence. The latter new feature is supported for 1050, 2740, 2741, 2770, 2780, 3270, tape, disk, TWX, and card reader/line printer devices.

New macro instructions in CICS/VS provide some interesting and potentially very useful file operations. Among these are phonetic conversion and weighted retrieval. The former provides a means to convert a name into a key based on the phonetic sound of the name to allow organizing data based on names that might be misspelled, mispronounced, or misunderstood. The latter allows the application program to search a group of records in a VSAM data set on the basis of selection criteria. For example, all customers whose accounts are more than 30 days past due and which exceed $1,000 could be identified from a customer master file.

A major extension of CICS/VS is the restart/recovery capabilities not previously available. Warm start and jour-

will indicate a need for early data base planning that can be started in paralled with the CICS installation.

- **The individual applications programs are then written, using CICS macros in-line with COBOL, PL/1, or Assembly-language statements to produce an executable program with CICS interface logic imbedded within it. (This process is described in more depth below).**
- **A system test is carried out using CICS's terminal simulation tools for initial testing, followed by tests with the actual terminals. Performance measurements are made to determine response time under various system loading conditions, etc.**
- **Changes are made to the CICS internal table parameters, I/O buffer areas, etc., to "tune" the system for desired response time and overall throughput versus demands upon system resources.**

Of the activities listed above, the application program compilation is of particular interest. At the time that the user programs are written, various CICS macros must be inserted to handle the telecommunications interfaces. For high-level languages (COBOL or PL/1), the source program is put through the following steps:

- **A special pre-processor run where the high-level language statements are prefixed by "PUNCH" and enclosed in quotes. These "PUNCH" statements are acceptable to a standard 360/370 Assembler. The CICS macros are not altered.**
- **A standard Assembler run that converts the CICS macros into COBOL or PL/1 statements and strips the "PUNCH" and quotes from the original source code. The output from this Assembler run, paradoxically, is a source program in the high-level language.**
- **A standard compilation that produces an object version of the program containing interface logic to the appropriate CICS data communications and data handling modules.**

Although this multi-step process does introduce additional complexity into the debugging task (e.g., "fatal" CICS macro coding errors can be carried only as warning-level diagnostics into the final object program, where they will then cause an ABEND during execution), users report that the overall complexity of the above process is not prohibitively greater than ordinary high-level language compilation. The reason for the seemingly unnecessary multi-step process is that in this way, the standard COBOL and PL/1 compilers can be "tricked" into producing re-entrant programs, thus enabling multi-tasking to take place. The process for Assembler-language programs is somewhat simpler, with no pre-processing run required.

The intricate process described above for applications program development in high-level languages is a good example of exactly why a carefully thought-out and coordinated CICS project implementation plan is necessary for a smooth and successful implementation of CICS. For instance, the optional (albeit infrequent) use of user-written direct interfaces with the operating system instead of standard CICS macros can limit the extent to which resulting applications programs are compatible between the DOS and OS versions of CICS. This problem and numerous others related to the flexibility of CICS can easily be countered by well-developed standards.

The DOS and OS Standard versions of CICS are multi-thread systems, while the DOS Entry version is a single-thread system. All three versions operate as a single task within a partition or region under multiprogramming or in a dedicated environment. The internal structure of CICS sets up a sort of intrapartition "multiprogramming" environment that is under direct CICS control. Many of the storage

CICS
IBM Corporation

Relationships between the CICS Versions and Operating Systems and Data Base Programs

Program ID	Operating System (and mode)				Interface for:
	DOS	DOS/VS	OS	OS/VS	
CICS/DOS Entry 5736-XX6	yes	real or virtual*	NA	NA	VANDL-1
CICS/DOS Standard 5736-XX7	yes	real or virtual*	NA	NA	VANDL-1
CICS/OS Standard 5734-XX7 (Version 2)	NA	NA	yes	real or virtual*	IMS Version 2 or IMS/VS
CICS/DOS/VS 5746-XX3	NA	virtual	NA	NA	DL/1 DOS/VS
CICS/OS/VS 5740-XX1	NA	NA	NA	virtual	IMS/VS

*The virtual storage version should be used for optimum performance in VS mode under VS operating systems.

▷naling are among these capabilities. Journaling allows the user to log data to a journal (i.e., create an audit trail) during real-time CICS execution as an aid to recovery management. But the user must provide his own recovery mechanism, since CICS does not provide any support in the interpretation of a journal's contents.

Because CICS satisfies the fundamental IBM "tests" to qualify as a data base system, the DB identifier has been added to CICS's primary data communications (DC) classification, even though the DB facilities present in CICS are rather basic ones. Nonetheless these DB facilities tend to reduce on-line control program dependency upon the operating system by placing responsibility for certain critical response-time elements under the immediate jurisdiction of the data communications monitor.

In fact, a major source of current confusion about CICS is related to a common lack of understanding of its role in the hierarchy of IBM DB/DC Program Products, and, in particular, of how CICS fits into the big picture with IMS-2. In reality, CICS's classification as a DB/DC system is a mere technicality, and the great majority of present computer users are well advised to view CICS as a "straight" data communications monitor, or as the data communications half of a DB/DC system that uses IMS's DL/1 or some other external data base manager to provide the data base. As a rule of thumb, when the data structure for an on-line application can be handled by standard methods (e.g., ISAM) or by a slightly more powerful chained file capability, etc., CICS can be use without external data base support. However, if more flexible and intricate data structures are needed, an external data base manager should be used (e.g., IMS's DL/1).

CICS-OS Standard can be interfaced to full IMS-2, and CICS/VS can be interfaced to IMS/VS. Either DOS version of CICS can be interfaced to VANDL-1, the interim version of IMS under DOS or DOS/VS, or CICS/VS itself can interface to DL/1 DOS/VS files. DL/1 DOS/VS is the full DOS/VS version of IMS, and will be available for batch systems with DOS/VS initial shipment in December 1973. Two months later, it will be ready for CICS/DOS/▷

➤management and program management modules in CICS function in a way similar to that of comparable MVT modules. In fact, CICS is often compared to MVT and bears a striking internal resemblance to the structure of that operating system.

The DOS/VS and OS/VS versions of CICS are analogous, respectively, to the DOS Standard and OS Standard versions. The VS versions have been somewhat enhanced from a functional point of view, and partially optimized to run more efficiently in a paged environment than the real-memory versions would.

The CICS partition or region is physically divided into two types of main storage:

- Static Storage, in "high" memory, which contains the CICS Nucleus, service programs, control tables, access methods, and resident user-written applications programs; and
- Dynamic Storage, in the rest of the partition or region, which contains work areas, I/O buffers, applications programs to be processed, the storage cushion, and certain additional control areas. The size of Dynamic Storage greatly influences CICS throughput and response time.

The CICS modules can be grouped into three general categories:

1. Supervisory Functions

Task Management – a Task Control Program (TCP) that handles task origination, dispatching, priority change, suspension/resumption, runaway detection and correction, maximum task control, stall control, and task termination.

Storage Management – for storage request queues, inhibition of new work under full-load conditions, storage acquisition, initialization, disposition, and accounting.

Program Management – a Program Control Program (PCP) that loads, links, and transfers control to user-written "quasi-reentrant" applications programs and deletes, terminates, and returns control from these programs.

Program Interrupt Management – an optional CICS interrupt handler module to transfer control to the operating system upon encountering a program interrupt.

Time Management – for CICS exit time interval control, detection of system stall or runaway task loops, time-of-day, ➤

CICS
IBM Corporation

VS. The accompanying table illustrates the relationships presently existing between all the available versions of CICS and both the IBM operating systems and the data base (DB) programs with which each can interface. The table also shows the mode (real or virtual) for VS systems.

CORPORATE IMPLICATIONS

CICS is currently basking in the glow of some of the hottest marketing and development support ever bestowed upon a Program Product by IBM. With the possible exception of IMS-2, CICS is the most vigorous and significant cross-industry program offered by IBM today. The two DOS (or DOS/VS) versions of CICS and the OS (or OS/VS) version have a combined total of well over 500 installed systems. The next most widely sold non-IBM competitor, GTE's Intercomm (Report 70E-457-01), has about 70 installations, while nearly a dozen other packages (including Environ/1, Report 70E-132-02, and Task/Master, Report 70E-866-01) together account for fewer than 100 systems.

At the time this report was written, at least three copies of Intercomm were installed and operating under OS/VS1, beating the announced availability date for Intercomm/VS by a month. The package runs in a virtual mode and has been optimized to reduce page fixing. It is compatible with real-memory OS operation if its VS module is not used, and the improved package is provided to all new and present users.

Environ/1 has been run in a virtual mode, but it is not yet installed at any virtual memory user's site. Cincom, its vendor, promises to deliver a compatible VS version of the package to future users. The package will be the present version optimized and enhanced; it presently can run in either the real or virtual mode.

Task/Master, from Turnkey Systems, is up and running in the virtual mode at three users' installations. A new version that will be specially "tuneable" to virtual storage operation is scheduled for announcement as this report goes to press, with initial delivery in the fourth quarter of 1973.

Although all of these competitive systems promise better performance than CICS and some enjoy considerable success, none has the extensive acceptance, popularity, and field support that CICS does. Of particular interest is the recent emergence of IBM Program Products that rely upon CICS as a host. At present there are nearly a dozen such applications programs, and the number is growing steadily; they are identified under Pricing in the Characteristics section of this report. The availability of these systems must be considered among CICS's assets when comparing data communications monitors.

The CICS DOS-Entry (DOSE) version is a "single-thread" system that can process only one task or transaction at a time. Whenever an incoming transaction is encountered, the previous transaction and its associated data areas are "rolled out" for subsequent restoration to main memory when completion of that task is scheduled. If the incoming transaction requires a different applications program, the new program is read into main memory over the

time-dependent transaction synchronization, and automatic time-ordered transaction initiation.

Dump Management – a diagnostic facility for selective dumping of system control tables, storage areas, and programs.

Journal Management – consists of the Journal Control program, the Journal Control Table, and CICS macroinstructions used to log and retrieve data. Several journal data sets can exist on tape or disk, and, by using the macroinstructions, the user can place any data set referenced in the Journal Control Table. Journaling is a valuable CICS/VS recovery feature.

2. Data Management Functions

Terminal Management – a Terminal Control Program (TCP) handles communications between terminals and applications programs. All versions of CICS use BTAM for terminal data management, with GAM and TCAM also available for the OS Standard versions. TCP also handles terminal error recovery, device-dependent control functions, transmission facility control, and the simulated terminal capability.

File Management – the File Control Program (FCP) handles data base operations, including data set open/close, retrieval, browsing, exclusive control, and 2-level segmentation. Other file management functions include external data base manager interfaces to DL/1, DBOMP, etc.

Transient Data Management – for temporary storage of data to be used for subsequent processing. Extra-partition data transfers are also accommodated for system statistics, etc. This facility can also be used for message switching.

Temporary Storage Management – controls "scratch pad" areas in main memory and temporary data storage areas on direct-access devices.

3. Non-Real-Time Functions

System Initialization – a program that runs in the Dynamic Storage area and is used to configure the CICS nucleus, establish system performance parameters, set up storage boundaries, and bootstrap the CICS modules into memory.

High-Level Language (HLL) Pre-Processor – used to convert CICS macros or user-written assembler routines into high-level language for subsequent compilation.

CICS/DOS Maintenance and Linkage Editor – performs functions similar to those of the DOS Librarian and Maintenance and Linkage Editor for the CICS partition. This CICS module maintains the Relocatable Program Library with create, copy, delete, condense, link edit, and display functions.

Dump Utility – operates in batch mode to format CICS dumps.

In addition to the basic CICS functions described above, more than a dozen other system service functions are provided by ancillary application-level programs; these include Sign On/Sign Off (an optional security module), Master/Supervisor/Operator Terminal Authorization, system statistics (useful in reconstruction as well as control and audit procedures), abnormal condition handlers, orderly system shutdown, Asynchronous Transaction I/O Processors (ATP) for Standard CICS only, trace facilities (an optional module for debugging environments), dynamic open/close, and time-of-day controls. Of particular interest is the ATP facility, a sort of RJE capability without a Job Control Stream that allows transactions and data to be batched for processing. The ATP facility is designed for high-speed data entry terminals.

The highly modular construction of CICS, as indicated by the above system segments, has facilitated the growth of

CICS
IBM Corporation

Memory Requirements for CICS

CICS Version	DOS Entry	DOS Standard	OS Standard, Version 2		DOS/VS	OS/VS	
Operating System	**DOS or DOS/VS**	**DOS or DOS/VS**	**MFT or VS1**	**MVT or VS2**	**DOS/VS**	**VS1**	**VS2**
Total main memory:							
Absolute minimum CPU and "real" memory	360/25 with 48K	360/30 with 64K	360/40 with 128K	360/40 with 256K	370/125 with 128K	370/135 with 192K	370/145 with 384K
Typical* CPU and "Real" memory	360/30 or 370/135 with 96K to 128K	370/135 or 370/145 with 128K to 256K	360/50 or 370/155 with 512K	360/65 or 370/165 with 1,024K or larger	370/135 with 192K or larger	370/145 with 512K or larger	370/158 with 1,024K or larger
CICS partition/region:							
Absolute minimum size, bytes	30K	44K	64K	64K	48K Real***	48K Real***	48K Real***
Typical* size, bytes	70K	90K	175K	175K	100K	320K	460K
Partition/Region Sections:							
Static portion (includes Nucleus, management functions, control tables, access methods)	35K	40K	80K	80K	88K	256K	358K
Dynamic portion (includes working storage, application program(s). buffers)**	35K	50K	95K	95K	12K	64K	102K
Practical minimum resident operating system size (based upon CICS requirement only; does not include support facilities for ancillary programs such as IMS-2, etc.)	DOS-22K DOS/VS-36K	DOS-26K DOS/VS-40K	MFT-100K VS1-135K	MVT-180K VS2-250K	40K	135K	250K

NOTE: Total main memory size minus CICS partition/region and resident operating system requirements leaves available main memory for additional partitions/regions. Estimates for CICS/DOS/VS and CICS/OS/VS are for virtual address space, not main storage, except where designated "Real."

* Mature size attained by average (realistic) CICS installations.

** Size of dynamic storage is critical to CICS throughput and response time.

*** 64K virtual address space is minimum requirement.

existing program. DOSE is designed to handle limited-size terminal networks, generally no larger than 15 to 20 terminals. For larger configurations, or when the work load reaches a point where the arrival queues consistently hold enough work to occupy the system for more than 50 percent of the elapsed clock time, then some combination of the following steps is likely to be needed: (1) increase the amount of real memory, (2) install a more powerful central processor, or (3) upgrade to one of the Standard versions of CICS. The most common approach to improving throughput is to upgrade to a Standard version.

The cut-over point for transition between the DOS Standard and OS Standard version of CICS is not as clearly defined as that between the Entry version and either of the Standard systems. Both Standard CICS versions can handle large numbers of terminals (any configuration with more than 30 or 40 terminals would qualify as a "large" system in common parlance, although CICS networks with several hundred terminals have been installed), and there is no way to define a sharp line to separate DOS Standard and OS Standard job environments using CICS considerations alone. Even though there are a number of internal technical differences between the two Standard versions (e.g., OS has better concurrent I/O request handling to the same file), batch considerations are primarily used to differentiate between DOS Standard and OS Standard. In other words, both Standard versions provide essentially similar functional capabilities, and it is up to the user to determine whether DOS or OS will best satisfy his overall requirements. OS provides a generally more comprehensive data handling and data communications environment (usually on a larger and more powerful processor), with consequently better overall throughput and faster on-line response-time potential.

CICS by the addition of new functional capabilities without the need for redesign of the basic system.

HARDWARE/SOFTWARE REQUIREMENTS: CICS runs on System/360 or 370 computers under DOS or under OS/MFT or MVT (or their VS counterparts). Specific memory requirements for all versions are summarized in the table. In addition, each version of CICS requires at least one magnetic tape unit, 2.5 million bytes of direct-access storage (for CICS libraries and working storage), and one "Master Terminal." (The latter is a "logical" unit requirement and does not necessitate a separate, full-time terminal.)

In addition to the indicated memory requirements, the user must add a minimum of 16K bytes of "real" storage for IMS if the DL/1 data base interface is required (see IMS-2, Report 70E-491-01).

COMPATIBILITY: Application programs developed for CICS/DOS-Entry (5736-XX6) and CICS/DOS-Standard (5736-XX7) are upward-compatible with CICS/DOS/VS (5746-XX3). Upward compatibility at the source code levei also exists from CICS/DOS/VS to CICS/OS/VS (5740-XX1). Also upward compatible to CICS/OS/VS at the source code level are CICS/OS-Standard Version 1 (5736-U11, with Language/Terminal feature) and CICS/OS-Standard Version 2 (5734-XX7). Also, the DOS/VS version supports the IBM 7770 Model 3 Audio Response Unit, but CICS/OS/VS does not.

If a user is planning a future migration from CICS/DOS/VS to CICS/OS/VS, he should keep the following, albeit minor, incompatibility in mind: In CICS/OS/VS, scheduling of resources required to access DL/1 data bases must be handled explicity in the application programs by issuing a PCB (program control block) CALL. Under CICS/DOS/VS, users can do the same or rely upon default options (the former called explicit, and the latter, implicit, resource scheduling). If implicit resource scheduling was used, a CICS DOS/VS to OS/VS upgrade would involve finding appropriate points in each application program at which to insert a control card. If explicit resource scheduling had been used all along, most likely, all that would need to be

CICS
IBM Corporation

USER REACTIONS

Of the numerous CICS users contacted directly by Datapro, all reported that they are at least reasonably well satisfied, and many expressed a high degree of satisfaction with CICS. Almost universally, however, each user had his own list of CICS shortcomings; these ranged simply from confusion over the annoyingly equivocal nature of CICS as a DB/DC package at its current stage of development to severe criticisms that point to design faults in the system. Among the most meaningful of the latter are CICS's lack of storage protection for individual applications programs and its lack of an effective Terminal Error Program (TEP) to cope with terminal/line failures.

CICS operates as a single task within a partition or region, with individual applications programs handled as sub-tasks under CICS in that partition/region. The IBM System/360 and 370 computers provide only one hardware storage protection key per partition, so it is not possible to give the same degree of storage protection to each applications program within a single partition that could be afforded the programs if each were in a separate partition. To this extent, the architecture of CICS and the System/360 or 370 jointly preclude the use of the most effective type of applications program storage protection available in the 360/370 hardware. On the other hand, a certain amount of software-based storage protection can be developed, although this has not been done to date.

For terminal error handling, CICS initiates a basic default action that effectively shuts down any line that has a faulty terminal on it and enters an appropriate error message in a CICS internal table. In addition to this, user "hooks" or exits are embedded in CICS that enable the user to process error conditions by writing his own terminal error handling routines. Obviously, a generalized package of error routines would be a welcome addition to CICS, but no such development is available from IBM at this time. The terminal independence announced for VS versions of CICS will at least allow the owner of a faulty unit to quickly substitute another device, should one happen to be available.

Partly as a general response to user needs for customized terminal error handling programs, and partly as a response to the inherent complexity of CICS, a number of independent CICS consultants have emerged (e.g., Quantra Development Corporation, Rye, N.Y., Computer Horizons Corporation in New York, N.Y., and On-Line Software Incorporated in Hackensack, N.J.). Although IBM does not encourage the use of independent CICS expertise, these firms nevertheless have numerous well-tested enhancements and extensions to CICS that are of general interest. Depending upon the availability of local IBM talent, users may want to consider such outside sources for custom coding, etc.

RECOMMENDATIONS

No System/360 or 370 user with a combined data communications and data management requirement (e.g., on-line access to a data base), should attempt to satisfy his software needs without acquiring a thorough understanding of CICS. Although the ultimate solution to his done is locate the existing control cards to change the storage requirements for the PCB's. That task could be performed by lower-level personnel than those required otherwise.

PRICING: The CICS programs, for the most part, are standard IBM Program Products with full, centralized IBM programming support. In some cases, Field Developed Programs (FDP's) and/or Installed User Programs (IUP's) for use in conjunction with CICS will be offered with limited support for a six-month period following their initial introduction. The support period for such programs is indicated.

CICS Programs	Monthly License Fee	Test Allowance
Customer Information Control System/DOS-ENTRY (CICS/DOS-ENTRY), 5736-XX6. Single-thread System. Class A service.	$200	2 months
Customer Information Control System/DOS-STANDARD (CICS/DOS-STANDARD), 5736-XX7. Class A service.	500	2 months
Customer Information Control System/OS-STANDARD Version 2 (CICS/OS-STANDARD V2), 5734-XX7. Class A service.	700	2 months
CICS CPU Console Feature, 5798-ANK. Allows the CPU console to be used as the logical master terminal for any version of CICS. Error correction support for this FDP ends by May 1973.	55†	1 month
Customer Information Control System/DOS/Virtual Storage (CICS/DOS/VS), 5746-XX3. Class A service.	350	2 months
Customer Information Control System/OS/Virtual Storage (CICS/OS/VS), 5740-XX1. Class A service.	750	2 months
Ancillary Programs		
Storage and Information Retrieval System (STAIRS), 5734-XR3. A Class A Program Product for bibliographic/textual abstract data base management and retrieval under OS-STANDARD.	500	2 months
Interactive Training System, 5734-XXC. A Class B Program Product that provides facilities to create, execute, and maintain computer-assisted training courses under OS-STANDARD.	225	2 months
Course Structuring Feature for Interactive Training System.	25	2 months
Fast Information Retrieval for Surface Transportation (FIRST). A Special Program Product for full-scale truck company vehicle management system under DOS or OS STANDARD:		
OS Message Switching Module, 5795-AAA	280*	1 month
OS Equipment Control Module, 5795-AAB	1,000*	1 month
DOS Message Switching Module, 5795-AAC	280*	1 month
DOS Equipment Control Module, 5795-AAD	1,000*	1 month
Registered Representative System, 5734-F34. A Class B Program Product brokerage system that allows registered representatives in branch offices to access stock market and customer data and enter direct buy/sell orders. (Available	12,000**	2 months

requirements may well be an in-house development effort, a different IBM Program Product, or a competitive data communications monitor from a non-IBM source, CICS should nonetheless be examined for reference in a comparative study.

Of particular interest since the August 1972 announcement of System/370 virtual storage by IBM is the question of how virtual storage will affect CICS and data communications monitors in general. These systems typically operate under stringent response time criteria, and there is an extensive, intimate interface between such monitors and the operating system — to a considerably greater extent than exists for ordinary applications programs. Thus, the question is how will the response time and throughput of CICS and competitive systems be affected under paging?

Every user owes it to himself to thoroughly investigate both CICS and the non-IBM alternatives, with their promises of higher performance, usually at lower costs. Whatever data communications monitor is selected will entail a considerable commitment in time and money, and it is not easy to change systems in midstream.

CICS is a very good data communications monitor that has gained widespread acceptance. It is often selected by first-time data communications users, for whom it represents an excellent choice – particularly at the DOS Standard level. First-time users who take the time to assemble an appropriate development team will typically be able to get simple pilot applications up and running within three to six months.□

for OS STANDARD only by October 1973.)		
Data Base Organization and Maintenance Processor (DBOMP), 5736-XX4. This Class B Program Product is available for DOS ENTRY or STANDARD.	100	2 months
CICS Feature for DBOMP to allow on-line retrieval and maintenance of DBOMP files by CICS applications programs. (Available in January 1973.)	75	2 months
Alpha Search Inquiry System, 5736-N14. A Class B Program Product for all versions of CICS. Originally developed for the insurance industry to allow phonetically-based information retrieval.	200	1 month
Health Care Support Data Communications Program, 5746-H13***. This Class B Program Product provides on-line facilities for the medical industry health care programs. (Available for DOS-STANDARD only by March 1974.)	295	1 month
Customer Information File (CIF), 5798-AHX. A DOS-STANDARD data communications support facility for the CIF Banking package. Error correction support for this FDP ended in November 1972.	1,320†	2 months
Information Management System/360 Version 2 (IMS-2), 5734-XX6. This Class A Program Product is IBM's most sophisticated data base manager. For use with OS STANDARD.	550	2 months
Vancouver Data Language/1 (VANDAL), 5799-AEY. This Program Product RPQ is an entry-level DOS version of the sophisticated IMS/360 data base management system (DL/1). A CICS DOS ENTRY or STANDARD interface is available.	350	1 month
Display Management System (DMS), 5734-SC1. This Class B Program Product provides an interactive CRT display-oriented network system.	185	1 month

* charges waived after 3 years.
** charges waived after 5 years.
*** for DOS/VS only.
† charges waived after 1 year.

INITIAL DELIVERY: CICS has been delivered in several releases, as summarized below.

Release Number	Distinguishing Characteristics	Release Date
DOS ENTRY		
1.0	Basic single-thread system	September 1971
1.1	3270 support, Browse, DL/1 interface	July 1972
DOS STANDARD		
1.0	Basic multi-thread system	November 1971
1.1	3270 support, Browse, DL/1 interface	July 1972
OS STANDARD		
1.0	Basic System for Assembly language, 2260, 1050, etc.	June 1969
–	Language and Terminal Feature (COBOL, PL/1, additional terminal types) for Basic System	December 1970
2.0	Incorporated Language and Terminal Feature for COBOL, PL/1; support for 1030, 2741, 2780, 2770, 7770, 360/20, 1130, etc.; exclusive "lock-out" control shifted downward to record level from file level; performance improvements; Browse	December 1971
2.1	ATP, reusable queues, user exits.	May 1972
2.2	Support for 3270; DL/1 interface	July 1972
2.3	TCAM Release 4 support	December 1972

The OS/VS and DOS/VS versions of CICS are scheduled for delivery in January and February 1974, respectively.

CURRENT USERS: Well over 500 CICS installations of all types. (These are based upon current market research information; IBM does not release official installation figures.)■

Program Products - Applications
IBM Corporation

This report describes the currently available applications-oriented Program Products for the IBM System/360, System/370, System/3, and 1130 computers. You'll find a brief functional description of each program, together with its equipment requirements, service classification, and price.

INTRODUCTION

On June 23, 1969, IBM rocked the data processing establishment by making its famed unbundling announcement. An important part of the announcement was the future plans of the firm for making programming support available. Prior to unbundling, IBM's four classes of software were available to equipment users at no charge. The announcement changed all that, and, while users began to pay for some software, IBM cut its equipment prices by about 3% across the board.

Today, IBM programming support is divided into two basic classes, *System Control Programming (SCP)* and *Program Products.* The former are fundamental to the operation and/or maintenance of the system, and are no-charge items; the latter are related to an application of the system to user tasks, and are available to all at specified charges under a license agreement.

Program Products are divided into two subclasses: *Program Products–Systems (PP)* and *Program Products–Applications (PPA).* PP's are usually language compilers, sorts, conversion aid programs, and general-purpose utilities, while PPA's are industry and general application programs. Descriptions of the available PP's are included in the DATAPRO 70 reports on individual IBM computer systems (Reports 70C-491-02 through 70C-491-22).

This report concentrates on brief descriptions of the currently available PPA's. A short functional description of each PPA is presented, along with a statement of the program's hardware and operating system requirements, license fee, service classification, and IBM form number of the most general nonpromotional, no-charge manual for the program. Listed in alphabetical order by title, you'll find descriptions of the PPA's for the System/360 and 370, followed by those for the System/3 and the 1130.

PPA's falling under the heading "System/360 Application Programs" can be used on System/370 computers as well, unless specifically noted otherwise. As time goes by, however, programs designed for use exclusively on the System/370 will creep into the list; one current example is the System/370 Automatically Programmed Tool set of three programs.

Each program's *ID number* tells you something about its machine and operating system requirements: 5736 = DOS (or DOS/VS); 5734 = OS/MFT or MVT (or, correspondingly, OS/VS1 or OS/VS2); 5735 = multiple environment (System/360 and 1130, for example); 5746 = DOS/VS; and 5740 = OS/VS1 or OS/VS2. Unless noted, a real memory requirement can be satisfied in a virtual memory environment, and the application program in its environment can be accommodated within the virtual environment of VM/370.

Also, generally, 5701 = System/3 Model 10 Card System; 5702 = System/3 Model 10 Disk System; 5703 = System/3 Model 6, and 5711 = 1130 System. Exceptions to this scheme do occur from time to time.

Some PPA's are important enough to warrant special consideration, and are described and evaluated in detail elsewhere in DATAPRO 70. IBM's CICS and IMS (Customer Information and Control System and Information Management System) are examples; see Reports 70E-491-02 and 70E-491-01, respectively. Charged-for programs that are not PPA's, but that are important to users, such as VIDEO/370, are also described within this report.

As an additional reference aid, the IBM System/370 report (70C-491-04) includes, at the end of the price list, a listing of both PP's and PPA's in a matrix showing their prices and required operating environments.

The *License Agreement for IBM Program Products* (IBM Form 120-2065) establishes the initial agreement between IBM and a customer and the initial order for Program Products. A supplement to the Agreement (Form 120-2061) is sent to the user by IBM for each subsequent Program Product that the user acquires. A separate Agreement is required for each system that runs under control of its own supervisor (that is, one Agreement will do for a multiprocessor MP65 system, but two 360/65's sharing files only would require two Agreements). A Program Product must normally be used on the specific CPU to which it is licensed. Minimum term of use of a Program Product is three months, though fees for a few are waived after one month of use. When use of a Program Product is discontinued, the user signs and gives to IBM a Certificate of Discontinuance (Form 120-2068), indicating that the Program Product has been destroyed.

Service classifications for PP's and PPA's have the following interpretations:

- Class A–no-charge Central Programming Service and no-charge Field Engineer Programming Service (unless

▷

Program Products — Applications
IBM Corporation

▷ the call is due to a customer error or when no specific problem is identified); automatic distribution of corrections; and, in some cases, field engineers will apply program bypasses, or "fixes", in the field.

- Class B—no-charge Central Programming Service and hourly or on-call Field Engineer Programming Service.
- Class C—Field Engineering Service on an hourly or per-call basis for problem diagnosis and corrections of problems requiring under 8 hours.

In addition to the PPA's described in this report, IBM now offers two other classes of separately priced applications programs: *Field Developed Program (FDP's)* and *Installed User Programs (IUP's).* Limited support is provided for the FDP's and IUP's (which were first made available in August and October 1971, respectively); it consists only of pertinent error-correction information during the first six months after initial general availability of each program. A full list of FDP's and IUP's with prices, dates when support ends, and reference manual numbers can be found in the *IBM Customer Information Card for FDP's and IUP's* (GB21-9949).

Users should also be aware that literally thousands of IBM's earlier Type II and II programs are still available. Released prior to the unbundling announcement, these programs are offered at no charge—but their documentation may be poor, or even nonexistent, and little or no free IBM support is provided. □

SYSTEM/360 APPLICATION PROGRAMS

Active Certificate Information Program

This program is designed for implementation within the cashier's department of a brokerage firm or commercial bank to provide on-line data entry, inquiry, and posting to a certificate inventory and requirement file. IBM 2260 Display Stations are used to track the movement of stock certificates to and from the Active Box Section of the cashier's department, displaying the user's security position at any point in time. A summary file of pending delivery instructions is updated daily and posted by the Active Certificate Information Program as the execution of deliveries is indicated on the 2260 Display Station.

Among the specific features of the program are cage and vault CRT displays of security resource information; summarized security requirements; reconciled physical box counts; computer assisted look-up of CUSIP number or firm's security number; accommodations of common and preferred stock, corporate, municipal, and government bonds; rights and warrants; and inquiry by ticker symbol.

The program runs in a 44K partition under DOS or DOS/VS. The minimum machine configuration includes three 2311 Disk Drives and sufficient tape units to back up the disk files. Basic manual: IBM Form GH20-0775.

	Monthly License	Service Class
Active Certificate Information Program (DOS) 5736-F32	$300	B

Advanced Life Information System (ALIS/2)

ALIS/2 is written in Assembler Language and comes in a basic system—Daily Cycle and Valuation Programs—with a Home Office Inquiry Module. The system is designed to handle the major processing requirements for ordinary life insurance from the point in time immediately following policy issue through maturity, expiration, or termination of the contract, providing immediate direct-access inquiry.

ALIS/2 processes scheduled and non-scheduled transactions, provides figures for annual and interim statements to update policies with cash values, calculates dividends and other forms of participation, and updates policies with renewable term premiums on or just before anniversary.

The Home Office Inquiry Module of ALIS provides CRT displays of a partial image of the policy master record (plan, face value, mode premium, issue, paid-to and billed-to dates, etc.); the name and address of the insured or payor; and the quotations for cash surrender, conversion to nonforfeiture option, withdrawal par values on deposit, withdrawal of paid-up additions, maximum loan value, loan payoff, and premium.

Each ALIS/2 user must modify or "tailor" the system to conform to his own unique business practices. Therefore, the first step is to determine these modifications and then generate a policy master record file containing all of the information required by ALIS/2. The policy master record file and unscheduled transactions are then processed; scheduled transactions require no input other than the information contained in the policy master record. An output analysis program routes the results to various subsystems for final printing. Intermediate files are also produced for use by the policy exhibit and valuation programs. The Home Office Inquiry feature of ALIS can be used as soon as a policy master record file has been generated.

The ALIS/2 Daily Cycle and Valuation Programs run in a 56K partition under DOS or DOS/VS, while the Home Office Inquiry version requires a 44K partition. The minimum machine configuration includes a 2321 Data Cell, one disk drive, and four tape units. Basic manual: IBM Form GH20-0126.

	Monthly License	Service Class
ALIS/2 (DOS) 5736-N11	$500	B

Agribusiness Management Information System (AMIS)

AMIS is an enterprise accounting system for farm or agricultural cooperatives and large ranches. As the user submits checks, deposit slips, production reports, and other normal working documents to this Assembler–language system, AMIS produces a comprehensive set of reports, to the individual item level, showing product status, profit or loss, etc. Among the special farm business management controls provided by AMIS are full tax liability accounting, tight control over variable expenses, employee payroll, net worth reports, and depreciation allowance notices. ➤

Program Products — Applications
IBM Corporation

AMIS runs in a 20K partition under DOS or DOS/VS. The minimum machine configuration includes three 2311 Disk Drives. Basic manual: IBM Form GH20-0765.

	Monthly License	Service Class
AMIS (DOS) 5736-D51	$225	B

Automatically Programmed Tool (System/370)
Basic Positioning (BP)
Intermediate Contouring (IC)
Advanced Contouring (AC)

The APT (BC, IC, and AC) programs form an upward-compatible family of numerical control processors for machining operations from point-to-point through 3-axis contouring. The APT programs extend the capabilities found in the type II N/C processor programs for the System/360.

APT-BP supports point-to-point and simple line-circle contour machining. APT-IC extends these capabilities to more complex two-dimensional surfaces. APT-AC includes the foregoing capabilities and adds three-dimensional surface operations. The APT programs are written in FORTRAN IV and Assembler H-level and require the OS/VS system assemblers and linkage editors. They operate under OS/VS and VM/370.

Basic requirements include a 370/135 CPU with 147K (for BP) or 245K (for IC) or a 370/145 with 393K (for AC) under OS/VS1; or a 370/145 with 262K (for BP or IC) or 524K (for AC) under OS/VS2. Also required are three 3330 Disk Drives, a printer, and a card reader and punch. Basic manual: IBM Form GH20-4228.

	Monthly License	Service Class
Automatically Programmed Tool:		
BP (OS/VS) 5740-M51	$150	B
IC (OS/VS) 5740-M52	300	B
AC (OS/VS) 5740-M53	500	B

Basic Courts System

This is an on-line, terminal-oriented system that provides a means for maintaining and viewing the up-to-date court calendar and status of all persons and cases before the court. It has calendaring, case history, name index, and identification number index portions. The system uses FASTER-LC or DOS FASTER MT to automatically retrieve and maintain court information. A unique feature is the ability to allow entry of "prose" data, such as a judge's non-standard sentence or calendar clerk's notes.

FASTER LC or DOS FASTER MT are prerequisites for the Basic Courts System, whose batch portions are COBOL programs that are executed as background jobs under DOS. Used with FASTER LC, the Basic Courts System requires a 56K DOS partition, a 360/30 or larger CPU with decimal arithmetic, selector channel, and interval timer, 2311 or 2314 disk space for files and work areas, a printer-keyboard, a card read punch, a printer, a 9-track tape unit, and terminals as required. Larger storage makes hard-copy terminals possible. With DOS FASTER MT, a 114K partition in a 360/40 or larger CPU is required in addition to the minimum peripherals listed above. Basic manual: IBM Form GH20-0888.

	Monthly License	Service Class
Basic Courts System (DOS) 5736-G26	$625	B

Bill Processor System, IMS/360 Bridge

Please refer to Production Information Control System (PICS).

Brokerage Accounting System Elements (BASE)

BASE provides more than 75 standardized reports associated with daily security transactions. It accepts daily transaction activity through its Edit program in a random sequence, and edit-checks codes and syntax during input. Files are created and maintained and reports generated in the areas of customer accounts, interest charges, proxy statements, dividends, transfers, clearing programs, floor brokerage accounts, fails, when-issues trade blotters, and a stock record. Generally, COBOL is used for processing programs, Assembler language for subroutines, and RPG for reports.

BASE requires DOS, Assembler D, COBOL, RPG, Tape Sort/Merge, and Disk Sort /Merge. It will operate in a 64K CPU with decimal arithmetic and a selector channel; storage protection is also required for multiprogrammed operation. A console printer-keyboard, card read punch, line printer, four tape units (at least two 9-track), and two 2311 Disk Drives are required. The program requires a 52K partition, and a 10K DOS supervisor is recommended. Basic manual: IBM Form GH20-0789.

	Monthly License	Service Class
Brokerage Accounting System Elements – BASE (DOS) 5736-F31	$800	B

Budget Accounting Information System (BACIS)

BACIS enables public institutions to be more responsive to interested citizens by performing budget reporting, analysis, and administrative functions. It records, accumulates, analyzes, and presents financial data by source, organizational unit, project, fund, function, line-item, or any other user-defined classification. It includes 40 variations of 17 basic reports, and can produce the same report in more than one format.

The program runs with a 6K 2311-resident DOS supervisor, an ANS COBOL subset, the DOS Assembler, DOS Sort/Merge, and two DOS utilities on systems as small as a 48K 360/25. The CPU must have a selector channel, since magnetic tape (two drives) is used. Also required are a printer-keyboard, a card read punch, a printer, and two 2311 Disk Drives. Basic manual: IBM Form GH20-1002.

	Monthly License	Service Class
Budget Accounting Information System – BACIS (DOS) 5736-XT1	$260	B

Budgets and Plans Generator (BUDPLAN)

BUDPLAN is a three-phase program wirtten in PL/1 that creates, processes, and maintains financial plans and corporate models. Phase 1 analyzes users' logic statements and generates routines to be input to phase 3; it also generates overlay structures for phase 3 to conserve memory. Phase 2 interprets print specifications in BUDPLAN and generates print commands for input to phase 3. Phase 3, the budgets and plans generator program, processes control cards against planning data and produces reports according to the user's planning logic and print specifications.

BUDPLAN runs under OS/MFT or MVT and uses BSAM, QSAM, and BDAM access methods. Either the PL/1-F compiler or the PL/1 Optimizing Compiler with Resident Library and Transient Library is required. Also needed are Linkage Editor-F and certain OS utilities. BUDPLAN will run on a 360/40 (192K) or 370/135 (144K) with the universal instruction set. It uses 100K with the PL/1 Optimizing Compiler or 108K with PL/1-F. In addition to OS requirements, two disk drives, a card reader, and a printer are required. Basic manual: IBM Form GH19-1038.

	Monthly License	Service Class
Budget and Plans Generator – BUDPLAN (OS) 5734-F51	$150	B

Business Analysis/BASIC for S/3-6 and S/360-ITF

Business Analysis/BASIC is a set of 30 interactive routines for System/3 Model 6 or System/360 and 370 (OS, DOS, or TSO) written in the BASIC language. It provides procedures for data generation and maintenance, spread-sheet analysis, investment analysis, cost-volume-profit or break-even analysis, depreciation analysis, and time-series analysis. The user need not know a great deal about programming in order to use Business Analysis/BASIC.

On a System/360 or 370, the system to run the program must have the appropriate ITF program product in BASIC or an appropriate feature for PL/1 (see MATH/BASIC). The program's routines require either an ITF user area of 50K bytes and a 10.8K-byte symbol table for short precision arithmetic, or 60K and 16.3K for long precision. Under DOS, a 128K CPU and 100K partition are required; under OS this becomes 256K and 110K; and under TSO, 512K and 190K. On a System/3 Model 6, the minimum 8K BASIC system (5406-B2 CPU, disk, printer) and the BASIC program product are all that is required. Basic manual: IBM Form GH20-1175.

	Monthly License	Service Class
Business Analysis/BASIC (S/370 ITF/DOS, OS, or TSO; S/3-6) 5703-XM3	$50	B

Capacity Planning

Please refer to Production Information Control System (PICS).

CATALIST

CATALIST has the function of aiding conversion to a System/360 from an IBM 1401 CFO-62 system (Consolidated Functions Ordinary, a contract maintenance system for the ordinary life insurance company). It translates CFO-62 1401 Autocoder programs into System/360 Assembler-language programs, retaining original statement labels and comments. Although 90 to 95% of the CATALIST-generated program often requires no modification to obtain a tested program, the amount of manual conversion effort can vary among users, and can be substantial. The converted programs generally require 1.5 to 1.75 times the core storage of the original.

CATALIST is written in PL/1, but a compiler for PL/1 is not required, since execution modules are distributed in object form. The program requires a DOS partition of at least 52K on a 360/30 or larger CPU. Conversion requires a total of three passes. The minimum system includes storage protection, decimal arithmetic, an interval timer, a selector channel, a printer-keyboard, a 2311 Disk Drive, two magnetic tape units, a printer, and a card read punch. Basic manual: IBM Form GH20-0813.

	Monthly License	Service Class
CATALIST (CFO Autocoder to Assembler Language Instruction Set Translator –DOS) 5736-XX2	$1,200	B

Check Processing Control System (CPCS)

CPCS performs processing functions associated with entry, distribution, proof, adjustment, and control of MICR (magnetic ink character recognition) documents in commercial or federal reserve banks. Direct connection of IBM 1419 and (RPQ-type) 2956 MICR reader/sorters connected via RPQ 2947 Model 4 Check Collection Controllers is supported. Significant features within CPCS are document image processing, pass-to-pass control, reject re-entry facilities, on-line balancing and adjustments, and document history files.

CPCS is written in COBOL (ANS) and System/360 Assembler language for an OS/MFT or MVT environment. Most application tasks are coded in COBOL, with system tasks Assembler-coded. The minimum system requires a 360/40 or larger CPU with 256K bytes of storage, decimal arithmetic, a card reader, a printer, normal OS requirements, two 2314-type or 3330 disk modules, two tape units, three 2260 Model 2 displays, and a 1419 or 2956 MICR reader with various optional features. Available 9/73. Basic Manual: IBM Form GH20-4179.

	Monthly License	Service Class
Check Processing Control System (OS) 5734-F11	$1,100	B

Computer System Simulator II (CSS II)

CSS II operates under OS to provide a simulation program that allows analysis of the operation of a computer system. The user writes programs in a CSS II instruction set, in flowchart fashion. The program runs, creating simulated system behavior, which is then described in quantitative output statistics, automatically provided. Job throughput, channel utilization, distribution of response times, and storage use statistics are examples of the output. Multiple processors, block multiplexor channels, and communications lines and terminals can be modeled. CSS II is written entirely in Assembler language.

The minimum system required to run CSS II is a 256K System/360 with double-precision floating-point, interval timer, selector channel, printer-keyboard, card read punch, line printer, and two 2311 Disk Drives. Basic manual: IBM Form GH20-0874.

	Monthly License	Service Class
Computer System Simulator II (CSS II OS) 5734-XX5	$2,500	B

Consolidated Functions Ordinary II (CFO II)

CFO II is a systems approach to maintenance, service, and processing of individual life insurance contracts. It was initially developed for the IBM 1401 in 1962. CFO II will run on systems as small as a 360/20, but is compatible for growth to larger S/360 and 370 systems, being offered in Assembler language for all these systems. It has 16 daily cycle programs to serve small insurance companies with about 15,000 ordinary life insurance policies. Larger companies, those with 20,000 or more policies in force, should examine the Advanced Life Information System (ALIS) instead.

CFO II runs under the Tape or Disk Processing System (TPS or DPS) on a 360/20 or under DOS or OS on a System/360, Model 25 and up, or System/370. It also uses a sort/merge program. On a 360/20, the minimum configuration is a Submodel 5 with 24K bytes of storage, 20K of which is available to the program. A card punch, card reader, printer, I/O channel, and four magnetic tape units are also required. On larger S/360's and 370's, a 22K DOS partition or 30K OS partition or region is required, in addition to decimal arithmetic on the CPU, a card read punch, printer, one disk drive, a printer-keyboard, and four magnetic tape units. Basic manual: IBM Form GH20-0883.

	Monthly License	Service Class
Consolidated Functions Ordinary II – CFO II (360/20 TPS or DPS, S/360 or 370 DOS or OS) 5736-N13	$300	B

Consumer Goods System (COGS)

This system is written in PL/1 and comes in two basic, independent modules–Forecasting and Allocation.

COGS Forecasting applies an adaptive forecasting technique to a minimum of two years of product sales history to produce a short-term (one year or less) forecast. A variety of models are set up by the system to make forecasts based upon the historical data, and the user selects the model that suits him best, using a built-in simulation capability to test the model.

COGS Allocation addresses the distribution/inventory control problems of consumer goods distributors and sets up an order allocation system based upon forecasted product usage and current inventory information by stocking location. COGS Allocation controls the ordering process, sets up orders for the distribution network to satisfy customer requirements, balance on-hand requirements, and economic order quantities (EOQ), using reorder points, shelf life, delivery time, etc.

COGS Forecasting runs in a 102K partition or region under OS/MFT or MVT (or their VS counterparts) or in a 63K partition under DOS or DOS/VS. The minimum machine configuration includes two 2311 Disk Drives and two tape units under DOS (or DOS/VS), or four 2311 Disk Drives and two tape units under OS (or VS).

COGS Allocation runs in a 100K partition or region under OS/MFT or MVT (or their VS counterparts) or a 56K partition under DOS or DOS/VS. The minimum machine configuration includes three 2311 Disk Drives and two tape units under DOS (or DOS/VS), or four 2311 Disk Drives and two tape units under OS (or VS). Basic Allocation manual: IBM Form GH20-0721. Basic Forecasting manual: IBM Form GH20-0722.

	Monthly License	Service Class
COGS – Allocation (OS) 5734-D32	$150	B
COGS – Allocation (DOS) 5736-D31	150	B
COGS – Forecasting (OS) 5734-D33	200	B
COGS – Forecasting (DOS) 5736-D32	200	B

Continuous System Modeling Program III (CSMP III)

CSMP III and its Graphic Feature are designed for the simulation of continuous systems. Continuous systems are dynamic systems whose behavior is best described by differential equations. Such equations are solved by continuous integration–a process that is simple for any analog computer system. Digital computers, however, were unsuitable for performing such solutions until the state of the digital computer art allowed them to achieve speeds that made solutions obtainable in reasonable time frames. CSMP III provides the necessary facilities on a digital computer and also isolates the scientist or engineer from the task of learning digital computer programming by providing a high-level application language of its own.

CSMP III is available in two versions, one with and one without the Graphic Feature. The feature is of particular interest to those using or about to acquire large-scale analog or hybrid analog/digital systems.

An impressive array of functions is available with CSMP III. In addition, the program can use FORTRAN IV statements in either its batch or interactive mode. FORTRAN IV can be developed in specialized and extended forms under CSMP III, but all FORTRAN programs developed under the program must also be executed under CSMP III. The modelling program itself is written in FORTRAN and Assembler, as is the Graphic Feature. Each operates under OS/MFT or MVT. BPAM, BSAM, and QSAM data management facilities are employed by CSMP III.

The minimum basic CSMP III machine configuration requires a 360/40 with 131K bytes of storage and floating point. A 102K-byte partition is required when the FORTRAN IV (G) compiler and 88K-byte Linkage Editor F are used, but execution can be accomplished with less storage. A 2311 or equivalent disk, in addition to the OS residence disk, and an extra tape drive are required for program installation, but not for execution.

In addition to the foregoing, the Graphic Feature requires a 2250 Model 1 or 3 Display Unit with certain special features. A 150K partition is needed to use the aforementioned FORTRAN compiler and Linkage Editor. Basic manual: IBM Form GH19-7000.

Program Products — Applications
IBM Corporation

	Monthly License	Service Class
Continuous System Modeling Program III (CSMP III OS) 5734-XS9	$ 90	B
Graphic Feature 6065, 6066, and 6067	340	B

Coursewriter III/DOS Version 3
Coursewriter III/OS Version 3

Coursewriter III can be used operationally for CAI (Computer Assisted Instruction), educational research, and curriculum development. It provides terminal interfaces to students and other users on 1050, 2260, 2265, 2740, 2741, and Teletype 33/35 terminals. It also supports IBM 3275 and 3277 display terminals, but only with those functions available on the 2260.

Coursewriter III is built upon a CAI experience base that began with Coursewriter I on an IBM 1401 in 1958. Coursewriter II operated on the IBM 1500 Instructional System. A later version, the original Coursewriter III/OS (5734-E12), is still available but is superseded by Coursewriter III, Version 3. A particularly nice feature of the program is that course materials can be stored on removable disk packs.

Coursewriter III requires at least a System/360 or 370 that can support a 40K DOS or 55K OS partition, a console typewriter, a card read punch and printer, a direct-access storage device, and at least one 1050 or 2740 terminal. In addition, the DOS version requires DASD space for files, work areas, and DOS systems libraries. The OS version additionally requires decimal arithmetic, storage protection, and storage space sufficient for the OS supervisor. The actual partition sizes for Coursewriter III Version 3 under both DOS and OS depend on the terminal configuration. Basic manual: IBM Form GH20-0987.

	Monthly License	Service Class
Coursewriter III/DOS Version 3 5736-E11	$200	B
Coursewriter III/OS Version 3 5734-E14	275	B

Customer Information Control System (CICS)

Please refer to Report 70E-491-02 for full information.

DATA/360

DATA/360 is available in two versions—Version I and Version II—that can be run under DOS or OS (or their VS counterparts). DATA/360 is a key-to-disk data entry system that must be run on-line, using the 360 (or 370) CPU as the controlling device for the data collection logic. Data can be entered through 2260 Display Stations and stored on 2311, 2314, or 3330 disk files.

In Version I, a variable number of display stations can be supported under DOS, and up to 48 displays can be supported under OS. Provisions are made for generalized edit procedures and for overall simulation of the IBM 29 Card Punch and 59 Card Verifier. Record lengths can be up to 117 characters, with format control under DOS for up to 16 fields and under OS for up to 24 fields. For both versions, operator/production statistics are accumulated.

In Version II, the 3270 Information Display System can also be used as a data input device. Other enhancements of Version II over Version I include a larger number of supported CRT terminals (up to 96 2260 Model 2's or 128 3277 Model 1's under DOS or OS), support for magnetic tape output, and considerably greater edit flexibility and format control.

Of special interest to DATA/360 users is the more recent availability of VIDEO/370. Please refer to that Program Product description for further information.

DATA/360-I runs in a 24K nucleus partition under DOS, plus 552 bytes for each 2260 CRT; or in a 26K nucleus partition or region under OS, with 1100 additional bytes for each CRT. Version II runs in a 28K nucleus partition under DOS, plus 714 bytes per CRT for a 20-station (240-screen), two-control-unit system; or in a 36K nucleus partition or region under OS, with 1300 additional bytes for each CRT. The minimum machine configuration includes one tape unit under DOS or OS (or their VS counterparts).

Basic DOS Version I manual: IBM Form GH20-0838. Basic OS Version I manual: IBM Form GH20-0853. Basic DOS Version II manual: IBM Form GH20-1037. Basic OS Version II manual: IBM Form GH20-1036.

	Monthly License	Service Class
DATA/360 Version I (DOS) 5736-X52	$ 50	B
DATA/360 Version I (OS) 5734-X53	100	B
DATA/360 Version II (DOS) 5736-X55	125	B
DATA/360 Version II (OS) 5734-X58	125	B

Data Base Organization and Maintenance Processor (DBOMP).

Please refer to Production Information Control System (PICS).

Display Management System (DMS)

DMS is used in conjunction with the operating system and the Customer Information Control System (CICS, see DATAPRO 70 Report 70E-491-02) to implement interactive processing operations which feature the 2260 and/or 2265 Display Stations. It consists of a series of application modules operating under a supervisor. The program acts upon parameters supplied by users on three forms to produce a customized set of display station images. The display images thus created correspond to the user's data files and utilities. Importantly, DMS provides display screen paging. DMS is written in OS Assembler language and operates under CICS. It requires use of the operating system utilities.

The machine configuration for DMS varies in its basic requirements, depending on terminal types used, whether terminals are connected locally or remotely, etc. The system must include, however, a CPU and I/O units necessary to support OS (typical for the system is a 131K CPU for OS/PCP and a 262K CPU for OS/MFT or MVT). Decimal arithmetic, storage protection, and a selector channel are required. A 100K (MFT) or 115K (MVT)

Program Products — Applications
IBM Corporation

➤ partition suffices for up to 10 terminals, 5 of which can be 2260/65's. A tape unit is needed for program maintenance and distribution. Basic manual: IBM Form GH20-0841.

	Monthly License	Service Class
Display Management System (DMS) 5734-XCi	$185	B

Electronic Circuit Analysis Program II (ECAP II)

ECAP II performs direct current (DC) and transient analysis of linear and nonlinear electrical networks. It features detailed user-oriented diagnostic messages as an aid to persons unfamiliar with computers, and can provide both printed and plotted output.

Interestingly, ECAP II is available in versions for the IBM 1130 equipped with as little as 16K words and for the System/360—but the latter requires a 360/65 running under OS. The partition size required for S/360 operation can be as large as 400K. The S/360 version has no teleprocessing capabilities, so the ECAP II user must submit his input as a batch job and wait for his results. The wait can be a long one, since from 170K to 400K bytes of main storage must be available in addition to the storage required by the operating system and PLAN (Program Language Analyzer), plus 2 million bytes of auxiliary storage.

Most ECAP II programs are coded in Fortran H, Version II or 1130 FORTRAN, with a few coded in Assembler language. Basic manual: IBM Form GH20-0983.

	Monthly License	Service Class
Electronic Circuit Analysis Program II (OS) 5734-EE1	$170	B

EPIC: SOCRATES, FAST, Budget/Finance, and Student

EPIC is the name of a group of four program products—SOCRATES, FAST, Budget/Finance, and Student—that provide administrative and financial records for schools from elementary level to universities. Although the four program products and their features are completely independent, when a user installs more than one of the programs, files are shared for common data (e.g., master schedule and student ID files). Thus, the applications enjoy the benefits of a data base approach without the costs of a separate data base program.

SOCRATES is a student scheduling program. Its master schedule generation and post-scheduling maintenance features can help educational institutions efficiently utilize school resources.

FAST is a test scoring program that contains the full content of the widely used 1401-1440 FAST, plus enhancements. It produces test score results and statistical analyses from a wide variety of input.

Budget/Finance is a series of interrelated programs designed to automate construction of school budgets and accounting procedures. It can help to regulate school expenditures. It also provides a data base often vital to the school district during expansion.

Student, whose proper name is Student Records and Grade/Attendance Reporting, provides grade reports and report cards. It also supplies a statistical analysis of grades, honor roll lists, and so on.

All four EPIC program products are written in a subset of COBOL that can be compiled and run on an 1130, System/3 Model 10, or System/360 or 370 under DOS or OS.

The minimum 1130 configuration includes a 16K-word CPU, any 120 or 132-position printer, a CPU-mounted disk (and, nearly always, an additional disk in a 2310), and any COBOL-supported card reader and punch. Additional support is available for more storage, more disks, and the 1231 Optical Mark Reader.

For a System/3 Model 10, the minimal requirements are a 32K-byte CPU with one 5444-type disk, any 120 to 132-position printer, a COBOL-supported card reader and punch, and a console printer-keyboard. Additional support exists for more CPU and disk storage, an OMR, and 1442 card image reading and punching.

A System/360 or 370 DOS or OS system has the following minimum requirements: storage size sufficient for the supervisor plus the COBOL compiler or 22K (whichever is larger), one 2311 Disk Drive beyond that required by the supervisor, any 120 or 132-position printer, a COBOL-supported card reader and punch, and a console printer-keyboard. Additionally, more CPU and disk storage, an OMR, and card column binary or card image modes are supported. Basic manual: IBM Forms GH20-1129 (SOCRATES), GH20-1131 (FAST), GH20-1132 (Budget/Finance), and GH20-1130 (Student).

	Monthly License	Service Class
EPIC: SOCRATES 5735-E91	$175	B
EPIC: FAST 5735-E92	95	B
EPIC: Budget/Finance 5735-E93	110	B
EPIC: Student 5735-E94	80	B

Fare Quote/Ticketing (FQ/TK)

The FQ/TK program is written in Assembler language and comes in one basic module with an optional Tarriff Maintenance Feature. FQ/TK is an ancillary DOS or DOS/VS program, for use under the Programmed Airline Reservation System (PARS) and Airlines Control Program (ACP), that automatically prices and prints standard airline tickets in an on-line environment. The program is designed to increase ticket agent and airline sales agent productivity by mechanizing the fare quotes for inquiries on full fare, family plan, and excursion rates. FQ/TK allows for connecting service between two trip segments separated by a valid connecting point between the origin and destination. Fares are based on the specific connecting point and the local rates of the carrier providing the connecting service.

A Passenger Name Record (PNR) data base used in PARS is also used as the data base for FQ/TK. The basic PARS utilities programs (FACE, RECOUP, etc.) are used with FQ/TK. A copy of the auditor's coupon is produced for use as a revenue accounting input for all automated tickets. The special Tariff Maintenance Feature is used to create and maintain the on-line tariff data base for the Fare Quote/Ticketing program. Basic manual: IBM Form GH20-0873. ➤

	Monthly License	Service Class
FQTK-On-Line Fare Quote/Ticketing (DOS) 5736-T11	$2,800	B
FQTK Tariff Maintenance Feature	5,700	B

Fashion Reporter

The Fashion Reporter provides a comprehensive retailer's inventory control system that produces management reports reflecting the activity and inventory status of merchandise in individual style or price-line categories. Written in PL/1 and RPG, the system can be run to produce either an update or a report output cycle. This lets the system be used daily or oftener to process transactions against the inventory file, while allowing output report to be prepared weekly or on any other periodic basis specified by the user.

Specific output reports include: (1) a Class/Priceline Sales Summary of styles grouped at the class level, showing weekly sales for the last six weeks, sales to date, on-order, on-hand, and in-sight class totals; (2) a Style Page showing a complete picture of a given style at any level, including all styles in a full department; (3) a Pre-Age Report that extracts and formats inventory records by department and class, with varying aging criteria; and (4) an Aging Report showing current on-hand balances by store distributed into user-specified aging periods; the dollar value of the quantities is calculated and accumulated, with class and department totals broken down by percentages for each aging period.

Fashion Reporter runs in a 24K partition under DOS or DOS/VS. The minimum machine configuration includes one disk drive and four tape units. Basic manual: IBM Form GH20-0935.

	Monthly License	Service Class
Fashion Reporter (DOS) 5736-D11	$165	B

Filing and Source Data Entry Techniques for Easier Retrieval (FASTER)

FASTER is available in a Low Core (LC) version for DOS and in Multitasking/multithreading (MT) versions for both DOS and OS. Essentially, FASTER is an Assembly-language data communications monitor intended primarily for financial industry applications using ISAM files. With the announcement of CICS/VS, the FASTER family of products has largely been superseded (see Report 70E-491-02).

The entry-level LC version is a single-thread monitor that provides support to 1050, 2740 Model 1, 2260, and 2265 terminals for line control, communications interface, and message processing functions.

The MT versions of FASTER are intended to provide growth paths for FASTER LC users, and essentially permit greater throughput by multitasking the user's application programs. Specific functional enhancements of the MT versions include 3270 Information Display System support, up to 7 concurrent transactions in DOS or 13 under OS, 15 priority levels, ISAM and DAM file support, direct linkage for Assembler and COBOL languages, improved audit trail data recording, etc.

FASTER LC runs in a 36K minimum partition size under DOS, while FASTER MT runs in an 80K minimum partition under DOS or in a 120K minimum partition or region under OS. In all cases, adequate disk storage is required for the associated files. Basic LC manual: IBM Form GH20-0810. Basic MT (DOS) Manual: IBM Form GH20-0903. Basic MT (OS) Manual: IBM Form GH20-1031.

	Monthly License	Service Class
FASTER-LC (DOS) 5736-G22	$100	B
FASTER-MT (DOS) 5736-G24	285	B
FASTER-MT (OS) 5734-G21	450	B

Financial Terminal System (FTS)

FTS provides a BTAM data communications interface to DOS data bases for user-written financial industry applications programs. The primary terminal devices supported are the 2740 typewriter-style terminal, the 2260 CRT, and the 2972 and 2980 banking terminals. Specific services provided by FTS include system error handling, master terminal support, 2260 paging, terminal network simulation for program development, system performance recording, terminal-status inquiry and orderly shutdown procedures. These services are provided by FTS for subtasks, terminals, storage, and individual applications programs.

FTS runs in a 48 K partition under DOS or DOS/VS. The minimum machine configuration includes enough disk storage space for data base requirements and work areas, and one tape unit. Basic Manual: IBM Form GH20-0763.

	Monthly License	Service Class
Financial Terminal System – FTS (DOS) 5736-F12	$400	B

Generalized Information System, Version 2 (GIS/2)

The GIS/2 system is written in Assembler language and has one basic module with 11 optional features. As a sort of "super-RPG," GIS/2 provides a generalized high-level language to handle complex multi-file data requests with extensive logic and computational requirements. Data requests can be entered directly into the computer either on a batch basis or via a remote terminal. GIS/2 provides sets of generalized routines which enable data set creation, maintenance, and retrieval.

Basic GIS/2 is intended for retrieval functions on a batch basis against up to three SAM or ISAM OS data sets simultaneously. The 11 optional features provide for specific output reports, more complex conditional search or procedural logic, a GIS procedure library, a file modification capability, arithmetic statements, count/sum/average statements, 16-file (or data set) support, teleprocessing support, a hierarchical file capability, file restructure/create, and edit/encode.

The GIS/2 user describes his new or existing files in a procedure that does not involve detailed programming. These file descriptions enable the user to address the contents of his files by means of symbolic names. GIS/2 procedures specified by the user can link to non-GIS routines referenced in the procedure specification. User

Program Products — Applications
IBM Corporation

routines written in other languages, such as PL/1 or COBOL, can have limited access to GIS/2 files. Access restrictions can be placed on sensitive data. An error trace capability provides selective recording of the occurrence of processing errors and the alternative actions to be taken.

Of particular interest is the GIS/2 DL/1 Query Support Feature. Used in conjunction with IMS-2 (requires 40K bytes), this feature allows GIS/2 operations to be performed upon IMS-type data bases as well as SAM and ISAM files. An additional 10K bytes is required for this interface.

An earlier version of GIS—available only as a full monolithic system with no options—has been superseded by GIS/2 but remains available. Improvements of GIS/2 over the earlier version include a throughput improvement of 2 to 3 times, HASP compatability for data communications support under MFT, and several internal data management improvements that give GIS/2 greater flexibility.

Basic GIS/2 runs in a 124 K partition or region under OS/MFT or MVT (or their VS counterparts). The minimum machine configuration includes three disk drives and one tape unit. Basic manual: IBM Form GH20-0892.

	Monthly License	Service Class
GIS/1 (OS) 5736-CX1	$1,500	B
GIS/2 (OS) 5734-XX1	450-1,500	B

General-Purpose Simulation System (GPSS)

GPSS is available in two main versions for either DOS or OS (or their VS counterparts). Written in Assembler language, both basic GPSS versions are designed to provide a general-purpose simulation language for modeling and examining systems in the engineering and management science domains. Facilities are provided to alter the application model so that optimal conditions can be identified, bottlenecks isolated, and the effects of delays determined. All parameters used by GPSS are allowed to vary with time. The user typically prepares a logical and statistical model of his system, using block diagrams.

GPSS V is a major functional enhancement over the earlier GPSS Version 2. Specific enhancements include a PL/1 interface, increased numbers of parameters for each transaction, byte- and single-precision floating point save values and matrices, free-form coding, extended indirect addressing capability, etc.

Either the OS or DOS versions of GPSS run in partitions or regions under OS or DOS (or their VS counterparts) in from 52K or 62K to 170K or 178K bytes for Version 2 or V, respectively. The minimum machine configurations are those for the standard operating systems; no additional disk or tape units are required for GPSS operation. Basic GPSS Version 2 manual: IBM Form GH20-4077. Basic GPSS V manual: IBM Form GH20-0826.

	Monthly License	Service Class
GPSS Version 2 (DOS) 5736-XS 1	$20	C
GPSS Version 2 (OS) 5734-XS 1	20	B
GPSS V (DOS) 5736-X53	55	B
GPSS V (OS) 5734-X52	55	B

Graphic Analysis of Three-Dimensional Data (GATD)

GATD is an interactive on-line graphic analysis program written in FORTRAN. It uses the 2250 Graphic Display Terminal to permit three-dimensional data analysis by geologists, engineers, meteorologists, medical researchers, and others concerned with spatial, physical relationships. The user specifies a numerical model of a three-dimensional surface through the Problem Language Analyzer (PLAN) language and interacts by means of the 2250's light pen to define or modify the mathematical surface model, make logical decisions based upon the display, and build contour maps, perspective diagrams, cross-sections, and "fence" diagrams.

GATD runs in a 150K partition or region under OS/MFT or MVT (or their VS counterparts). The minimum machine configuration includes two 2311 Disk Drives and up to five tape units. Basic manual: IBM Form GH20-0539.

	Monthly License	Service Class
GATD—Graphic Analysis of Three-Dimensional Data (OS) 5734-XX2	$300	C

Health Care Support/Accounting System
Health Care Support/Data Communications Program

Please refer to CICS, Report 70E-491-02.

Health Care Support/Electrocardiogram Analysis

HCS/ECG Analysis provides batch processing of electrocardiograms from user-provided digital tape. The tape must be in a special format, and data must be collected three leads at a time. Data is then processed four times to produce a 12-lead scalar analysis. In addition, a set of Frank orthogonal (XYZ) vectorcardiographic leads is collected for rhythm interpretation. The program has interfaces to pass charge information to either the Shared Hospital Accounting System or the Health Care Support/Accounting System that runs under the Customer Information and Control System (CICS).

HCS/ECG Analysis is written in PL/1 and Assembler language and operates as a batch job under the DOS or OS real or virtual storage operating systems. The DOS version requires a 70K DOS or 95K OS partition on a 360/40, 370/125, or larger CPU. Under DOS, the DOS PL/1 Optimizing Compiler is required; under OS, the PL/1-F Compiler is used. Also required is disk work space and (under OS) two disks for compiling and link-editing or (under DOS) two magnetic tape drives for compilation. Basic manual: IBM Form GH20-1249.

	Monthly License	Service Class
Health Care Support/Electrocardiogram (ECG) Analysis; DOS – 5736-H15, or OS – 5734-H11	$350	B

Information Management System (IMS)

Please refer to Report 70E-491-01 for full information.

Program Products — Applications
IBM Corporation

➤ Interactive Training System and Course Structuring Feature

Please refer to IMS or CICS, Report 70E-491-01 or 70E-491-02, respectively.

Inventory Control

Please refer to Production Information Control System (PICS).

Law Enforcement Manpower Resource Allocation (LEMRAS)

LEMRAS is a police administration system that distributes patrols to answer calls for service. As requirements for police patrol visits are received, they are matched by this Assembler and FORTRAN IV language system against available resources, taking into account routine and other administrative requirements. With every dispatch, a history file of call activity is built up on a geographical basis by time, thus allowing LEMRAS to forecast patrol requirements and making it possible for police administrators to ready a sufficiency of resources in advance. Geographical statistics are kept for areas and street intersections, event recording is made for up to 20 types of events, and time records can be kept to the hour. Individual police work forces can consist of from 2 to 25 mobile units, up to 10 districts or precincts can be defined, and up to 10,000 blocks and 10,000 intersections can be referenced.

In addition to the basic task of allocating resources to calls, LEMRAS does forecasting through use of an exponential smoothing technique that weighs more heavily the most recent events. Other functions include reports on system statistics and statistical summaries of call delays.

LEMRAS runs on a minimum 65K DOS system and requires at least one 2311 Disk Drive. Basic manual: IBM Form GH20-0629.

	Monthly License	Service Class
LEMRAS (DOS) 5736-G21	$350	C

LEARN ATS-OS and LEARN ATS-DOS

LEARN ATS is available in OS and DOS versions. It assists users in learning the ATS/360 (Administrative Terminal System/360) language for use in source data entry, file maintenance, and text processing applications. The announced OS version runs only under Version 1 Level 1 of ATS/360 OS. The announced DOS version runs only under Version 1 Level 2 of ATS/360 DOS.

LEARN ATS has three major components: additional and modified macros for accessing ATS/360 from a terminal, eight lessons which present the instruction and are non-alterable ATS/360 special permanent storage documents as well, and a student workbook. Every ATS/360 user has an ongoing educational program, and LEARN ATS can be used at every installed ATS location. LEARN ATS is written in either OS or DOS Assembler. Under DOS, LEARN ATS uses the console interrupt key to call for cessation of ATS/360 peripheral functions. Therefore, DOS background programs that use the console interrupt key or operator-initiated messages cannot run while ATS/360 is running. Otherwise, multiprogramming is fully supported.

LEARN ATS-OS runs under control of the OS version of ATS/360, and thus requires the identical system configuration. Additionally, 300 bytes are added to the ATS/360-OS supervisor, and five tracks of 2311 or four tracks of 2314 disk storage are required. The new programs under ATS conform to the ATS standard of requiring less than 8K. ATS/360-OS allows up to 60 application programs. The minimum ATS/360-OS configuration is the OS requirements plus 22,528 bytes of dynamic storage plus space for QSAM (queued sequential access method), a multiplexer and a selector channel, a console printer-keyboard, two 2311 or one 2314 Disk Drive, a 270X communications controller, and a 2741 Communications Terminal with a Courier 72-type print ball. One magnetic tape drive is needed for system generation.

The basic requirements for DOS are similar, with the minimum CPU storage set at 64K; but that small size could preclude use of some ATS/360 DOS document transmission functions. Basic manuals: IBM Forms GH20-0745-1 (OS) and GH20-0746-1 (DOS).

	Monthly License	Service Class
LEARN ATS-OS 5734-XX8	$40	B
LEARN ATS-DOS 5736-XX3	40	B

MATH/BASIC for System/3 and ITF/360

MATH/BASIC is a set of conversational routines for use by persons not oriented toward data processing to obtain solutions to the most frequently encountered mathematical problems in science and industry. It consists of 40 routines that provide computing capabilities in the areas of: linear equations and matrix eigenvalue problems, zeros or polynomials and zeros and minima or functions, quadrature and differentation, interpolation, approximation, smoothing, ordinary differential equations, fast Fourier transforms, and special functions. No separate educational course is required; instructions are imbedded in the routines and the program reference manual.

MATH/BASIC is written in the BASIC language. On a System/3 Model 6, it will run in the minimum Basic problem-solving system, an 8K 5406 CPU with one disk drive and one 5213 Printer, plus the System/3 BASIC Program Product (5703-XM1).

On a System/360, MATH/BASIC runs under OS, DOS, or TSO (Time-Sharing Option) operating systems that have ITF (Interactive Terminal Facility) BASIC installed, or under ITF-PL/1 if the BASIC programming language feature is installed.

Under ITF on a System/360, MATH/BASIC requires an ITF user area of 50K bytes. Short-precision floating-point arithmetic requires a symbol table of 9K bytes, while long precision requires 16,300 bytes. Also, space must be provided in the ITF library to store the MATH/BASIC routines. All terminals supported by ITF are supported by MATH/BASIC. The following table is a general indicator of required memory under different operating systems:

Operating System	Minimum CPU Storage	Minimum Partition
ITF-OS	256K	110K
ITF-DOS	128K	100K
ITF-TSO	512K	190K ➤

➤ Basic manual: IBM Form GH20-4212.

	Monthly License	Service Class
MATH/BASIC 5734-XM8	$45	B

Mathematical Programming System Extended (MPSX)

MPSX expands the capabilities of the earlier Mathematical Programming System (360A-CO-14X, a Type II program). The optional Mixed Integer Programming Feature (MIP) of MPSX provides the ability to solve mixed integer linear programming problems, and the optional Generalized Upper Bounding (GUB) feature provides the ability to solve large, specially structured linear programming problems in a particularly efficient manner.

MPSX contains all the capabilities that are available with MPS/360: a control program containing its own language and compiler; a set of linear and separable programming procedures; a FORTRAN interface called Read Communications Format; access to the MARVEL program which contains a language processor for matrix generation, output analysis, and report writing; and an interface to the report generator program, MPSRG. MPSX improvements include substantially faster execution time (about two times faster on the average), an expanded problem size of 16K rows, simplified output filing, and an OPTIMIZE macro.

MPSX is written in Assembler language, uses the BSAM and EXCP data management methods, and runs under all currently supported options of OS/360. Minimum region size required by MPSX is 86K. Direct-access storage of the MPSX procedures requires approximately eight cylinders on a 2314 or an equivalent amount of space on other direct-access devices.

The problem size that can be solved by MPSX depends on the amount of CPU storage available. This storage is considered in two categories, program and data storage. The actual size of the MPSX program is about 40K. Large problems and use of the available options to their fullest extent can result in data storage requirements of up to 934K. Also, special files, in addition to system files, are required by the program and its options. Special files in addition to the required ones will further enhance performance, especially when some of them are on direct-access devices. Basic manuals: IBM Forms GH20-0849 and GE20-0350.

	Monthly License	Service Class
Mathematical Programming System Extended (OS) 5734-XM4	$130	B
Mixed Integer Programming feature (OS) 6009, 6010, and 6011	225	B
Generalized Upper Bounding feature (OS) 6059, 6060, and 6061	650	B

Matrix Generator and Report Writer (MGRW)

MGRW is a program specifically developed for use with MPSX (Mathematical Programming System Extended), discussed above, and runs as a procedure under that program. It is an array-oriented system that simplifies the user's task in generating mathematical models. It handles arrays whose rows and columns generally have mnemonic significance in their names. MGRW also has facilities for name manipulation, pictorial report writing, table maintenance, and analysis of linear programming solutions. It uses a list-driven search concept to eliminate constant searching of names in order to locate data.

MGRW can be assembled using the OS/360 F-level Assembler. It uses BSAM and QSAM data management facilities. The OS utility program IEHMOVE is required for system generation, and the F-level Linkage Editor is also required. The system configuration requirements are those for MPSX, plus some additions. MGRW uses a 190K partition together with MPSX Version 1 and shares the MPSX direct-access storage device. The program also can use two of its own special files and up to four separate sort files. Basic manual: IBM Form GH19-5012.

	Monthly License	Service Class
Matrix Generator and Report Writer (for MPXS) 5734-XMC	$400	B

Medical Information Systems Program (MISP)

MISP provides a data communications interface between user-written application programs and the DOS/360 operating system. Using the program, distributed in Assembler-language source language and macros, the system configuration and the macros for communication are generated in accordance with the desired environment. Optionally, BTAM serviceability features can be added.

Installation of MISP requires generation of a DOS supervisor to allow for operator communications, etc. MISP operates as one task in one DOS partition, and can operate in a dedicated environment or in one partition of a multiprogramming environment.

The minimum configuration requirements for MISP are similar to those for DOS, plus additional core, the decimal arithmetic feature on the CPU, an interval timer, and extra direct-access storage space to contain the application data sets. In addition, supported terminals on nonswitched lines or a combination of 1050's and 2740's are used. MISP's basic executive modules use 23.2K bytes of storage, and a system of about ten 2740 terminals, ten data sets, and about 50 programs would require approximately 6.5K bytes for tables and work areas. Basic manual: IBM Form GH20-0806.

	Monthly License	Service Class
Medical Information Systems Program (DOS) 5736-H11	$50	B

MINIPERT

MINIPERT is an interactive program that uses the Critical Path Method (CPM) to assist in the management of labor, machines, material, time, and money. It does this by treating scheduling problems as a network of serial and parallel interrelated activities. The longest time path through this network is called the critical path; all other paths provide "slack" or "safety" time. Proper management will schedule the critical and non-critical tasks to shorten the critical path as much as possible. ➤

Program Products — Applications
IBM Corporation

MINIPERT is written in the APL language provided by the APL/360 Program Products. No additional requirements beyond those for installed APL/360 apply in order to install MINIPERT. APL is available in DOS (5736-XM6, $275 per month, service class A) and OS (5734-XM6, $400 per month, service class A) versions. Basic manual: IBM Form GH20-0852.

	Monthly License	Service Class
MINIPERT (APL) 5734-XP3	$150	B

Order Allocation System

This is a set of programs and related documentation that assists apparel and shoe manufacturers in assigning or allocating inventory against open orders for shipment decisions. Input to the system consists of the order and inventory files; output is picking documents, an updated inventory, listings, reports, and control totals. A key input parameter is the user's rules for allocating shipments. The functional modules within the Order Allocation System are the following programs: order selection, condense inventory, allocation, shipping, print picking slips, adjustment (adjusts for variances between picking slips and orders actually shipped), update orders, expand inventory, and order report.

The Order Allocation System is written in DOS/360 Assembler language. Its generation requires use of the DOS supervisor, system control and basic IOCS, consecutive disk IOCS, direct access method, unit record and disk utilities, disk sort/merge, and Assembler. The system can run on a System/360 Model 22 or larger with 24K bytes of CPU storage (6K supervisor and 18K partition), card reader, card punch, printer, two 2311 Disk Drives (the system uses about 20 cylinders in addition to system residence requirements), and console printer-keyboard. Basic manual: IBM Form GH20-0604.

	Monthly License	Service Class
Order Allocation System (DOS) 5736-D41	$125	B

Planning Systems Generator II (PSG II)—DOS
Planning Systems Generator II (PGS II)—OS

PSG II, in each of its two versions, gives users the means to produce and evaluate a variety of business and financial plans. Users of PSG II have a set of related logic and printing specifications. The planning logic specifications consist of FORTRAN statements, and PSG II controls the related I/O operations. It also provides a library of "macros," or FORTRAN subroutines that are accessed by means of FORTRAN CALL's followed by lists of parameters.

PSG II is written mostly in FORTRAN IV, with some modules in Assembler language. PSG II/DOS runs under DOS in the background partition. PSG II/OS runs under OS/MFT or MVT. The FORTRAN IV (G-level for OS) compiler must be in residence. PSG II/DOS uses the ISAM access method; PSG II/OS uses QSAM, SAM, and BPAM, and also makes use of four OS utilities.

The minimum system configuration for PSG II is a System/360 with 131K (196K for the OS version) bytes of CPU storage and the universal instruction set, a card reader and punch, and one disk or five magnetic tape units. If the system output is via a printer, it must have 132 columns. PSG II/DOS can execute in a 100K-byte partition of main storage, 10K of which is allocated for the user's planning logic. With the same amount of storage set aside for planning logic and 15K for access methods and control blocks, PSG II/OS will execute in a 130K partition. When the permanent file facility is used, the OS partition should be enlarged by about 18K. None of the foregoing estimates includes space for the operating system's storage requirements. Basic manuals: IBM Forms GH20-1035 (PSG II/OS) and GH20-4202 (PSG II/DOS).

	Monthly License	Service Class
Planning System Generator II/OS (PSG II) 5734-XT1	$80	B
Planning System Generator II/DOS (PSG II) 5736-XT1	80	B

Power System Planning (PSP)
Power Flow and Output Capacity Feature
Short Circuit R_0 Feature

PSP allows electrical engineers to obtain solutions to electrical network problems. Its main module, Engineering Data Management Service (EDMS), provides facilities to validate, organize, and load data into a master electrical power network data bank for use by three simultaneous program components. The components simulate steady-state power line flows and station conditions (Power Flow and Output Capacity Feature), three-phase and single-phase line-to-ground faults (Short Circuit R_0 Feature), and transient analysis of synchronous machine swings during electrical system disturbances (Transient Stability). The first two of these components are separately charged-for features for the PSP main program. However, PSP itself includes basic Power Flow and Short Circuit components; the optional features simply extend and enhance those facilities. PSP and its features are written as PL/1-F and operate under OS/MFT or MVT. The BDAM, BISAM, QISAM, and QSAM access methods are used.

It is difficult to predict required system sizes for PSP and its features because the size depends on the PSP components and/or features used and the sizes of the problems to be solved. As an example, the MFT partition sizes for a 1,000-bus problem (700 for transient stability) are: EDMS Data Validation and File Load—130K, EDMS Data Retrieval—130K, EDMS Maintenance—145K, Power Flow or Power Flow and Output Capacity Feature—195K, Short Circuit or Short Circuit R_0 Feature—190K, and Transient Stability—220K. Also required are the universal instruction set, two 2311 Disk Drives, a 132-column printer, and a tape unit for program maintenance. Another tape for file backup is recommended. The foregoing partition sizes are all slightly larger under OS/MVT, by as little as 4K to as much as 15K bytes. The estimates do not include the operating system requirements. Basic manual: IBM Form GH20-0532.

	Monthly License	Service Class
Power System Planning (OS) 5736-U12	$300	B
Power Flow Output and Capacity Feature for PSP 6011, 6012, or 6013	90	B
Short Circuit R_0 Feature for PSP 6014, 6015, or 6016	75	B

Program Products — Applications
IBM Corporation

➤ Procedure Library—Mathematics

This product provides a set of computational procedures intended to help users develop PL/1 procedure libraries. It evolved from an IBM type II program, PL/1 Scientific Subroutine Package (360A-CM-07X). About 40% of the program product consists of procedures taken from its forerunner, modified according to new conventions. Another 20% of the library routines are similar to those in SSP, with coding revised to take advantage of newer algorithms. The remainder of the routines are new, representing functional capabilities that do not exist under SSP.

The procedures are written in PL/1 (F-level), and can be compiled using the OS PL/1-F compiler or the OS PL/1 Optimizing Compiler. Configuration requirements are those for PL/1. Basic manual: IBM Form GH20-0854.

	Monthly License	Service Class
Procedure Library– Mathematics (OS) 5734-XM3	$100	B

Production Information Control System (PICS)

PICS consists of four fundamental program systems—Requirements Planning, Capacity Planning, Shop Floor Control, and Inventory Control—plus two ancillary systems—the Data Base Organization and Maintenance Processor (DBOMP) and the Bill Processor System, IMS/360 Bridge. Each of these systems is treated individually below. Basic PICS manual: IBM Form GE20-0280.

Requirements Planning

Requirements Planning is a manufacturing-oriented system that is functionally similar in many respects to the Bill of Material Processor (BMP). This Assembler-language system is a major component of PICS and comes in a fullfledged version for OS with an optional feature for data base interrelation to handle multiple order files. A DOS Requirements Planning Interface to BMP Item Master and Product Structure files is also available.

The full Requirements Planning system includes a Chained File Management System (CFMS) for data base management. The CFMS data base contains item master, product structure, routing, and work center data.

Input to the Requirements Planning system includes gross requirements by shop day, calendar date, or time period generated manually from forecasts or customer orders, or automatically by the PICS Inventory Control System. Outputs include purchase orders for materials offset in consideration to shipment lead times, demand projections based upon shrinkage factors, seasonal demand history, safety stack levels, etc. Considerable flexibility is provided to handle special management considerations, including planned order adjustments, etc.

The DOS Requirements Planning Interface performs the basic time-series planning to determine planned orders based upon forecasts and orders, but uses the DBOMP data base manager instead of the self-contained CFMS data base in the full-scale Requirements Planning System.

Full Requirements Planning runs in an 80K minimum partition or region under OS, while the DOS subset of Requirements Planning runs in a 36K minimum partition. At least two 2311 Disk Drives are required. Basic DOS version manual: IBM Form GH20-0487. Basic OS version manual: IBM Form GH20-0751.

	Monthly License	Service Class
Requirements Planning Interface (DOS) 5736-M13	$25	B
Full Requirements Planning (OS) 5734-M51	200	B
Full Requirements Planning Special Feature	30	B

Capacity Planning

Capacity Planning comes in two basic versions, Finite Loading and Infinite Loading, each of which runs under DOS or OS as a major component of PICS. These PL/1 programs are used in a production or manufacturing environment to determine labor and machine requirements. An infinite loading capability, available at both levels, matches unlimited work requirements against production capability, resulting in an open-ended work schedule.

The finite versions take Earliest Start Dates (ESD), Latest Start Dates (LSD), and order networks into account, along with determination of priorities, to give a load leveling report. Load leveling is used to determine work start dates consistent with capacity and availability of raw material or components. The DOS capacity planning programs use the Requirements Planning Interface to BMP data bases, while the OS capacity planning programs use the CFMS data base produced by the full-scale Requirements Planning System.

The Infinite system runs in a 40K partition under DOS or an 80K partition under OS, while the Finite system runs in a 20K partition under DOS or an 80K partition under OS. Sufficient disk storage space for the data base and work spaces are required; typically, at least two 2311 Disk Drives are used. Basic manual: IBM Form GH20-0627.

	Monthly License	Service Class
Capacity Planning–Infinite Loading (DOS) 5736-M11	$75	B
Capacity Planning–Finite Loading (DOS) 5736-M12	225	B
Capacity Planning–Infinite Loading (OS) 5734-M53	100	B
Capacity Planning–Finite Loading (OS) 5734-M54	275	B

Shop Floor Control

The Shop Floor Control system is intended for use in environments where finished products are fabricated/assembled. This Assembler-language system is an integral part of PICS and develops a work list, including open order release/maintenance. Input consists of data prepared by the Capacity Planning system and project requirements developed by the Inventory Control system. The DOS version uses the BMP data base, while the OS version uses the CFMS data base manager in the Requirements Planning system.

Specific functions provided by the Shop Floor Control System include component availability checking of orders ➤

subject to release, a material shortage report of orders that cannot be released due to insufficient availability of material, open order splitting, and three order priority calculations for preparation of a work list, including calculations for order due date, shortest operation first, and slack time for remaining operations.

The Shop Floor Control System runs in a 90K to 100K partition or region under OS/MFT or MVT (or their VS counterparts) or in a 26K partition under DOS or DOS/VS. The minimum machine configuration includes at least two disk drives. Basic manual: IBM Form GH20-0753.

	Monthly License	Service Class
Shop Floor Control (DOS) 5736-M31	$100	B
Shop Floor Control (OS) 5734-M31	155	B

Inventory Control

This Assembler-language system is designed to handle a comprehensive inventory control environment under OS, with emphasis upon reorder point control and requirements forecasting based upon demand history analysis. As a major component of PICS, the Inventory Control System consists of: seven reorder point management programs to evaluate safety stock levels using any of seven calculation techniques; provision for order action notice based upon stock levels and calculated reorder points; economic order quantity considerations based upon averages, future requirements, etc.; and a variety of other functions, including user exits for in-house special processing.

Inventory Control runs in a 44K minimum partition under OS/MFT or MVT. Basic manual: IBM Form GH20-0752.

	Monthly License	Service Class
Inventory Control (OS) 5734-M52	$175	B

Data Base Organization and Maintenance Processor (DBOMP)

DBOMP, an Assembler-language outgrowth of the earlier, no-charge Bill of Material Processor (BMP), provides a relatively primitive data base capability intended primarily for manufacturing environments where limited logical relationships need to be expressed. Generally, DBOMP is useful for bill of materials applications where a part must be related to a major assembly or a major assembly must be "exploded" to subassemblies or detailed components. In these basic cross-reference environments on a small system, DBOMP's chained-file data structures are most satisfactory. For data communications requirements, a DBOMP feature providing an interface to CICS is available.

DBOMP runs in a 24K minimum partition under DOS or DOS/VS, with at least two disk drives. Basic manual: IBM Form GH20-4217.

	Monthly License	Service Class
DBOMP (DOS) 5736-XX4	$100	B
DBOMP CICS Interface Feature	75	B

Bill Processor Systems—IMS/360 Bridge

This Assembler-language utility program reformats data bases built under the Bill of Material Processor (no-charge program 360A-ME-06X), DBOMP (Program Product 5736-XX4), or Requirements Planning's (Program Product 5734-M51) CFMS into IMS HIDAM data bases. The dumping of the Bill Processor data base can be done in a standalone mode of operation; the restructuring into an IMS data base must be done under IMS (Report 70C-491-01).

This system runs in an OS partition or region of 58K bytes for DOS files or 84K bytes for OS files. Basic manual: IBM Form GH20-0961.

	Monthly License	Service Class
Bill Processor Systems—IMS/360 Bridge (OS) 5734-XX9	$150	B

Project Management System (PMS)

PMS comes in 2 major versions—PMS/360 Version II, and PMS IV. PMS/360 provides a series of PERT-like project management aids written in Assembler language for complex research, engineering, and fabrication projects. These aids are especially useful for determining the probable schedule and cost impact of proposed plan changes.

In addition to the aerospace environment for which PMS/360 was originally developed, add-on modules are available to support applications in manufacturing and distribution, scientific institutions, service industries, etc. These modules are: (1) the Network Processor (including CPM techniques for up to 254 subnets of 1,000 to 32,000 activity steps each); hourly, daily, weekly, or monthly time periods; nine milestone segments, etc.); (2) the Cost Processor (including an accounting calendar for variable cost period reporting with rate tables, budgets, actual costs, estimates, commitments, and obligations for up to 32,000 charge numbers at nine levels of work breakdown); (3) the Report Processor for output preparation; and (4) the Resource Allocation Processor to schedule the start date of each activity based upon task duration times and required completion dates.

PMS IV is an enhancement of PMS/360 that makes the three modular processors in PMS/360 extra-cost options. Specific enhancements include improved performance; dynamic alteration of sort, paging, and report format parameters at execution time; precedence network capability; larger number of networks (254); non-linear cost spreading; user-defined default values for rates; and generally larger tables for more detail entries per summary item.

Either version of PMS with the Network Processor runs in the following main memory sizes with the indicated associated disk requirements under OS/MFT or MVT (or their VS counterparts):

No. of Subnet Activities	Main Memory	Disk Storage*
1,000	44K	700K
3,560	108K	2,492K
8,680	236K	6,076K
18,920	492K	13,244K
32,000	1,004K	22,400K

*Assuming a disk requirement of 700 bytes per activity.

Program Products — Applications
IBM Corporation

With only the basic Report Processor, PMS IV requires a 44K-byte partition or region; with the Resource Allocation module added, PMS IV requires a 75K-byte partition or region. At least two 2311 Disk Drives and a Tape Unit are required by either version. Basic PMS/360 manual: IBM Form GH20-0690. Basic PMS IV manual: IBM Form GH20-0855.

	Monthly License	Service Class
PMS/360 (OS) 5734-XP1	$300	B
PMS IV with Report Processor (OS) 5734-XP4	50	B
PMS IV Network Processor Option	50	B
PMS IV Cost Processor Option	50	B
PMS IV Resource Allocation Processor Option	200	B

Property and Liability Information System (PALIS)

PALIS is designed as the foundation of a property and liability insurance information system. It is a set of seven programs that serve insurance companies by creating and maintaining a policy-in-force file. PALIS programs are all compatible with one another, and all run under DOS. All insurance logic in the programs is written in COBOL, and all control system modules are written in Assembler. The PALIS programs are: Basic Program, Automobile Program, Other Lines (of commercial insurance) Program, Additional Functions (new processing facilities based on the 1968 MLIRB plan), Additional File Facility (file maintenance on 2319/2314 Disks, 3330 Disks, or 2321 Data Cells), Automobile Enhancements (continuous policy renewal options using only one physical master record and ability to process multiple policy changes at once), and Homeowner's Enhancements (also extensions based on the 1968 MLIRB plan, plus some other features). The first three PALIS programs are Type II routines, available at no charge.

PALIS uses the following DOS components: supervisor or system control and basic IOCS, DAM macros, consecutive tape IOCS, Assembler, Report Program Generator, COBOL, disk-resident tape sort/merge, and unit record, disk, and magnetic tape utilities. Also, QTAM is used for inquiry, and data cell utilities are needed if files are on the 2321. The System/360 Flowchart program is used to generate PALIS flowcharts.

A typical PALIS system might include a 148K to 196K 370/135 CPU with console printer-keyboard, card reader and punch, 2000-lpm printer, three 120KB tape units, and two 3330 Disk Drives. Additional inquiry features might include two 2260 or 1052 terminals on the system. The message control program requires 24K bytes nominally, 50K bytes for Automobile or Homeowner's, and 22K additional bytes for Other Lines. Configuration requirements, however, can vary widely from these guidelines. Basic manual: IBM Form GH20-0283.

	Monthly License	Service Class
PALIS Basic Program 360A-IF-10X	No charge	B
PALIS Automobile Program 360A-IF-11X	No charge	B
PALIS Other Lines Program 360A-IF-13X	No charge	B
PALIS Additional Functions 5736-N21	$300	B
PALIS Additional File Facility 5736-N22	250	B
PALIS Automobile Enhancements 5736-N24	225	B
PALIS Homeowner's Enhancements 5736-N25	225	B

Requirements Planning

Please refer to Production Information Control System (PICS), above.

REsource ALlocation for Project Control (REAL/360)

REAL/360 can precisely allocate project resources on a short-range basis. It is an extension of an earlier Type II IBM program, Project Control System/360. REAL/360 schedules by means of the critical path method (see MINIPERT). It is written in FORTRAN IV (basic level) and Assembler, and runs under DOS. It requires a 53K partition on a 64K 360/30 or larger CPU, floating-point arithmetic, two discs, a card reader and punch, a printer, and a console printer-keyboard. Basic manual: IBM Form GH19-0014.

	Monthly Licence	Service Class
Resource Allocation for Project Control–REAL/360 (DOS) 5736-XP2	$170	C

Rigid Frame Selection Program (RFSP)

RFSP is a structural analysis and design system, written in FORTRAN, that provides a method of determining least-weight designs for members of different types of structures. Material requirements, optimization reports, member segmentation, analysis, and design calculations, and a cutting list are prepared for two- and three-hinged frames in steel, reinforced concrete, or laminated wood.

The user specifies material choices, external structure dimensions, and applied loads in IBM's free-form Problem Language ANalyzer (PLAN) language, and RFSP calculates the material requirements by type and size, computed for different combinations of dead, live, and wind loads on each structure.

RFSP runs in a 60K partition or region under DOS, OS/MFT, OS/MVT, or their VS counterparts. The minimum machine configuration includes one 2314 Disk Drive and one tape unit. Basic manual: IBM Form GH20-0598.

	Monthly License	Service Class
RFSP–Rigid Frame Selection Program (DOS) 5736-EC1	$25	C
RFSP–Rigid Frame Selection Program (OS) 5734-EC1	25	C

Securities Order Matching

This system is designed to aid firms in the securities industry. It processes security transactions by matching customer orders and execution reports, maintaining a file of

Program Products — Applications
IBM Corporation

➤ all active orders, and producing a file of completed trades each day. All orders processed must be in the New York Stock Exchange or American Stock Exchange standard formats, as set forth in the May 1971 Floor Communications Standards Manual. Unmatchable orders are routed to a 2260 display for special handling by a human operator.

Securities Order Matching must receive its input from the Telecommunications Control System (5734-F31), a program described elsewhere in this report. Prospective users should compare the combination of Securities Order Matching and TCS, which have a combined monthly license fee of $4,600 and offer only a particular application and message routing, to newer IBM and non-IBM data base/data communications offerings; these are normally considerably less costly and significantly more advanced.

Securities Order Matching is written in OS Assembler language and runs under OS/MFT or MVT with the following OS components and options: utilities, Linkage Editor E, primary data management (BSAM, QSAM, and BPAM), OS TCAM, Assembler-H, BDAM, Sort/Merge-H, and the TCS program. (The H-level requirements are for System/370; the 360 can use F-level.) Configuration requirements vary with user processing volumes and processing techniques, as well as with desired performance. Besides the OS requirements, Securities Order Matching with TCS requires a CPU with 512K storage (a partition of 140K is used for Securities Order Matching and another 100K partition is used for TCS), at least one 2260 Display Station, at least one 270X communications controller, one direct-access storage device for TCS and four for Securities Order Matching, storage protection and decimal arithmetic on the CPU, and a 1052 Printer-Keyboard. System generation, maintenance, and off-line operations require a card reader, printer, and magnetic tape, and a communications console is recommended. Basic manual: IBM Form GH20-4188.

	Monthly License	Service Class
Securities Order Matching (OS) 5734-F32	$2,100	B

Shared Hospital Accounting System (SHAS) Compatible Teleprocessing

SHAS Compatible Teleprocessing provides QTAM support of IBM 2780, 2740, and 1050 terminals for the no-charge SHAS system (prior-use program 360A-UH-11X). Terminal support for the 1050 is standard under SHAS, and this Assembler-language teleprocessing option significantly extends the SHAS capabilities by adding 2780 and 2740 support.

SHAS Compatible Teleprocessing runs in a 40K partition under DOS or DOS/VS. Basic manual: IBM Form GH20-4001.

	Monthly License	Service Class
SHAS Compatible Teleprocessing (DOS) 5736-H13	$275	B

Shared Laboratory Information System (SLIS)

SLIS operates with DOS and SHAS (Shared Hospital Accounting System) to provide support for hospital clinical laboratories. It accepts batched input of lab requisitions and test results, and produces lab working documents, ward reports, cumulative summaries, and statistical reports.

SLIS is written in COBOL (DOS) and Assembly language. It uses the executive facilities and census subsystem of SHAS and the DOS sort/merge and utilities. The minimum non-teleprocessing system configuration includes a minimum DOS COBOL system, 32K CPU, two magnetic tapes, three disks, and 22K in the background partition. A teleprocessing system adds to these requirements those of a minimum DOS-QTAM system with an interval timer. Basic manual: IBM Form GH20-0709.

	Monthly License	Service Class
Shared Laboratory Information System (DOS) 5736-H12	$250	B

Shop Floor Control

Please refer to Production Information Control System (PICS).

Simulation Language (for PL/1)—SIMPL/1

SIMPL/1 is a PL/1-based simulation language. Using SIMPL/1, a model of the system under study is written in SIMPL/1 source code and input to the SIMPL/1 preprocessor, which outputs PL/1 source code suitable in turn for input to the PL/1 Optimizing or Checkout Compiler. Generally, SIMPL/1 extends the number of source statements (from input to output of the preprocessor) by about 390 plus 15 to 20 times the number of SIMPL/1 source statements.

A System/360 or 370 supporting OS/MFT or MVT and providing at least a 104K partition plus one direct-access storage device will suffice for SIMPL/1 preprocessing and execution. Basic manual: IBM Form GH19-5035.

	Monthly License	Service Class
SIMPL/1 (OS) 5734-XXB	$250	B

STAT/BASIC for System/3 and ITF

STAT/BASIC is an interactive program that provides 40 procedures from among the most common statistical techniques for numerical analysis. The program, like MATH/BASIC, runs on a System/3 Model 6 or on a System/360 under ITF, DOS, OS, or TSO. The procedures provided are in the areas of data generation, elementary statistics, regression and correlation analysis, multivariate analysis, analysis of variance, nonparametric statistics, time series analysis, and biostatistics. A System/3 can handle a 300-observation (rows) by 30-variable (columns) matrix for most problems, but a System/360 cannot exceed a 100-by-15 problem at any time. Configuration requirements are the same as for MATH/BASIC. Basic manual: IBM Form GH20-1027.

	Monthly License	Service Class
STAT/BASIC (S/3-6 and 360/ ITF, DOS, OS, and TSO)	$45	B

➤

Program Products — Applications
IBM Corporation

Storage and Information Retrieval System (STAIRS)

Please refer to CICS, Report 70E-491-02.

Subroutine Library—Mathematics (SL-MATH)

SL-MATH is a FORTRAN analog to the PL/1-coded Procedure Library—Mathematics. Its description follows that of the latter program, but the FORTRAN language is used instead of PL/1. SL-MATH is available in two versions, one for the System/360 (DOS or OS), and one for the 1130 or 1800. It is distributed on magnetic tape, and 1130 or 1800 users must arrange separately to convert the program to cards for use on those systems.

SL-MATH runs under the following programming systems: System/360 DOS and OS, 1130 Disk Monitor System Version II, 1800 Time-Sharing Executive System, and 1800 Multiprogramming Executive System. The minimum requirements for compilation are a System/360 sufficient for any level of FORTRAN, or an 1130 or 1800 that can run the Basic FORTRAN IV compiler. Basic manual: IBM Form GH12-5103.

	Monthly License	Service Class
SL-MATH (360/OS or DOS) 5736-XM7	$100	B
SL-MATH (1130 or 1800) 5711-XM2	100	B

Tariff Publishing System (Bound Tariffs)

This system gives publishers of bound freight tariffs who use the 1050 Data Communications System for input generation a computerized technique for the conversion, publication, and maintenance of tariffs applicable to the surface transportation industry. Additionally, the data base it creates can be of use to carriers and shippers.

Three major program functions are incorporated within Tariff Publishing System: Unit Process Program, Maintenance Program, and Pagination Program. Unit Process translates input data from the 1050 transmission code to S/360 EBCDIC internal code; this allows full use of the 135 tariff characters. Maintenance processes input data against the previous issue of the tariff and all outstanding supplements. Pagination processes the cumulative supplement tape or new issue tape to produce a printer tariff; it follows all carrier conventions and sets the pages in optimum fashion. The user must convert or form a correctly formatted tape version of master tariffs, but after conversion, only changes need be input; the computer brings forward all other material.

The Tariff Publishing System programs are written in Assembler-F and operate under System/360 DOS, using DAM, consecutive disk and tape IOCS, indexed sequential file management system, unit record/disk and tape utilities, and disk or disk/tape sort/merge. Any System/360 or 370 with decimal arithmetic and enough storage for a 26K partition, a valid DOS/360 SYIPT device for input to tariff transactions, three 1600-bpi magnetic tape units, and 350 cylinders of 2311 disk space over that required for DOS, plus a 1403 Printer equipped with the special wide hammer feature and a 1416 print train cartridge with the special tariff set will suffice as the minimum system. Additionally, a 1050 Data Communications System is required, since cards containing tariff data are prepared on it. Its control (1051) and printer-keyboard (1052) must have automatic EOB, accelerated carrier return, and a dual-case print element. A 1058 Printing Card Punch with operator panel is required, and it is recommended that the first 1050 system in an installation be equipped with one 1056 Card Reader. Basic manual: IBM Form GH20-0730.

	Monthly License	Service Class
Tariff Publishing System (DOS) 5736-T21	$200	B

Telecommunications Control System (TCS)

Please refer to Securities Order Matching. TCS provides the message and network control functions for the Securities Order Matching system. Basic manual: IBM Form GH20-4187.

	Monthly License	Service Class
Telecommunications Control System (OS)	$2,500	B

Text Processor

The Text Processor system is written in Assembler language and consists of two modules: basic EDIT/360 and full PAGINATION/360. The system is intended for use by newspapers, book or magazine publishers, commercial printers, etc. The PAGINATION version of Text Processor is a comprehensive text processing, editing, and page makeup system containing all of the features of EDIT/360 as well as those of two earlier no-charge programs: COMPOSITION/360 (included in EDIT/360) and HYPHENATION/360. The primary enhancements of PAGINATION/360 over EDIT/360 are for page makeup functions, including the classification of input material by type (text, footnotes, graphics, etc.), the generation of page space requirements by input category, and the resolution and assignment of final page position based on user-defined parameters and page dimensions. The program can selectively reset pages whose format or content has been changed, and several different forms of jusitified page output are available.

PAGINATION/360 input consists of disk-resident textual material, text formatting commands, page formatting commands, and output requests. For page makeup, the program processes the text into justified lines and composed pages according to graphic and stylistic requirements described by the user. The program provides the user with composed pages according to specifications entered via page makeup commands.

EDIT/360 users are provided with the basic function of initial text entry, user-specified input edit criteria, text and command listing, justified printing, index listings, and modification of stored documents.

The Text Processor EDIT and PAGINATION modules run in a 34K or 76K partition, respectively, under either DOS or DOS/VS. The minimum machine configuration includes one or two 2311 Disk Drives for EDIT or PAGINATION, respectively. Basic EDIT manual: IBM Form GE20-0324. Basic PAGINATION manual: IBM Form GH20-4037.

Program Products — Applications
IBM Corporation

	Monthly License	Service Class
Text Processor–EDIT/360 5736-K11	$250	B
Text Processor–PAGINATION/ 360 5736-K2	450	B

Traffic Profile Analysis System

This system is used in conjunction with its prerequisite program, the Tariff Publishing System, to create a list of point-to-point freight tariffs in accordance with a carrier user's interest profile. It prepares the interest profile cards, which are then fed to the Tariff Publishing System for final output. Its configuration requirements are the same as those for Tariff Publishing. Basic manual: IBM Form GH20-0730.

	Monthly License	Service Class
Traffic Profile Analysis System (DOS) 5736-T22	$300	B

Vehicle Scheduling Program Extended (VSPX)

VSPX determines the routes a group of vehicles must travel to meet specific delivery commitments to customers at given locations. The program tends to optimize basic factors, such as travel time and number of vehicles. VSPX consists of two main sections, the Network Analysis Program and the Schedule Production Program. The first program analyzes a network representing the potential calling points by computing either actual or approximate distances between all points. The second program can repetitively produce schedules which meet various restrictions such as route time, speed, vehicle capacity, and customer requirements. The two parts can be executed independently.

In addition to performing the daily routing function, VSPX can be applied to redefine fixed routes, aid in determining feasible locations of warehouses and depots to service customers, plan for a new fleet, and provide statistical and cost data relative to the efficiency of a fleet.

VSPX (OS) is written in Assembler language and in PL/1. It runs under OS/MFT or MVT. Use is also made of the access methods QSAM, BSAM, and BDAM. To install the program, the OS Utility Program IEHMOVE is required. VSPX (DOS) is also written in Assembler Language and in PL/1. It operates under DOS. Tape-Tape or Tape-Disk and Tape-Card utilities are used to install the program.

Execution of VSPX (OS) requires a 60K region in a CPU with decimal feature, one 2311 Disk Drive (in addition to OS system devices) or sufficient storage space on a 2314, 2319, or 3330, one input device (card reader, magnetic tape drive, or direct-access storage device), and one output device (printer, magnetic tape drive, or direct-access storage device). One magnetic tape drive is required for the generation of VSPX. The DOS version requires only a 24K partition, but adds a printer-keyboard to the configuration, and also a tape unit if VPSX is to be included in the core image and relocatable libraries. Basic manual: IBM Form GH19-2000.

	Monthly License	Service Class
VSPX (OS) 5734-XM5	$175	B
VSPX (DOS) 5736-XM3	100	B

Visual Data Entry On-Line System (VIDEO)

VIDEO/370 is a key-to-disk data entry system that must be run on-line, using the System/360 or 370 CPU as the controlling device for the data collection logic. VIDEO/370 includes all of the functions of the earlier DATA/360 programs, except that support is provided only for the 3270 Information Display System as an input device (either local or remote). Additional functions include intensified display of erroneous fields, keyboard upper-case shift under program control, paging of records by variable increments, unlimited formats per batch, balance totals, crossfooting, extensive security provisions, and message transmission to the system control operator.

VIDEO/370 runs under DOS, OS/MFT, OS/MVT, or their virtual storage counterparts. Minimum CPU sizes are a 65K 360/30 or a 98K 370/135 for the DOS version and a 131K 360/40 or a 147K 370/135 for the OS version. The maximum number of 3270 control units and terminals varies with the CPU size and response time requirements. At least one disk drive and one tape unit are required. Basic DOS manual: IBM Form GC27-6968. Basic OS manual: IBM Form GC27-6966.

	Monthly License	Service Class
VIDEO/370 (DOS) 5736-RC3	$210	A
VIDEO/370 (OS) 5734-RC5	210	A

SYSTEM/3 APPLICATION PROGRAMS

Apparel Business Control (ABC) System

The ABC System is an order processing system for the apparel business. It incorporates seven basic order-related functions: Order Edit and Pricing, Order Writing, Bookings Reporting, Fabric Requirements Reports, Finished Goods Requirements Reporting, Stock Allocation, and Invoicing. The system of functions is tailored to the user's needs in accordance with a modifier program fed from questionnaire responses that have been keypunched into 96-column cards.

ABC is written entirely in System/3 Card RPG II. The minimum system required includes an 8K card-oriented S/3-10 CPU (5410-A2), a 5203-1 Printer, 5424-A1 Multi-Function Card Unit, and either a data recorder or data entry keyboard. The storage required can be affected by the user's color code table requirements, but the minimum system will allow about 150 of these. Basic manual: IBM Form GH20-0822.

	Monthly License	Service Class
Apparel Business Control (ABC) System (Card S/3-10) 5701-D12	$75	B

Program Products — Applications
IBM Corporation

Appropriation Accounting System

This is a general ledger, revenue, and appropriation program. It accumulates, analyzes, and presents financial data by fund, source, organizational unit, project, function, or any user-determined classification. It thus serves System/3 users in the same way that the Budget Accounting Information System (BACIS) serves System/360 and 370 users.

The Appropriation Accounting System is coded in Card RPG II and requires only an 8K System/3 Model 10 (5410-A2 CPU) with a multi-function card unit and a 132-column printer to run. Basic manual: IBM Form GH20-1049.

	Monthly License	Service Class
Appropriation Accounting System (Card S/3-10) 5701-G22	$100	B

Bill of Material Processor (BOMP)

BOMP establishes, maintains, and retrieves data from four basic manufacturing files: item master, product structure, work center master, and routing records. It can reorganize those files and retrieve information at various levels. The files are on disk. BOMP is able to run on a disk-oriented System/3 Model 6 or 10. It is an integral part of IBM's Production Information and Control System (PICS), and prospective BOMP users should refer to the PICS manual (GE20-0280).

BOMP is provided in Disk RPG II source code, and requires the Disk System Control Program (SCP) and Disk RPG II to run. The minimum S/3-10 configuration is a 16K disk-oriented CPU (5410-A14) with a disk drive multi-function card unit, and printer. The minimum S/3-6 is a 16K CPU (5406-B4) with a printer and a disk drive. A second disk drive is recommended on either system, since master file reorganization requires a new disk area plus an associated temporary disk output file. Basic manual: IBM Form GH20-0965.

	Monthly License	Service Class
BOMP (Disk S/3-6 or 10) 5702-M41	$50	B

Business Analysis/BASIC

Please refer to this program's description under the "System/360 Application Programs" heading.

Card Bill of Material and Requirements Planning

Card BOMR helps System/3 Model 10 users install requirements planning applications. It consists of the following programs: Activity Audit and Bill of Material, Increase Level and Replace Level, Where-Used, Requirements Generation, and Requirements Planning. The programs are used in conjunction with the S/3 card system utilities sort/merge. Aspects of the system are related to those of the System/360 and 370 Production Information and Control System, and it is a good idea for prospective users to study the PICS manual (GE20-0280).

Card BOMR runs on an 8K card-oriented S/3-10 CPU (5410-A2) with a multi-function card unit and a printer. For a quick-deck product summary run, 120 print positions are required on the printer. Basic manual: IBM Form GH20-0770.

	Monthly License	Service Class
Card Bill of Material and Requirements Planning (Card S/3-10) 5701-M42	$65	B

Citation Processing

Citation Processing handles accounting control for local governments in the area of parking and moving citations (parking tickets and traffic summonses). Its four subsystems are: Daily Parking and Follow-Up, Daily Moving and Follow-Up, Disposition Processing, and Monthly Statistical Reporting. The system runs on a minimum of a card-oriented 8K S/3-10 CPU (5410-A2) with a multi-function card unit and a 120-column printer. Basic manual: IBM Form GH20-4204.

	Monthly License	Service Class
Citation Processing (Card S/3-10) 5701-G23	$120	B

EPIC: SOCRATES, FAST, Budget/Finance Accounting, and Student

Please refer to EPIC's description under the "System/360 Application Programs" heading.

Health, Welfare, and Pension Fund System

The Health, Welfare, and Pension Fund System meets basic contribution accounting requirements for jointly administered health, welfare, and pension funds. It prepares and maintains employer, client, and management files, prepares claim and pension checks, and supports an inquiry facility for primary files. Written in RPG II, it runs on a 12K System/3 Model 6 with a disk drive and a 132-column printer. Basic manual: IBM Form GH20-1189.

	Monthly License	Service Class
Health, Welfare, and Pension Fund System (S/3-6) 5703-N11	$175	B

Hospital Patient Billing
Hospital Accounts Receivable

These two programs are totally separate, but they have identical system requirements on small card-oriented S/3 Model 10's and are likely to be installed together by the user.

Hospital Patient Billing performs processing runs necessary to bill patients and/or insurance companies for hospital charges and accounting for hospital revenues by source. It has several programs in its repertoire, each of which is

Program Products — Applications
IBM Corporation

loaded from an object card deck at run time. They are: Daily Admissions (report), Admission List, Insurance Analysis, Preliminary Census and Final Census, Transfers, Charge Pricing, Charge Posting, Insurance Prorations, Daily Balance Forward/Summary Bill, Final Bill, Insurance Bill Data, Dismissal List, Medicare Bill Data, and Revenue Report.

Hospital Accounts Receivable accepts input data from final bills and is used to generate audit lists, totals, and daily and periodic reports.

Both program products are written in Card RPG II and make use of S/3 card utilities. The minimum system required is an 8K card-oriented S/3-10 (5410-A2) CPU with a multi-function card unit and a 120-column printer. Basic manuals: IBM Forms GH20-0842 (Billing) and GH20-0762 (Accounts Receivable).

	Monthly License	Service Class
Hospital Patient Billing (Card S/3-10) 5701-H12	$65	B
Hospital Accounts Receivable (Card S/3-10) 5701-H11	35	B

Inventory and Requirements Planning

This four-phase Program Product is an integral part of IBM's Production Information and Control System (PICS) and uses files created by the S/3 Bill of Materials Processor (BOMP). PICS is discussed with the other System/360 and 370 application program products, and is outlined in IBM Form GE20-0280. BOMP is discussed in an earlier System/3 application program product description.

The system's four phases are: Inventory Planning, Projection, Requirements Planning, and Execution (processing of input transactions and status report preparation).

Both the System/3 Model 6 and Model 10 can use the Inventory and Requirements Planning system. The system configuration requirements are the same as for BOMP with the addition of Disk Sort (5702-SM1 on the S/3-10 and 5703-SM1 on the S/3-6). If the S/3-6 does not have an on-line data recorder, it must also have the S/3-6 Conversational Utilities (5703-UT1). Basic manual: IBM Form GH20-0971.

	Monthly License	Service Class
Inventory and Requirements Planning (Disk S/3-6 or 10) 5702-M52	$75	B

Job Analysis System/3 (JAS/3)

JAS/3 aids the planning functions of management by helping to plan, control, and supervise project-oriented work using the critical path method. It has a processing capacity of from 300 to 736 work items, depending on core size.

JAS/3 is written in S/3 Basic Assembler language and runs under control of S/3 Disk SCP. On an S/3-10 with the Dual Programming feature, JAS/3 must run in program level 1. It requires a minimum 9K partition in storage for a 300-item program, and will use up to 18K if possible. The minimum configuration is a System/3 Model 6 or disk-oriented Model 10 with 12K of CPU storage (5406-B3 or 5410-A13, respectively), any printer, and any disk unit. Basic manual: IBM Form GH20-1085.

	Monthly License	Service Class
JAS/3 (S/3-6 or 10 Disk) 5702-XP1	$70	B

Law Enforcement System (LES)

LES provides local governments' police departments with management, statistical, and administrative information. It is also capable of interfacing with FBI information systems. Functions within LES include: Uniform Crime Reports (FBI) Offenses Known to Police (used in determining crime trends), Uniform Crime Reports Return of Persons Charged (information forwarded to the FBI), Uniform Crime Reports Age, Sex, and Race of Persons Arrested (over and under 18), Accident Reports (useful for pinpointing accident danger areas), Dispatch Analysis (useful in allocating police resources), Field Interview Reports (on persons having contact with police at unusual hours or at places where they have no legitimate interest), and Police Personnel (administrative reports).

LES is coded in Card RPG II, and runs on the minimum card-oriented, 8K S/3-10 (5410-A2 CPU) with a multi-function card unit and a 132-column printer. Basic manual: IBM Form GH20-0981.

	Monthly License	Service Class
Law Enforcement System–LES (Card S/3-10) 5701-G21	$80	B

Optimum Blending

Optimum Blending is a small linear programming system for blending applications in the food processing and agriculture industries. Its blend function produces an optimal blend of raw material ingredients based on product specifications and ingredient analysis, and its report function prints mixing instructions. No user programming is required for its use.

The minimum system for Optimum Blending will depend upon the size of the blending problem. For program execution and report program compilation, an 8K card-oriented S/3-10 (5410-A2) CPU with a printer and multi-function card unit will suffice. To assemble the blending program, 12K bytes of storage, a disk drive, and the Universal Character Set on the printer are required. Basic manual: IBM Form GH20-0933.

	Monthly License	Service Class
Optimum Blending (Card S/3-10) 5701-D51	$90	B

Order Point Technique for Inventory Management

This is a set of four programs—Inventory Analysis, Stock Status, Transaction Processing, and Order Action—that help

Program Products — Applications
IBM Corporation

to implement an order point inventory system in a manufacturing environment. The final program in the set checks the current inventory status to determine recommended order action, basing the calculation on lead times and safety stocks required and the current average demand calculated in the Stock Status program.

The programs are written in Card RPG II and use either S/3 Card Utilities or off-line sorting. They will run on the minimum card-oriented S/3-10 (5410-A2), with 8K, a multi-function card unit, and a 120-column printer. Basic manual: IBM Form GH20-0741.

	Monthly License	Service Class
Order Point Technique for Inventory Management (Card S/3-10) 5701-M41	$50	B

Property and Liability Agency Accounting System

This system can be used by a small property and liability insurance agency to handle all its accounting processing as well as analysis and control of its book of business. The system also maintains a client file for multiple usage. It runs on a minimum card-oriented S/3-10 (5410-A2) CPU with 8K, a multi-function card unit, and a 120-column printer. For low-volume applications, data entry to the system can be via an on-line data entry keyboard; for higher volumes, a separate data recorder is recommended. Card RPG II and S/3 Card Utilities are also required. Basic manual: IBM Form GH20-0918.

	Monthly License	Service Class
Property and Liability Agency Accounting System (Card S/3-10) 5701-N21	$65	B

System for Television and Radio

This is a tool for management of commercial time and advertiser contracts in the broadcasting industry. It maintains files, can move "spots" vertically (time-of-day) and horizontally (day-of-week), prepare daily logs prior to broadcasting, and prepare invoices and data preparation forms. Its use is intended for small- to medium-sized broadcasters having total revenues of $1 million to $8 million.

The system runs on a System/3 Model 10 with 24K bytes of core (5410-A15 CPU) and a disk, a multi-function card unit, a 200 lpm, 132-column printer, and a printer-keyboard. It is written in Disk RPG II, and requires the Disk RPG II compiler and S/3-10 Disk Sort Program. The disk-resident card utility program helps installation greatly. A 9.8-million-byte two-disk configuration is recommended for efficiency. Basic manual: IBM Form GH20-4229.

	Monthly License	Service Class
System for Television and Radio (Disk S/3-10) 5702-K11	$375	B

Unit Inventory Technique

Unit Inventory Technique assists retailers in installing a unit control system. At its minimum level, up to nine stores can be accounted for, and more stores can be added by increasing the core size. Initially, the system maintains basic files and operating reports. It can examine items for sales performance and inventory levels. Color and size reporting is also possible.

Unit Inventory Technique is written in Card RPG II, with a portion of the file update routine coded in S/3 Basic Assembler. Hence, although the program will run on a 12K card-oriented S/3-10 (5410-A3) CPU with any model of multi-function card unit and printer, it requires use of a Disk S/3-10 for any modification of the Assembler-coded routine. Card RPG II is required. Basic manual: IBM Form GH20-0931.

	Monthly License	Service Class
Unit Inventory Technique (Card S/3-10) 5701-D11	$75	B

Utility Billing System

This is designed to handle the utility billing requirements of local governments and small utility companies. It computes and/or prints: meter reading books, consumption proofs, metered service charges, utility bills, cash proofs, revenue summaries, meter status reports, cut-off listings, cash postings and cycle balances, past due notices, new account listings, and rate/revenue statistics.

Utility Billing is coded in Card RPG II and will run on the minimum S/3-10 card-oriented system: a 5410-A2 CPU (8K) with any printer and multi-function card unit. The system must have the Card RPG II compiler and Card Utilities. Basic manual: IBM Form GH20-0882.

	Monthly License	Service Class
Utility Billing System (Card S/3-10) 5701-G24	$80	B

1130 APPLICATION PROGRAMS

Charge Materials Allocation Processor (CMAP)

CMAP allows ferrous and nonferrous foundries and other metal blending facilities to calculate least-cost initial charges of raw materials for melting furnaces. CMAP solves problems by calling the linear programming routines of Linear Programming System/1130 (LPS/1130), a Program Product described later in this report.

CMAP's source language is FORTRAN, and it operates under control of the 1130 Disk Monitor System, Version 2 (DMS-2). Program execution requires an 8K 1130 disk system with card reader, and system generation requirements add a card punch. Up to 32K of core is supported, and the system called upon to run the program and the one used to generate programs must have the same amounts of core and numbers of disk drives. Core storage over the 8K requirement and additional disks are used only by LPS/CMAM and result in only a slight performance

Program Products — Applications
IBM Corporation

increase. A printer in the configuration adds flexibility. Basic manual: IBM Form GH20-0456-1.

	Monthly License	Service Class
Charge Materials Allocation Processor–CMAP (Disk 1130) 5711-P11	$20	C

Construction Cost Control System (CCCS)

CCCS assists construction contractors in monitoring and controlling their job construction costs. It maintains files for up to 500 employees, 50 government projects, 2400 subcontractor work categories, 4800 labor work categories, the same number of material work categories, and 500 equipment work categories. Its output consists of payroll checks, a certified payroll for government project reporting, union reports of hours worked, a general ledger recap, subcontractor reports, quarterly tax schedules, W-2 forms, 941A reports, labor, material, and equipment comparison reports, and a job summary report.

CCCS programs are coded mostly in 1130 Basic FORTRAN IV, with some use of 1130 Assembler language. They run under 1130 DMS Version 2 on an 1130 Model 2B or 4B (8K with internal disk drive), two disk cartridges, a card read punch, and a printer. A card sorting facility is also needed. More core or faster I/O units will improve performance. Basic manual: IBM Form GH20-0975.

	Monthly License	Service Class
Construction Cost Control System –CCCS (1130 Disk) 5711-M62	$75	B

Construction Estimating Program (CEP)

CEP enables construction contractors to estimate costs for construction projects more accurately and faster than by manual methods. Concrete, masonry, and steel are divisions provided with CEP, with linkages provided for other divisions (referring to divisions defined in the AGC Manual, "Uniform System for Construction Specifications, Data Filing and Cost Accounting: Title One—Buildings").

CEP is written in FORTRAN and operates under the 1130 Program Language Analyzer (PLAN), which in turn operates under the 1130 Disk Monitor System (DMS) Version 2. An 8K disk 1130 with a card read punch and printer is the minimum system required. Basic manual: IBM Form GH20-0742.

	Monthly License	Service Class
Construction Estimating Program (Disk 1130) 5711-M61	$50	B

Continuous System Modelling Program II (CSMP II)

CSMP II is a digital analog simulator useful to engineers and scientists interested in modelling continuous systems. (Please refer to CSMP III for the S/360 for a description of continuous system modelling.)

CSMP is an extension of a program originally developed for the IBM 1620. It is written in FORTRAN IV and uses the 1130 Disk Monitor System (DMS) Version 2. The minimum system configuration required is an 8K disk 1130 CPU, a card reader and punch, and a printer. A highly recommended optional I/O unit is the 1627 Plotter. Basic manual: IBM Form GH20-0848.

	Monthly License	Service Class
Continuous System Modelling Program (Disk 1130) 5711-XS1	$50	C

Electronic Circuit Analysis Program II (ECAP II)

Please refer to the ECAP II description under the "System/360 Application Programs" heading.

Linear Programming System (LPS/1130)

LPS/1130 gives mathematical optimization capabilities to the 1130 user. Up to 500 rows on an 8K 1130 and 1500 rows on a 16K or larger 1130 can be used to solve problems in linear programming.

LPS will run on an 8K disk 1130 system with a card reader and punch or paper tape reader, but card read/punch capability is required for system generation and maintenance. LPS is written primarily in FORTRAN, with some use of Assembler language, and runs under the 1130 Disk Monitor System (DMS) Version 2. The system used for program generation and the one used for execution should have the same core and disk capacities. More or faster core and/or more disk storage speeds execution. Basic manual: IBM Form GH20-0562-1 with TNL GN20-2033.

	Monthly License	Service Class
Linear Programming System–LPS (Disk 1130) 5711-C01	$30	C

Project Control System II/1130

PCS II/1130 uses the critical path method to solve problems in planning and supervising project-oriented work. It processes up to 2,000 activities and allows up to 4,500 precedence relationships.

PCS II/1130 is written in FORTRAN and uses the 1130 Disk Monitor System (DMS) Version 2. It will run on a minimum 8K disk 1130 with a card reader and punch and a printer. Basic manual: IBM Form GH20-0878.

	Monthly License	Service Class
PCS II/1130 (Disk 1130) 5711-XP1	$75	B

Subroutine Library—Mathematics (SL-MATH)

Please refer to the description of SL-MATH under the "System/360 Application Programs" heading. ■

Decible III
Independence Computing & Software Corp.

MANAGEMENT SUMMARY

Decible III is a preprocessor program that accepts a decision table as input and produces a complete COBOL Procedure Division section corresponding to the decision table as output. The package also accepts all valid COBOL procedure statements along with the decision table input.

A common problem in program design, as The Man said to Cool Hand Luke, is often a failure to communicate. Decision tables are a means of aiding *logical* communications between persons. Narrative descriptions are often ambiguous, incomplete, and hard to follow, and flowcharts, while more exact, can be ponderous and difficult to prepare and maintain. On the other hand, decision tables are concise, clear, easy to prepare and understand, and can include all combinations of conditions.

A decision table is a 4-quadrant figure. Each quadrant is called a "stub," and the stubs are named (clockwise from the upper left) the condition stub, the condition entry stub, the action stub, and the action entry stub. Condition statements are written across the condition and condition entry stubs at the top, and action statements are written across the action and action entry stubs at the bottom. The vertical columns formed by condition entries and corresponding action entries in those stubs form the rules. In general, the number of rules equals 2^c where c equals the number of conditions.

This type of table can be applied to, say, an input editing problem. Consider a program that must consider as input an item type, location where sold, and other variables, and follow different processing paths based on such decisions as: (1) is there a sales tax in this city? (2) state? (3) county? (4) federal sales tax? (5) is the item taxable in any or all of these jurisdictions? (6) and so on. Not only does the decision table clearly lay out all the conditions, actions, and rules, but it also makes it comparatively easy to apply changes in the conditions or actions.

What's more, the decision table clearly shows all conditions for which the rule is a simple branch out of the table, all conditions which are impossible or invalid in combination, and all possible logic. Decible III goes a step further with its sequenced action entries: that is, the action entries for any rule can be numbered (in general decision tables they are just "yes," "no," or "don't care"), so that the actions taken under any rule can be in a sequence optimized for processing efficiency considerations or to prevent those rare occurrences when the sequence of rule actions could have illogical consequences. The user can indicate that he doesn't care about the sequence of some action entries by default or by giving them the same number. The Decible III Optional ELSE Rule acts to save the user from having to study and code all conditions for which the action is general.

Decision table theory is not new, and Decible III can be considered as a good idea well applied to data processing. In creating a Decible III decision table, there is only one difference from creating any other logical decision table;

▻

Decible III is a decision table preprocessor that can be used on virtually any computer system that can support COBOL. Properly used, it can facilitate programming and eliminate nearly all the logical errors that can occur in a program.

CHARACTERISTICS

SUPPLIER: Independence Computing & Software Corporation, 235 White Horse Pike, West Collingswood, New Jersey 08107. Telephone (609) 854-8924, (215) 925-9229, or (212) 757-0677.

BASIC FUNCTION: Automatic translation of logical decision tables into compilable COBOL source code, for virtually any computer system capable of supporting COBOL. Any combination of COBOL statements and decision tables is accepted as input. Decible III is also available in an analogous PL/1 version, but there are no customer installations of this version at present. Also, the vendor once supplied a FORTRAN version of Decible III, but plans for its future marketing are now uncertain.

OPERATION: Decible III input consists of condition statements that are valid COBOL conditional statements and action statements that are valid COBOL Procedure Division statements plus any other valid COBOL coding, in any combination, on 80-column cards, a magnetic tape source language file, or a disk source language file. If the input is from tape or disk, cards can be used to update and/or change the program simultaneously with processing of the program.

The output of the Decible III preprocessor is optimized compilable (and executable) COBOL coding with the actual decision tables automatically documented as COBOL NOTE paragraphs within the single normal listing. The Procedure Division sections thus created are named by the table names assisgned by the user in the Decible TABLE statements, and the decision table sections can be executed in the same manner as any other COBOL Procedure Division section, e.g., the program can PERFORM or GO TO a decision table or pass into a decision table from coding ahead of it.

In addition to the compilable COBOL program on cards, tape, or disk generated by Decible III, the package output consists of a printer listing of the program and diagnostics, plus an optional source language file on tape or disk. New statements entered via cards can be inserted in user-designated sequence into these files, which are automatically sequenced by 10's when created. This is a feature of the Decible source language library maintenance system. Not only statement records, but also blocks of records, can be inserted in sequence using this feature.

A provision for initial set actions is contained within Decible III, so that setting of counters and switches and reading of input can take place immediately upon entering a table and prior to any testing of conditions. The initial set actions can be any valid COBOL statements except conditional statements, GO TO, or Decible III special actions.

The Decible III special actions are as follows: (1) LOOP produces coding to branch back to the beginning of the condition testing logic *without* re-executing any initial set actions; (2) RE-ENTER is the same as LOOP except that initial set actions *are* re-executed; and (3) EXIT produces coding to branch to the end of the decision table. If the table was called by the PERFORM verb, control passes to the statement following the PERFORM statement upon EXIT; otherwise, control passes to the coding or other decision table following that decision table.

►

▷ the Decible III condition statements must all be valid COBOL conditional statements, and the action statements must all be valid COBOL Procedure Division statements. The tables can be created on any COBOL coding sheet, with only the simple format rules that columns 7-11 and 72 must contain blanks.

USER REACTION

Datapro was able to contact every single experienced user of Decible III, and their responses were quite favorable. Users' computer systems ranged from small IBM 360/30's under DOS to large 370's under OS and OS/VS2, as well as one Burroughs B 6700.

All of the IBM computer users have Decible III up and running in regular use, and all report that its output COBOL source code is always compilable and executable. What's more, all reported trouble-free installation, typically in one day. The Burroughs B 6700 user has had Decible III compiled on his system, but there were some syntax errors. These have been worked out by the vendor, with purchase contingent upon the user's final acceptance. The B 6700 user states that he likes the concepts of the package, and that the vendor is striving to meet the acceptance criteria.

Two of the IBM equipment users are using tailored 24K-byte versions of Decible III on small (32K) 360/30's. These users feel that, over a total of about 4-1/2 years' time, the package has saved them an average of about 30% in program test time and has also produced quality documentation as a byproduct.

Users of larger IBM 370's have Decible III in full 90K-byte versions and have an aggregate of more than 2 years' experience with the package. They praise the savings in coding and test time for programs and the documentation quality that permits different groups of personnel to pick up each other's work on programs. A dual-370/145 user stated that he's never found a problem in compiling or executing Decible III-produced code.

In summary, Decible III appears to be a sound implementation of the valuable decision table concept that deserves investigation by COBOL users who feel that they could profitably apply that concept to their systems. The vendor checks out as honest and hard-working, and is said to provide excellent training.□

► The Optional ELSE Rule is the rule to be followed only if none of the other rules can be satisfied. This is handy in, say, validity checking, where the user is interested only in certain specified sets of conditions. The standard rules in those cases can indicate only the few invalid combinations, while the ELSE Rule can handle all valid combinations. The ELSE Rule is indicated in the decision table by having the rightmost rule of the table have condition entries containing all dashes (–).

The end of a decision table is signalled by any statement that does not fit the format of a Decible III statement (e.g., any card with a non-blank character in columns 8-11 or 13 or a character other than C, A, S, *, or space in column 12, or an end of file indicator). The vendor states, incidentally, that Decible III is written in a sort of a "pidgin" COBOL for maximum compatibility with a variety of computer systems.

Sequenced action entries are a standard part of Decible III, so that users can select the execution sequence of actions under any rule, either for optimization or logical purposes. Sequences need not be indicated, in which case the actions will be sequenced in order of appearance by default; or actions with the same user-supplied sequence number will be executed in their order of appearance. The sequenced action entries can be numbered 1 through 99.

Decible III will handle up to 9999 separate decision tables, each with a unique table number and name. Each table can have from 1 to 20 rules (including the ELSE Rule), from 0 to 20 condition statements, from 2 to 30 action statements, and up to 100 cards, not counting the Decible TABLE and COMMENT cards. The package also has a complete set of diagnostics, consisting of error and warning messages. Out-of-sequence cards are ignored and printed with 3 asterisks to the left of their contents.

Warning messages indicate the existence of error conditions that the system can correct. In certain cases the corrections involve assumptions about the programmer's intent, and may therefore be incorrect. Hence, warning messages should be checked against the intentions of the programmer. An error message indicates an unrecoverable error that makes further processing of the decision table impossible. In that case, only the COBOL NOTE paragraph containing the decision table itself is produced. The 18 warning and error messages have associated numeric codes and are quite clear in meaning; for example "rules nn and mm not unique" means that the same set of conditions will pass rules nn and mm. Error recovery is facilitated by the fact that a mistake in one table does not prevent the compilation of the program and debugging of other sections of the program.

Finally, Decible III has the capability to support shorthand translation that permits 3-character abbreviations to be translated into longer data names or phrases, thus reducing the amount of coding necessary to produce complete documentation.

PERFORMANCE: Users currently running Decible III report that its output COBOL code has always been compilable and executable. Please see the User Reaction section of this report for further user impressions.

HARDWARE/SOFTWARE REQUIREMENTS: Decible III can theoretically be used on any computer system capable of supporting COBOL (e.g., nearly any IBM, Honeywell, UNIVAC, Burroughs, or NCR system), but only IBM System/360 and 370 users have Decible III *running* at this time. (See User Reaction.)

PRICING: Decible III can be purchased for $9,000 or leased for $300 per month for 36 months plus a $1,000 installation fee. These prices include an unconditional warranty, 2 days of a training course, and 2 days of special service. Leases are cancellable by the lessee after the initial 3 months, and 100% of those initial 3 months' payments can be credited towards purchase, as can 80% of the lease payments through the remaining 9 months of the first year of lease. Second and subsequent installations receive an 80% discount on purchase and a reduction in lease fees to $60 per month. Contracted maintanance after the first 3-year period costs $250 per year and includes updates, new versions, telephone consulting, and new versions of Decible III made necessary by conversion of hardware or software. Prices for the PL/1 version are the same.

INITIAL DELIVERY: June 1971.

CURRENT USERS: 5 as of December 1, 1973, all using COBOL; 4 on IBM System/360 or 370 computers and 1 on a Burroughs B 6700.■

INQUIRE
Infodata Systems Inc.

MANAGEMENT SUMMARY

INQUIRE is a data base management system that is extremely well suited for information retrieval applications. As a result of Infodata's continuing enhancement program, the system is currently in Version 5. Its 42 current users are divided mostly among food, chemical, pharmaceutical, and petroleum companies and government agencies. The varied applications of these users demonstrate INQUIRE's versatility as an information retrieval system, especially where large, stable data bases are involved.

In order to provide fast retrieval for highly complex queries, INQUIRE takes advantage of a number of file organization techniques in setting up an Index file and a Search file. These highly structured files use a keyword scheme to establish access points to the individual data items, and a "threaded list" to pick out all qualifying records from the data base. While these complex measures produce a very efficient retrieval capability, they also entail an overhead for the administrative and housekeeping aspects of updating the data base.

It is this overhead aspect that makes INQUIRE significantly better suited for analytical or keyword-oriented information or data retrieval applications than for highly transaction-oriented applications such as inventory control or order entry systems. The INQUIRE command language is intended for interactive on-line use, but the great majority of users pre-store the commands and process in batch mode. All maintenance transaction input data has to be rigidly structured, with adherence to punctuation and individual maintenance command structure rules. Each updated item can be described by one or more keywords, the item attributes (field values and/or unique item ID numbers), or a variety of operators that indicate the way the attributes are to be combined and interpreted for each item. Thus, input will usually require reformatting for maintenance transactions against the data base, unless an intelligent terminal can be used for data collection and preparation in the proper INQUIRE format. In 1974, Infodata will introduce a method of processing transactions without reformatting.

The retrieval requests presented to INQUIRE are free-form, but they must naturally be carefully thought out (as indeed they and all user aspects of INQUIRE or any data base usage should be!), and entered according to precise grammatical rules. Although INQUIRE provides for English-like query statements, the user must be knowledgeable in the syntax of the language as well as the specific nature of his data base. Extensive diagnostics are provided by the Command System to assist the user in preparing a query.

INQUIRE provides high-speed information retrieval for complex inquiries on IBM System/360 and 370 OS, OS/VS, and TSO systems. Its performance, enhancement programs, users' group, and vendor reliability have earned the package a good reputation.

CHARACTERISTICS

SUPPLIER: Infodata Systems Inc., 5205 Leesburg Pike (Suite 701), Falls Church, Virginia 22041. Telephone (703) 578-3430.

FUNCTION: INQUIRE is a keyword-oriented information retrieval system, written primarily in PL/1, that uses a threaded-list file structure to provide response to inquiries in an on-line or batch environment. A powerful set of retrieval commands associated with the self-contained data base permits individual items to be retrieved on the basis of complex sets of qualifying criteria. Trade-offs between the processing overhead required to modify the highly structured INQUIRE data base and the flexible retrieval capabilities must be made. The design of INQUIRE heavily emphasizes the retrieval aspects of data management, thus making INQUIRE most suitable for analytically oriented data retrieval use such as market research, laboratory testing, bibliographic, and text lookup applications.

INQUIRE, in fact, began life as a bibliographic system, but a transaction-oriented front-end processor is now under development, and commercial use of INQUIRE is increasing. Also of importance to commercial customers is the newly released and improved sort, which is designed to operate efficiently in virtual storage.

OPERATION: INQUIRE uses standard IBM OS access methods and file organizations to search a data base primarily through keywords or access points, using a "chain" technique. Thus, because the data base is accessed through index and pointer chains, search time does not increase directly with file size. The keywords can be either externally supplied or derived from fields within the item (internally supplied). Optionally, the files can be searched sequentially. The data base is stored on disk in up to 5 files that can be created, searched, and maintained by three INQUIRE subsystems: the Command, Loader, and Utility Systems.

The INQUIRE files consist of:

- An Index File in ISAM format with one record for each unique keyword (ISAM key) in the data base. Index File records are fixed-length, up to 120 bytes long, and blocked according to user specifications.

- The Search File in BDAM format, containing one logical record for each item or entry in the data base. The function of the Search File is to contain the "threads" or pointers that tie together all of the items or records described by the same keyword, as well as all of the keywords used for indexing a data item. Search File records are fixed-length and can be blocked up to 1 track long, depending upon the

INQUIRE
Infodata Systems Inc.

A macro language allows commands for common reports to be prestored, with individual users able to override and modify the prestored commands to suit special needs.

Though the structured nature of the retrieval requests calls for strict user discipline, it also permits excellent control over the specific content and format of the output information desired. A properly structured query, although it may require some effort to prepare, can result in the rapid retrieval of precisely the information desired. Moreover, the output is in a highly legible format, using default print parameters that can easily be overridden by the user if he so desires.

Infodata has organized an INQUIRE Users' Group, whose last meeting drew more than 100 attendees. At these sessions users interchange ideas and interact with company representatives to collectively express their will in shaping the direction of future improvements to INQUIRE. As part of a continuing policy, Infodata has an "engineered expansion" program whereby enhancements or upgrades are released for INQUIRE every four months. This policy has promoted a high level of user satisfaction with both INQUIRE and Infodata Systems.

USER REACTION

Over 18 months ago, Datapro contacted a number of the then 26 INQUIRE users, and found them uniformly satisfied with the system as a remarkably effective searching tool and with Infodata as a very reliable firm to do business with. A recheck with users today shows the same consensus.

Datapro interviewed the very first INQUIRE user, and found him still satisfied with the product and its vendor. This user reports having found only about 6 bugs in the system in over 4-1/2 years, and they were all fixed promptly by telephone. This user likes the product's efficiency and the ease with which users can get a fairly complex application running in a short time. Moreover, he rates the new INQUIRE features as "great." Specifically, Blocked Files have saved him about 40% in disk space, and he finds the High-Performance Sort very fast, especially on larger sorts. The user also likes the Users' Group "wish list," wherein users submit suggested improvements; it's valuable because Infodata typically responds.

Another user interviewed has taken INQUIRE from OS to OS/VS1 with no difficulty or problems; it just runs a bit slower, as could be expected. This user and others also generally echo the foregoing user comments.

In summary, INQUIRE is a high-quality system that provides unusually effective retrieval capabilities for applications in those environments where data base modification activity is not unduly high. Prospective

number of keys per item (8 bytes/key); a total of up to 16,777,216 records or items can be contained in both the Search and Search Overflow files.

- The Data/Item File in BDAM format contains all the information in a complete data record. Data records that are variable in length can be blocked into fixed-length blocks with provision for overflow from one block to the next. Data records can be up to 32,768 bytes long, with maximum file size limited only by OS. Each variable-length field (or repeating group, with a variable number of repeats) within a data record has an 8-byte overhead associated with it, making the use of variable-length fields economical only when the fields vary widely in length. When the fields vary by an average of no more than 8 characters in length, they should be specified as fixed-length.

- A Search Overflow File is used if needed to handle overflow from the Search File. This allows the Search File itself to occupy minimum disk space.

- A Decode File is available for code translation or table lookup on output.

The INQUIRE software systems that build, maintain and retrieve data from the data base consist of:

- A Loader System that creates the data base, performs data validation and editing, and provides for bulk additions to the data base. Rejected items are placed on a separate file for examination, correction, and resubmission by the user. Each new record is physically placed at the end of the existing Data/Item File, with updates to the appropriate keyword frequency counts (Index File) and respective "chained lists" of pointers or "threads" (Search File). As a result, new items are added to the file without the need for any reorganization. The basic Loader System consists of three individual programs and uses the IBM standard sort. A fourth program is available for editing keywords.

- A Command System that processes user requests for direct (on-line) or batch access to the data base and provides an interface to the macro library for maintenance, retrieval, and output as required. The macro library facility provides a shorthand method of expressing a query by pre-storing often-used command strings. These strings can define an output report or a complex computation. The macro capability can also be used in a tutorial way to help the user build a query in interactive mode by prompting him to specify parameters that define the request. An algorithm in the Command System automatically seeks the least-effort path for retrieving information where a series of keywords are involved, and search time is further reduced by the record chaining structure and parallel keyword processing that ensures disk arm movement in one direction only. Commands are entered with descriptions, command words, and operation in free-form. The Command System provides about 250 diagnostic messages for the user.

- A Utility System that provides the services necessary for batch-mode maintenance of the INQUIRE data base and the housekeeping functions associated with that maintenance. If, after a substantial number of additions to the data base, there is a significant increase in the number of unique keywords for the data base, the Index File can be "copied" to provide better

INQUIRE
Infodata Systems Inc.

▷ INQUIRE users should explore the overhead factor carefully with Infodata, because the distinction as to what level of update activity is tolerable is subjective, and depends in a complex manner upon a number of unique variables for each application. □

➤ extent allocation and extra space. In some cases, a "move" will also be useful to change disk device types with corresponding generation of new addresses. For Search and Data/Item Files, the space occupied by deleted records can thus be recovered, and the keywords and item formats can also be changed at this time. All changed data can be retained on a backup file for audit trail purposes. A "copy" with new address generation usually requires about one-fourth to one-third of the processor time required to generate the original data base (including subsequent updates).

Of particular interest is the macro capability in INQUIRE. As a supplement to the imbedded command language, the macro facility permits commands or segments of commands to be pre-stored in a permanent or temporary library for insertion into the INQUIRE command stream whenever the macro identifier is used.

A set of built-in macros provides one set of specified data, such as date-today, user ID, file names, etc. Another set of built-in macro functions provides conditional logic, text insertion and translation, terminal user prompting, and input from other files. Macro functions and conditional logic enable the user to build his own commands, suitable for the data base being used and the types of questions being asked.

The basic INQUIRE command language consists of the following capabilities:

- Keyword searching – full Boolean logic and links and role searching.
- Field comparison searching – separately or with keyword searching, arithmetic and character comparisons, range comparisons, and text searching.
- Computation – up to 10 levels of nesting, built-in functions, and date arithmetic.
- Report formatting – page headings, explicit or defaulted column title generation, numeric editing, conditional printing of fields, text insertion, and table lookup.
- Sorting.
- Totals and control breaks – before or after change, totals, average, and maximum and minimum.
- Cross tabulation – any number of tables generated in 1 search, and printed either at control break levels or as totals.
- Maintenance – add or delete records, and change field values or keywords.
- Terminal operation – identical capabilities and language as batch operation, operation with TSO, and on-line save of questions for later batch processing.

The Blocked File option saves sort, CPU, and I/O seek time. It allows for unlimited item size and for an unlimited number of fields. The High-Performance Sort option substantially improves sorting efficiency, both in computer time and in the number of records that can be efficiently sorted. The Statistical Option adds mean, standard deviation, standard error, and other calculations to the cross-tabulation feature. Options to interface INQUIRE with IBM's IMS and CICS,as well as a Multi-File Processor, will be announced in 1974.

HARDWARE/SOFTWARE REQUIREMENTS: INQUIRE runs on an IBM System/360 or 370 with the Universal Instruction Set, under control of OS, OS/VS, or TSO. The basic Loader, Command, and Utility systems combined require a partition or region of about 120K bytes plus space for OS. About 10K additional bytes of main memory are needed when utilizing the maintenance commands (ENTER, REMOVE, etc.). INQUIRE is provided with adjustable internal tables that permit the system to take advantage of as much main memory as is available, up to more than 200K bytes. A minimum of one direct-access device (2301 Drum, 2321 Data Cell, or 2311, 2314, or 3330-type disks) is also required. INQUIRE is supplied with all service modules and control information needed to install and run the package.

PRICING: Infodata's Usage License Plan provides for a permanent non-exclusive right to use INQUIRE for one or more applications according to the following schedule:

	One-Time Charge	Monthly Lease	Monthly Field Service
Basic INQUIRE:			
Initial Application	$14,000	$900	$275
Subsequent Application	*	*	*
Unlimited Usage	38,500	–	275
Optional Features:			
Blocked Files	4,500	280	50
High-Performance Sort	1,550	95	10
Statistical Package	2,500	160	35

* Prices on request.

Installation of the system is billed on a negotiable time and expenses basis, and system training is provided for $1,250 plus travel and living expenses.

Prices to U.S. Government Agencies are contained in Government Pricing Schedule Special Item 132-30 (Rental of Software Programs), Contract Number GS-OOC-00139.

INITIAL DELIVERY: July 1969.

CURRENT USERS: 42 as of October 31, 1973. ■

General Ledger System
Infonational

MANAGEMENT SUMMARY

This system isolates "out-of-control" situations by using forecasts, budgets, and actual performance criteria to monitor and report financial information. Analysis can be performed at the lowest operating entity, building up to division or company level. The system runs on IBM System/360 and 370 computers under OS, DOS, and VS as well as on Burroughs, DEC, Honeywell, ICL, NCR, and UNIVAC equipment.

The Infonational General Ledger System was designed to produce accounting information that enables management to conduct financial planning as well as make daily operational decisions, while alleviating repetitive accounting transactions. To meet these objectives, there are such features as automatic consolidation of multiple companies with different charts of accounts, automatic budget calculation, and automatic accrual reversals. The user can run a "pro forma" trial balance, enter adjustments, and reprocess. One user reported that, prior to using the system, this procedure required 6 people and 77 accounting sheets. Another user, using the optional Report Writer module, was able to continue his old planning methods, since this module allowed for varied accounting formats in his reporting and could handle his old control formats.

The Chart of Accounts independence within the system allows each company to determine its own chart of accounts and descriptions. No account range or sequence is imposed upon the user. Similar companies may share a single chart of accounts. A revised forecast of the yearly performance is produced monthly. This projection, consisting of actual performance to date plus an adjusted forecast for the remaining months, closely approximates the expected performance for the fiscal period. Management can see not only where the company stands but where it might be going.

Many automated general ledger systems in the past have produced a surplus of information with voluminous supporting detail for management's attention. Though Infonational's General Ledger System provides several levels of reporting, including cost center, division, and company, its greatest value is in producing analytic and exception-type reports. The General Ledger System was developed by system designers who not only understood the demands of general ledger accounting but the central role of the user in maintaining the system's efficiency and accuracy. With the delivery of GL III in October 1974, Infonational has confirmed its commitment to remain current, thus demonstrating why this system has gained the acceptance of leading public accounting firms.

USER REACTION

Datapro has obtained nine users' ratings of the Infonational General Ledger System through three responses to the 1974 survey of proprietary software users and six individual telephone interviews. The overall results were as follows:

	Excellent	Good	Fair	Poor	WA*
Overall satisfaction	5	4	0	0	3.6
Throughput/efficiency	3	5	0	0	3.4
Ease of installation	5	2	1	0	3.5
Documentation	4	2	2	1	3.0
Vendor support	4	5	0	0	3.4
Training**	1	5	0	0	3.2

* Weighted average on a scale of 4.0 for Excellent.

**Not user-rated in 1974 survey.

▷

CHARACTERISTICS

SUPPLIER: Infonational, 6626 Convoy Court, San Diego, California 92111. Telephone (714) 560-7070. Infonational also has marketing offices in Boston, Chicago, Dallas, New York, San Diego, England, New Zealand, and Australia.

BASIC FUNCTION: The General Ledger System automates the general ledger function. It allows for automatic handling of: consolidation of multiple companies, budget calculations, accrual reversals, and recurring journal entries. Comparative reporting (operating results to planned forecast) and responsibility reporting (detailed analysis of each level of management control) are provided, while the user is allowed to control the amount of detail to be reflected on the major financial statements. The system consists of 30 programs, all coded in COBOL.

OPERATION: The General Ledger System is designed to be independent of any predetermined numbering structure. Account numbers, organization codes, and their descriptions can be assigned by the user and can be changed, as required, without additional programming.

Input may come from one of the following sources:

- Manual input by coded journal vouchers—these are keypunched and entered into the system through the edit, batch balance, and voucher register routines.
- Computer-generated journal entries—these include reversals and machine-generated recurring and budget entries. All computer-generated entries appear in the voucher registers.
- Entries from administrative systems—for example, payroll and accounts payable entries can enter the system at several steps in the processing cycle: prior to validation, after validation but prior to voucher registers, or as direct entry into the General Ledger. This input can be in the form of cards, tape, or disk.

Error-clearing suspense accounts can be established for each company within the system. They are used to process coding errors and out-of-balance conditions. Each significant code is matched against the corresponding master file for complete validation. Error entries are then transferred into the error-clearing account. Total debits and credits are compared for each journal voucher to insure balance. An offsetting entry to the error-clearing account is generated when the amounts are not equal. The user has the option to override this feature by choosing to have the system reject unbalanced batches.

After editing and processing the input, the system can produce the following reports:

- Input Edit and Balance—a detailed listing of all input batches, indicating any coding errors or omissions. English-language error messages are displayed, allowing for a quick analysis of rejected items.

➤

General Ledger System
Infonational

▷ These users were running the system on the following computer and operating system configurations: IBM 360/40–1, IBM 370/125–2, IBM 370/158–1, IBM 370/145–3, Burroughs B 4700–1, Honeywell 115–1; operating under IBM DOS–3, DOS/VS–2, OS/MVT–1, OS/VS–1, Burroughs MCP–1, and Honeywell Mod 1 MSR–1.

The system performed as advertised immediately for five of the nine users. Six users made modifications to the system, and three users had the vendor make them. One user indicated that modification was feasible to the extent that he has plans to tie in his own inventory control program.

Overall, the users expressed a high degree of satisfaction with the system and the vendor. Many of these users have or are buying Infonational's other packages. As with nearly all proprietary applications software packages, the generalized nature of these products makes definitional demands upon the user. However, if the user knows his requirements, Infonational's General Ledger System is, as one user stated, "...an extremely powerful piece of software." Prior to deciding to buy any applications package, careful analysis of your requirements and firm decisions as to how *you* are going to use the system will translate into the satisfaction many of the users we contacted felt with Infonational's General Ledger System. □

➤ ● Batch Balance Recap–a summary report that shows the control totals for each batch and flags out-of-balance conditions. Debit and credit totals are listed separately for ease of control.

- Company Recap–a summary that highlights any error condition by company.
- Voucher Register–a detailed listing of all input by journal voucher number. This report can be used to audit input transactions and to balance to controls. Debits and credits are listed in separate columns. All recurring vouchers and accrual reversals are listed on this report each month.
- General Ledger Listing–a detailed year-to-date report of the general ledger that serves as a general ledger book and contains a record of all manual and system-generated transactions. Control totals are supplied for overall, control account, subaccount, and fiscal year month.
- General Ledger Account Analysis–a summary of the general ledger showing the total monthly transactions for each month by account. Control totals are furnished for overall, control account, and subaccount.
- Trial Balance–a listing of current period activity and year-to-date balances in each account. Preliminary profit and loss information in the form of sales, expenses, and profit is given.
- Flexible Budget Calculation–budgets can be developed for assets, liabilities, and equity accounts in addition to income and expense. Provision has been made for such features as the budget being calculated from external units such as production quantities and labor hours.
- Organization Expense Analysis–a supplemental report to the Profit and Loss Statement that shows totals for each expense element incurred by each organization for the current and year-to-date periods. Comparisons to forecast and budget are provided at the department, division, and company levels.
- Expense Ledger Analysis by Department–provides a comparison of actual performance to budget and forecast for each expense account in the general ledger. Each expense element is displayed department by department.
- Profit and Loss Statement–prepared for each organization, with comparisons to previous periods and plans.
- Balance Sheet–prepared for each organization, with a comparison to last year or last month.
- Current Outlook (Projected Profit and Loss)–a monthly reforecast of the expected performance for the fiscal year. Actual performance to date is combined with the estimated forecast for the remaining months to produce a pro forma profit and loss statement.
- Fiscal Year Timespread Profit and Loss–a listing by responsibility center that displays month-by-month data for forecasting or historical trends. A maximum of 13 period totals with year-to-date balance for each element of income and expense can be included.

The General Ledger System also provides for variable formatting and/or content of the General Ledger Detail Listing, the Profit and Loss Statement, the Balance Sheet, and the Current Outlook. The user specifies the variations through a series of input instructions which can be modified through a simple update procedure at program execution time. The General Ledger Detail Listing can be detailed for all months, or the balance forwarded for previous months and detailed for the current month. Three variables can be changed on the Profit and Loss Statement: percentage columns can be used as a percentage variance from plan or used to express expenses as a percent of sales; comparative amounts can be expressed as the budget or forecast amount or expressed as the variance from budget or forecast; and the amount of detail appearing in the report can be varied, with control exercised separately for the level of detail in each category. The Balance Sheet can be produced with or without comparative data, with comparisons made to forecast, budget, last month, or last year's data. The amount of detail appearing in the report can be varied, with each level of detail controlled separately.

There are three optional modules available: a Report Writer that allows management and other non-programming personnel to design financial reports, an Allocation Module used to pool and distribute expenses, and a Detail Selector Module for producing reports with comparative analysis at user-selected levels of management control.

HARDWARE/SOFTWARE REQUIREMENTS: The system runs on any IBM System/360 or 370 computer under DOS, OS, or their VS counterparts. Other available versions include: Honeywell's 115 (Mod 1 MSR), Series 200/2000 (OS), and Series 6000 (GCOS); NCR Century (B1); Univac 9000 Series (OS and DOS) and 1100 Series (EXEC 8); DECsystem-10; and ICL/1900 (GEORGE III). Needed are a minimum machine configuration of 64K bytes of main memory and three file devices, including one disk drive.

PRICING: The General Ledger System and its optional modules are available on a perpetual license basis at the following rates:

General Ledger System	$15,000
Report Writer Module	3,500
Allocation Module	2,500
Detail Selector Module	1,000

These prices include personnel training and installation assistance. Documentation, which consists of a user manual, systems and programming documentation, and operating instructions, is also included.

INITIAL INSTALLATION: GL I–July 1968; GL II–September 1972; GL III–October 1974.

CURRENT USERS: 131 as of January 1975, with approximately 20 percent of the installations in Europe. ■

MARK IV File Management System
Informatics Inc.

MANAGEMENT SUMMARY

MARK IV, currently in use at more than 700 computer installations in the United States and 37 other countries, is a leading example of a successful software product. Just as importantly, it is continuing to prosper and grow.

The history of MARK IV goes back to the GIRLS package (General Information Retrieval Language System) for the IBM 7090 computer system and MARK I, II, and III for the IBM 1400 series computers. The MARK IV version for System/360 or 370 computers under OS and DOS has now been in use for over five years.

Today, versions of MARK IV can provide graduated degrees of its capabilities to installations as small as an IBM System/360 or 370 DOS system, a UNIVAC Series 70 (ex-RCA) TDOS system, or a UNIVAC 9400 or 9480 DOS system. MARK IV can also be used effectively on medium and large systems, such as an IBM System/360 or 370 under OS (MFT or MVT) or a System/370 under DOS/VS, OS/VS1, or OS/VS2. Happily, the MARK IV versions for IBM real memory and virtual memory systems are fully compatible and priced identically.

The package's full name, MARK IV File Management System, does not reveal its additional and powerful capabilities for information retrieval and report generation. MARK IV is essentially a system for processing information by manipulating data files, and quite a few files can be handled during one run.

When MARK IV is used to update a master file it is not even necessary to name the transaction definitions in the run. They are called implicity, simply because the user indicates that a transaction file exists. But the user can still choose specific transaction definitions when several exist for the same file.

MARK IV's high degree of data independence is of heightened importance in these days of high user interest in data base systems, and especially so because of MARK IV's optional interfaces to IMS and TOTAL. MARK IV builds and maintains highly structured files without any need for the user to consider the logical structures of the files. Any field can be referenced at any time in combination with any other field, without the user even knowing where the fields exist in the data base.

The file maintenance method is primarily directed toward production files. Once the master file has been defined and created, the transaction processing for each type of transaction is defined. When updating the master file, all that is required is to input a run control card naming the file and transaction definitions; the transaction file is then passed against the master file.

The preceding is typical of the concept of programming with MARK IV. It is primarily oriented toward production programming. It is capable of handling most of the routine programming jobs in the typical commercially

The highly successful MARK IV package can facilitate the programming of many file-oriented business data processing problems and provide powerful file creation, file maintenance, information retrieval, and report writing facilities. Versions are available for a variety of IBM, UNIVAC, and Siemens computers. Fourteen options include IMS or TOTAL data base retrieval.

CHARACTERISTICS

SUPPLIER: Informatics Inc., MARK IV Systems Company, 21050 Vanowen St., Canoga Park, California 91303. Telephone (213) 887-9121. Offices are also located in Chicago; River Edge, NJ; Dallas; and Rockville, MD. In Europe, contact Informatics S.A., 18 rue Camille Martin, 1203 Geneva, Switzerland; telephone 022/45 22 00. In Japan, contact the Informatics agent, Computer Applications Co., Ltd., 3-1 2-Chome Hitotsubashi, Chiyoda-ku Tokyo; telephone 263-7241.

BASIC FUNCTION: To allow programming of common, file-oriented business processing problems, for IBM System/360 and 370 DOS and OS real and virtual memory computer systems, and also for UNIVAC Series 70 (ex-RCA) TDOS systems, UNIVAC 9400 and 9480 DOS and OS/4 systems, UNIVAC Series 90 OS/4 and OS/7 (when available) systems, and Siemens 4004 PBS systems. MARK IV provides extensive facilities for creation and maintenance of files, as well as for report generation. For those uses, it can replace high-level procedural languages such as COBOL. On-line operation under IBM's TSO is also available, and MARK IV has optional interfaces for retrieval from TOTAL or IMS data bases as well as a general interface to IMS.

OPERATION: MARK IV is actually a compatible family of six systems with graduated capabilities. The real memory and virtual memory models of MARK IV are identical in function and price. Their identities and relationships are as follows: MARK IV/214 (and MARK IV/224 for VS); MARK IV/234 (and MARK IV/244 for VS); and MARK IV/260 (and MARK IV/270 for VS). The following discussion pertains to MARK IV/260 or 270; the restrictions that apply to the lower-priced MARK IV models are discussed later.

All MARK IV files can be on punched cards, magnetic tape, or direct-access devices. Special features, discussed latter, extend the file handling and processing capabilities.

In general, MARK IV provides for creation of highly structured master files. Files are updated by passing a transaction file against the master file; the update processing specifications are defined and stored in the system prior to the maintenance run.

Information in the form of reports is retrieved from the master file via requests. Two methods are provided for making requests against the master file: one is for simple requests and the other for more complex requests.

Two auxiliary file handling capabilities available with the expanded request procedure greatly enhance the processing possibilities. Up to three independent Coordinated files can be used for supplemental input for each master record. Up to 10 Subfiles can be output in addition to the New Master and Report files. The subfiles output from one request can be used as Coordinated files for input to another request.

MARK IV File Management System
Informatics Inc.

oriented installation, and is also very useful for implementing one-time jobs, whether these are "ad hoc" reports or complete but short-lived or prototype systems.

MARK IV is not a programming language/compiler in the usual sense of translating a set of statements into an independent object deck that must be link-edited before execution. It does compile code in main memory, however, and it allows the user to specify that the compilation be optimized to favor either execution speed or economy of storage. The internally compiled code requires no further steps before execution. In effect, the MARK IV statements represent the parameters that control the linking and execution of the many subroutines contained in MARK IV.

A particularly powerful facility of MARK IV is the Free-Form specification that allows great latitude in formatting printed output. This feature makes it practical to use MARK IV for the many routine jobs, such as payrolls and tax reports, that require preprinted forms such as checks and tax forms.

Reports are generated by submitting Requests that specify the data items to beoutput. The new Basic Request form puts five card types on one form and is said by the vendor to actually handle 80 to 90% of all MARK IV requests in typical usage. Besides the Basic Request form, which can be used freely by non-EDP personnel, the older Information Request form can also be used by specially trained non-EDP personnel. The new form does not include all card types, nor all fields on certain card types, so as to be kept simple, but it can be used to extract records from a master file based on logical combinations of data fields in the master plus constants; output summary totals can be generated to nine levels. The expanded method permits arithmetic computation, in addition to logical relationships, to be used in determining records to be selected or to generate data items for output; it is with this method that the free-form output can be generated.

Requests can be cataloged and retrieved for execution with control cards. This gives, in effect, production-program capability. Moreover, the file concept of MARK IV also carries over into the data base area, since elements of the data bases are collected as logical records in logical files. In this mode, while logical files may be constructed in almost any manner, the physical data tends to exist only once.

MARK IV is a complete programming system that the vendor and many users regard as being a step above COBOL; a hierarchical order would place MARK IV at the top, then COBOL, and BAL at the bottom. However, a user can include his own routines in any language acceptable to his sytem for handling problems not easily tackled through MARK IV.

Because of the ease with which simple reports can be generated, the typical MARK IV installation uses it for one-shot report generation in addition to production jobs. In fact, a few installations have justified the program on that basis alone. Temporary files generated for such a

Master file maintenance can be batched with several requests to accomplish multiple jobs with one pass of the master file. Requests can be cataloged for later execution of the same task. All requests are held in source form and decoded at execution time; i.e., an object code deck is not generated for each job.

INPUT: In the full MARK/260 system there are a total of 11 input forms. These forms give rise to 22 different card types. Naturally, not all forms or all card types within a form are required for typical jobs.

A breakdown of the form and card types is as follows:

- **File definition: 1 form; 2 card types.**
- **Transaction processing definition: 1 form; 2 card types.**
- **Basic request: 1 form; 5 card types.**
- **Simple report request: 1 form; 4 card types.**
- **Expanded report request: 3 forms; 8 card types.**
- **Cataloging and retrieval of cataloged requests: 2 forms; 2 card types, as needed.**
- **Run control: 1 form, 5 card types, as needed.**

One additional form is available to define temporary fields. The card types are identified on the forms by preprinted two-character identifiers.

In addition to all the cards generated from the MARK IV forms, conventional JCL cards are required to initiate execution under the operating system.

MASTER FILES: Included in MARK IV is a powerful capability for defining files. A master file can be organized for sequential processing (SAM) or for random processing through the indexed sequential (ISAM) technique. IBM's VSAM will be supported in the next release of the VS operating system versions of MARK IV.

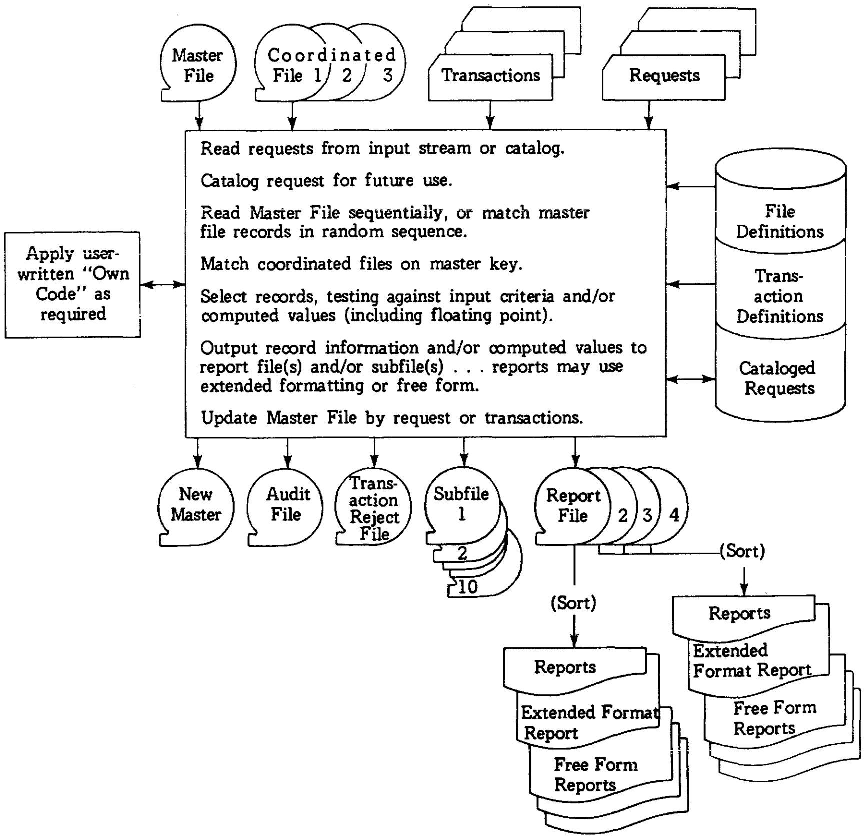
For any master file, data can be grouped in a hierarchical fashion. A record contains one or more levels of information, each containing one or more segments. The importance of the hierarchical structure to the user of MARK IV (as opposed to the designers of the MARK IV processor) is that repeated groups of data can be conveniently handled in the file specification and processing. Using the traditional personnel file as an example, multiple segments can be set up to hold data about the schools attended, previous employment, and references; each category will in turn be composed of several fields. It is highly unlikely that all employees listed in the file will have the same number of items in each of these three categories. Under MARK IV, each of these categories can be treated as a segment and defined. The file format itself is variable in length; as new segments in each category are added or deleted, the record length automatically expands or contracts. The number of segments in each category is carried in a count field on the next higher-level segment. A total of nine levels can be defined; there is no limit on the number of segments subordinate to a particular segment. A maximum of 99 segment types can be defined.

For sequentially organized files, up to three key fields can be established to govern the order in which the segments appear on the physical medium. For indexed sequential files, only one key field is allowed.

Fixed-format records can be defined, as well as records with a fixed number of repeating fields.

The hierarchical structure is not difficult to work with because all field positions are numbered starting at the beginning of a segment, not at the beginning of the record.

MARK IV File Management System
Informatics Inc.


Master File
Coordinated File 1 2 3
Transactions
Requests
Read requests from input stream or catalog.
Catalog request for future use.
Read Master File sequentially, or match master file records in random sequence.
Match coordinated files on master key.
Select records, testing against input criteria and/or computed values (including floating point).
Output record information and/or computed values to report file(s) and/or subfile(s) . . . reports may use extended formatting or free form.
Update Master File by request or transactions.
Apply user-written "Own Code" as required
File Definitions
Trans-action Definitions
Cataloged Requests
New Master
Audit File
Trans-action Reject File
Subfile 1
2
10
Report File
2
3
4
(Sort)
(Sort)
Reports
Extended Format Report
Free Form Reports
Reports
Extended Format Report
Free Form Reports

MARK IV in a nutshell. This diagram, reprinted from the Informatics Reference Manual (copyrighted in 1970), depicts the capabilities of the full-blown system, MARK IV/260. Various special features expand the file handling and processing capabilities.

one-shot job have a habit of hanging around. Normal controls and discipline should be adequate to prevent this tendency from tying up too many tape reels or disk packs.

There are currently six versions (models) of MARK IV. These are MARK IV/224, MARK IV/244, and MARK IV/270 for virtual storage systems, and MARK IV/214, MARK IV/234 and MARK IV/260 for real storage systems. Two earlier subset versions, MARK IV/210 and MARK IV/230, are no longer marketed. The real memory and virtual memory models of MARK IV are identical in function and price between corresponding models.

The forms used to specify operations with MARK IV—and there are many of them—appear at first glance to be complex and difficult to use. However, the entries are straightforward with only a few exceptions. In practice, it is not difficult to learn to use MARK IV advantageously. The chief objection to the handling of data entries on the input forms lies with the forming of logical relationships among data items for selection of records from a master file for processing. While the relationships are not hard to formulate and enter, once entered on the form they are difficult to comprehend visually.

Many users report that the execution times for MARK IV requests compare favorably with those of equivalent COBOL programs. This is not particularly surprising, because most file-oriented business problems are input/output-limited. But it is nice to know that most jobs probably won't take any more machine time than with

Each field can be defined as a character string, as a number in zoned decimal representation, as packed decimal, as fixed point (binary), or as floating point. A key strength of MARK IV is its extensive automatic data checking and conversion facilities. Every data field is verified every time it is accessed from a file. Invalid data is always flagged and never causes a memory dump. Conversions, even from alphanumeric to numeric, are automatically performed according to published algorithms.

The programmer must, of course, be aware of the file structure so that he can reference data in it. In coding the requests, the programmer need only know the names of the fields. To facilitate identifying the field names, a glossary can be printed, which, in effect, presents the complete structure of the file.

In addition to defining the physical and logical structure of the file, the same form can be used to define any editing (such as decimal point insertion and floating dollar sign) and titles (column heads) to be used when the data fields are printed.

Multiple definitions of the data items are permitted; i.e., the same physical field can be referenced by more than one name with independent editing patterns and titles.

The glossaries include the output titles which may be of assistance in identifying the data fields because data names are restricted to eight characters.

The same physical file can be multiply defined as well. This could be a neat way to restrict the availability of information in the file.

Existing data files can be defined in MARK IV terms and used. If desired, files can be conveniently restructured by using the existing file as a transaction file input for a master file creation run.

COBOL. An interesting feature allows the user to choose between optimizing processing speed or main storage allocation during any one run.

In summary, MARK IV is adaptable to most business data programming problems, though it would not be suitable for extensive computational procedures. MARK IV is strongest where higher-level programming languages such as COBOL are weakest: in managing the input and output of data. It is weakest where these languages are strongest: in specifying computational procedures.

In addition to standard batch and remote batch operation, MARK IV provides facilities for use in an on-line, time-sharing environment, either through commercial time-sharing companies or for in-house use. In conjunction with the On-Line special feature, a conversational prompter and text editor called EDIT IV is provided. The On-Line special feature utilizes a free-form language oriented to keyboard input. Complete compatibility between MARK IV structured forms and the free-form language is preserved. The On-Line special feature includes the facility to interactively develop and then execute MARK IV applications from the console. It runs under TSO and is essentially the same system that is available on the National CSS and Interactive Data Corporation networks.

MARK IV has been well supported by Informatics, as would be expected in view of its success in the marketplace. The program has also given rise to a unique organization, the MARK IV User's Group (IV League). The facilities provided by the full package are so flexible that it becomes quite advantageous for users to exchange techniques.

Since its first delivery about six years ago, MARK IV has undergone many changes, including a few major ones. Informatics estimates that about 1200 enhancements and modifications have been made at no charge to the user to improve and, in a few cases, correct the package. Major modifications are treated as extra-cost special features, and a total of 14 have been introduced.

USER REACTION

In order to gauge the current user feelings about this important package, Datapro sent direct mail queries to 23 present MARK IV users. These users were asked to fill out and return a questionnaire, and 17 of them obliged us by doing so. These 17 users employ a total of 22 CPU's.

The initial question was "What version of MARK IV are you using?" To this, 9 of the 17 responded "MARK IV 260" and 4 responded "MARK IV 270". The remaining three responded with the release number (e.g., 3.0) instead of the MARK IV model number.

The question "When did you begin using MARK IV?" was asked to determine whether the user's remarks were based on a reasonable amount of experience. One of the respondents declined to answer this question, but of those who did answer, installation dates ranging from as early as March 1968 to as late as January 1973 were indicated. Average user experience with MARK IV was 39 months.

AUXILIARY FILES: In the basic system, the auxiliary files, the coordinated files and subfiles, are straightforward sequentially organized files. The purpose of the subfiles is to collect data for a future processing run of some type. Also in the basic system, a coordinated file is referenced in relation to the master record. Only those records that are matched to the master record are available for processing. The auxiliary file facility is available only to users of the expanded reporting method.

REPORT GENERATION, SIMPLE: Once a Master file is created, whether it is a newly created file or a MARK IV-defined existing file, information in the file can be retrieved. There are two methods for doing this.

The first method makes use of only one form, the Information Request form. This form facilitates coding of simple inquiries for detail listing and/or summaries of selected information. No computation with data fields can be performed.

The overall printing height and width of the page are controlled by one-character codes. A range of form sizes can be specified, as well as an installation standard set up by the user.

Selection of records to be included in the output can be based on the logical combination of the values of specified data fields compared to a second data field or constant. The compare operations provided include equal, not equal, less than, greater than, less than or equal to, and greater than or equal to. Each data field serving as a criterion for record selection is identified by name (gleaned from the appropriate glossary). Each name of a data field appears on a separate line of the input form. Logical connectors, AND and OR, are identified by one-character codes. The code for the OR connector is blank, zero, or the letter "O," in recognition of one of the most common keypunching errors—mistaking zeros for "O's" and the converse.

Compound logical relationships are clarified by assigning logic levels, which take the place of the parentheses that would be used if the logical relationships were written by hand. The proper logic level to use is easy to figure out by writing down the relationship, counting the pairs of parentheses that surround a particular connector, and entering the count on the same line on which the connector appears. While this is a relatively simple and natural procedure, the result on the form may be difficult to interpret, leading to problems in verifying and debugging.

Once records are selected, the specified data fields and summaries are printed (or output to a print file). Again, the data fields are identified by name and the file glossary is necessary to determine the names. Arithmetic computation is allowed after data fields have been selected, and the user can select, compute, select, compute, etc., to any level desired in any request. Additionally, a field can be partially specified dynamically during either selection or output, with no need to over-define it in the file definition. Multi-line formatting is easily done with the "end line" specification. Although this selection references only the entries shown on the Information Request form, this is not a limitation of the system – it just happens that only the fields (capabilities) most often required for simple jobs are included on the Information Request form.

In essence, the report output consists of a column of data for each data field identified plus any summaries that are called for.

The titles appearing over the columns of data are taken from the Master file definitions. Each data field is written on a separate line on the input form. Sequence numbers can be assigned to control the order in which the data columns appear across the page.

Up to nine levels of control breaks can be defined, based on the data fields. The sequence of the report need not be in the same order in which the information comes

MARK IV File Management System
Informatics Inc.

Describing their computer systems, the users showed a cross-section that ranged from IBM 360/40's, 370/135's, and 370/145's under DOS (there were 5 of these systems, each with from 240K to 512K bytes of main memory) to large systems that included multiple 370's or multiple 360/65's, either under an MP (multiprocessing) or ASP (attached support processor) version of OS. There was one OS/VS1 user and one OS/VS2 user. Among the 12 large-system users, the smallest had a 370/145 with 512K bytes of main memory running under VS1. For each of the remaining 11 users, the CPU was a 360/65, 370/155, or larger, and main storage ranged from 1 million to 4 million bytes.

Asked how many bytes they used for MARK IV, some users responded with a range, some with the partition size, and others with a direct answer. The answers ranged from 28K to 400K, with the average at 118K. The average for the 5 DOS users was 112K, and the average for the OS and OS/VS users was 111K. The user who responded with 400K, by the way, indicated an application-dependent range of from 80K to 400K. The next largest answer, exclusive of a partition size, on a single machine was 128K bytes. The largest DOS user runs MARK IV in 200K on a 512K 370/145, using twice as much main storage as any other DOS user.

Then users were invited to respond with a check-mark, in boxes labeled "excellent", "good", "fair", or "poor", to these categories: their overall satisfaction with MARK IV; their rating of MARK IV's throughput and efficiency; the ease with which MARK IV is installed; quality of the vendor's documentation for MARK IV; and technical support provided by Informatics. Responses indicated considerable user satisfaction: 48% of the total responses were in the "excellent" category, 45% were in the "good" category, and 6% were in the "fair" category. (There was only one "poor" grade given by a user, which accounts for the final 1%.)

These raw scores are shown below, along with the weighted average (on a 4.0 basis, and rounded to 1 decimal place) and the percentage of "excellent" responses for each of the five categories:

	Excellent	Good	Fair	Poor	Weighted Average	Percent Excellent
Overall satisfaction	12	5	0	0	3.7	71
Throughput/ efficiency	4	10	3	0	3.1	24
Installation ease	12	5	0	0	3.7	71
Documentation	6	9	2	0	3.2	35
Vendor support	7	9	0	1	3.3	41

Asked whether MARK IV performed as advertised, 10 of 16 (63%) responded "immediately" and the remaining 6 responded "eventually." No one said "never."

Asked whether MARK IV required modification (other than enhancements, changes, etc.) on their systems, the overwhelming majority, 13 of 16 (81%), said "not at all." One of the remaining three, a DOS user, responded that Informatics modified his MARK IV system to fit his GRASP-enhanced operating system. (Informatics now markets a

from the master file. Up to nine data fields can be identified as sort keys ranging from major to minor. The sort sequence, for individual keys, can be ascending or descending. The sorting is performed by a standard utility routine prior to printing.

At each control break the following summaries can be produced: total, cumulative total, item count, maximum value in group, minimum value in group, and average value of the group. Summaries can be controlled to include several groups within one summary. The detail listing can be suppressed and the summaries only can be printed. If desired, subtitles (control break data field names) can be printed for the summaries printed at control breaks.

A title taken from the input form can be printed if desired. A limited horizontal format control can be achieved through the capability to specify the number of spaces to appear before a data column. Not using this facility causes an automatic default to two spaces between columns whose width is taken as the longer of the data value specification or the column title. The facility makes use of the standard provision for continuing to the next line any layout that exceeds the width of the page.

The programmer can also request the program to halt for mounting of special forms.

REPORT GENERATION, EXPANDED: The Expanded report preparation method represents a large expansion of capabilities over the simple method. The single form of the simple method is expanded to three forms: Record Selection and Processing, Output Specification, and Title.

The facilities for record selection provided over and above those of the simple method include additional input sources, arithmetic manipulation of input data (add, subtract, multiply, and divide), definition of result fields that can be used in subsequent processing, and conditional or nonconditional branching based on sequence numbers as labels.

Using this form, the master file can be modified.

Thus, there are three data fields possible for each line of entry: two operands and a result. For character fields, a portion of the field can be identified for use; this applies to only one field in each line. This facility is helpful for breaking out portions of a compound field, such as a date field, without redefining the master file.

Sources of input, in addition to the master file, include one of the three possible coordinated files, temporary fields, flags, or constants (second operand only). A temporary field can be a previous result, as mentioned above, or a field defined on a separate form. Flags indicate the status of processing of files, such as master file unchanged or matches found on the first two coordinated files. They can also be used to direct certain operations such as retrieving the operating system date, placement of page numbers, deletion of master records, and to force end-of-file.

The capability to specify data fields for output is also expanded. The source for the data fields can be the old or new master file, one of the three coordinated files, or a previously defined temporary field. Computation must be accomplished on the Record Selection and Processing form. Other capabilities include multiline formatting capability, the automatic display of a group total as a percent or ratio of another data field, and the ability to override the standard edit format in the file definition. The file definition edit specification is made by using code characters; the edit specification on the Output Content and Format Specification form is more like an editing mask. Partial character fields can also be identified with this form. The facilities for summaries, sorting, and control breaks are identical with those of the simple method of report specification, with one added feature: fields which are utilized as sort or

▷ GRASP-compatible version of MARK IV.) None of the users said they had to modify MARK IV themselves.

Finally, users were invited to respond in their own words about why they like and are using the package, whether they have experienced any problems with MARK IV or Informatics, and about other packages they may have tried and rejected. The remarks obtained were copious and generally favorable. Remarks of a negative nature, when they occurred, were not damaging to MARK IV's image. Rather, they gave some warnings that all should heed: don't attempt to use MARK IV for jobs that should be done by COBOL, and make sure that the programmers and other personnel using the package understand it well. As to competitive packages, probably the closest to MARK IV in its nature is ASI-ST, covered in Report 70E-051-01.

MARK IV, according to this sampling of its probable 600 customers at about 700 installations (about 20,000 people *use* the product), is a good to excellent buy as a multi-purpose file management system. □

➤ control break fields need not be included among those actually printed on the report.

The Output Specification form is used to identify all data fields to be output and to specify the form in which they are to be output. It also identifies whether the output is to be to one of 10 subfiles and/or 4 report files. For report files, this form also controls the overall format of a report. A great deal of flexibility is provided for positioning of dates, page numbers, and column headings, as well as height, width, number of lines per page, vertical spacing, etc. A capability for repeated images permits printing forms two or more up. This is particularly helpful to expedite the printing of many narrow forms, such as labels.

The Title form allows the specification of multi-line titles, comments that appear as a preface to a report, and comments that appear in a dictionary listing but not in a report. The last of the previous items is very useful for documentation purposes. The Title form provides one other very important capability: free-form layout. In this mode, the programmer has complete control of the layout. Use of free-form permits printing on pre-printed forms.

SPECIAL FEATURES: A total of 14 special features have been delivered to date for MARK IV users. Special features are separately priced.

The Table Lookup facility allows a programmer to conveniently reference a table of data through coded arguments. Three types of tables can be set up: displacement, sequential search, and binary search.

The argument displacement table is an ordered list in which the argument of an entry in a table is implied by its location; e.g., a table of the 12 months of the year whose arguments are the numerous 1 through 12.

A sequential search table is an unordered pairing of arguments and results. The entire table is searched sequentially from the beginning until a match is found. Obviously, this method is not recommended for large tables. It has the virtue of being easily modified because there is no specific sequence to worry about.

The binary search table is an ordered listing; the search technique is to compare first on the midpoint of the table, the then on the midpoint of the half containing the desired argument. The table is continuously halved until the proper entry is found. The half containing the desired argument can be identified because the table is in ordered sequence.

The Table Lookup special feature introduces one new form and two new card types.

The Time Processing special feature allows the user to process files containing hour, minute, and second data as an additional data type. Computations, formatting, and conversion of time data are all supported. The feature will also handle any three-unit system, such as inches, feet, and yards, drams, ounces, and pounds, etc.

The Indexed Coordinated Files special feature permits direct access by key to ISAM files and to IMS and TOTAL data bases. VSAM will also be supported in the next release of MARK IV for the VS operating sytems. The source of the key for the direct-access read can be a specified field in the master file, in another auxiliary file, or a temporary (internal) data field.

The Extended Transaction Processing special feature permits validation of data entries in the transaction file. Data entries are checked to see if they conform to an edit pattern that specifies the permissible characters for each position of a field. Certain standard character groups, such as A-Z or 0-9 plus blank, are provided, and the user can identify more if he chooses. Also provided with this special feature is the capability to define and treat a transaction file just like a master file.

The Extended File Processing special feature allows the programmer to handle up to nine coordinated input files (versus the previous three) independently of the master file. Without this special feature, only those records on the coordinated files that match master records are available to the programmer for processing. Hierarchical relationships can be established among the coordinated files that make certain files subordinate to others and accessible only through those files.

Extended File Processing also permits creation of indexed sequential (ISAM) subfiles. All files, except ISAM files, can have up to nine record keys in place of the previous three. VSAM will also be supported in the next release of MARK IV for VS operating systems.

The Text Processing special feature permits the programmer to search character strings (text) for the presence of a specified data item, such as a particular word. Once one or more matches are found, the record can be selected for processing, all hits can be replaced with a substitute character string, or the number of hits can be counted. Alternatively, text data can be scanned for a particular pattern, such as a 6-character string of which the first two characters must be numeric digits and the last four characters can be any alphabetic or blank. Data can be scanned from left to right or right to left. Variable-length fields are also supported.

The Data Base Interface/IMS special feature allows MARK IV users to define and access existing IMS data bases or to create valid IMS data bases and use IBM's IMS facilities. The data base can be accessed sequentially or randomly, and any legal type of MARK IV report or processing can be applied against the data base.

The Checkpoint/Restart special feature allows the user to set up checkpoints that can be written during processing and selection or report generation runs to save selected main memory contents and the states of peripheral units. Checkpoints can be automatically taken at selected intervals or under operator control.

Resource Optimization, once a special feature and now standard, gives the user the option of optimizing either processing speed or main memory use. Speed optimization typically results in an improvement of 2 to 10 times the unaided run time. Structured, hierarchical files always optimize main storage, and nonstructured, single-level files optimize processing speed by default. ➤

MARK IV File Management System
Informatics Inc.

The Extended Reporting special feature, originally developed by a user to meet his special needs, allows special formatting and calculation functions. Greater control over output page width, zero suppression and insertion, and blank line insertion or suppression are obtained at the cost of additional preparation time to count spaces on reports and provide pictures of the input data. Additional calculation capability includes multiple percent or ratio calculations on the same output line, with more control over the number of decimal places in the output.

The Formatter special feature is a batch program which converts a MARK IV free-form, keyword language to standard MARK IV input formats. This is useful for on-line typewriter-oriented systems which do not have the On-Line special feature with its interactive MARK IV prompting capability.

The On-Line special feature is an interactive typewriter-oriented prompter and text editor which enables an on-line terminal user to create and modify MARK IV input using a free-form keyword language. It converts such input into standard MARK IV input formats. It includes a run facility which allows the resulting input stream to be executed at the terminal, with input and output optionally provided at the terminal.

Data Base Retrieval/IMS and Data Base Retrieval/TOTAL are two special features that provide retrieval systems for report and subfile generation from data bases managed, respectively, by IBM's IMS (Report 70E-491-01) and Cincom's TOTAL (Report 70E-132-01). Technically, transaction processing is possible, allowing an in-core update; however, output files cannot be created.

The Extended Segment Processing special feature provides the ability to write and call subroutines in MARK IV. It extends MARK IV's standard branching functions to include a backward branch and several generalized branches that allows greater programmer control of the execution cycle when such control is required by the application.

MARK IV/214 and 224: These versions provide a subset of the full MARK IV capabilities oriented primarily toward report writing. The basic system allows retrieval from the information contained in a single master file. Both the simple and expanded report writing capabilities are included with a few limiations. Capabilities for the full MARK IV not included in the basic system may be added as optional features except for the file creation and maintenance capabilities provided by the automatic transaction processing features.

MARK IV/234 and 244: These models provide a subset of the full MARK IV capabilities but still comprise a complete file maintenance and information retrieval system. Retrieval from a master file and one coordinated file, as well as full transaction processing capabilities for file creation and maintenance is permitted. Also included are three subfiles, both the simple and expanded report writing capabilities in their complete form, and complete temporary field capabilities.

HARDWARE/SOFTWARE REQUIREMENTS: MARK IV requires partitions ranging from a minimum of 40K to a maximum of about 120K bytes for most applications. Sorting performed in the course of outputting reports is done using the computer manufacturer's sort utility routines. Various MARK IV models will operate on various manufacturers' systems, as the chart below shows. MARK IV with the On-Line special feature operates in a conversational mode under TSO on an IBM System/360 or 370, as previously noted, and MARK IV itself will operate in several time-sharing and/or virtual machine environments, including the Cambridge Monitor System on an IBM System/360 Model 67. In the chart below, the DOS/FG1 and DOS/FG2 versions are special high-efficiency versions. Unlike the normal DOS version for IBM, they can operate only in the indicated partition, not in foreground and/or background.

Latest MARK IV Release	Manufacturer, Computer Series and Operating System	Available Versions of MARK IV
4.0	IBM System/360 OS/MFT & OS/MVT	260, 234, 214
4.0	IBM System/370 OS/VS1	270, 244, 224
4.0	IBM System/370 OS/VS2	270, 244, 224
4.0	IBM System/360 or 370 DOS	260, 234, 214
4.0	IBM System/360 or 370 DOS/FG1	260, 234, 214
4.0	IBM System/360 or 370 DOS/FG2	260, 234, 214
4.0	IBM System/360 or 370 DOS/GRASP	260, 234, 214
4.0	IBM System/370 DOS/VS	270, 244, 224
3.0	UNIVAC Series 70 TDOS	260
3.4	UNIVAC 9400/9480 DOS	260
3.4	UNIVAC Series 90 OS/4	260
3.0	Siemens 4004 PBS	260

Additionally, a version will be released for the UNIVAC Series 90 computers under OS/7 when that operating system is installed in the field.

DOCUMENTATION: As befits a prosperous product, the MARK IV documentation is well done. Five volumes – User's Guide, Reference Manual, Operations Guide, Special Features, and Pracniques – do an excellent job of telling the user how to use MARK IV. The User's Guide by itself is a good introduction to MARK IV and will permit nonprogrammers to use many of the facilities without getting too wrapped up in details; but it is not indexed and is therefore difficult to use as a reference. The reverse side of each preprinted MARK IV form contains an annotated copy of the form; while this does not replace the Reference Manual, it is very helpful. A "Quick Reference Folder", in fanfold pocket-size form, is well laid out and quite useful in preparing MARK IV forms. This form is also known as the MARK IV "concertina."

PERFORMANCE: Please refer to the User Reaction section of this report.

PRICING: The U.S. purchase prices, with first-year maintenance and support, training, and installation, for the six MARK IV models are as follows:

MARK IV/214 or 224 – $10,600
MARK IV/234 or 244 – $21,100
MARK IV/260 or 270 – $37,000.

Purchase prices of the special features are:

Table Lookup – $2,700
Indexed Coordinated Files – $5,300
Time Processing – $1,100
Data Base Interface/IMS – $13,000
Extended File Processing – $5,300
Extended Transaction Processing – $5,300
Text Processing – $3,700
Checkpoint/Restart – $2,100
Formatter – $2,100
Extended Reporting – $1,100
On-Line – $10,000
Extended Segment Processing – $3,700
Data Base Retrieval/IMS – $7,900*
Data Base Retrieval/TOTAL – $7,900*

*Price with MARK IV versions other than 214 or 234; with those versions, the price is $5,300.

A number of lease plans are also available. Significant price breaks for multiple installations are provided under both lease and purchase plans.

SUPPORT: The purchase price of MARK IV includes installation and training for the user's personnel.

INITIAL DELIVERY: Early 1968. The latest release was Version 4.0, available November 1, 1973. It was actually the 15th version released to users in the field.

CURRENT USERS: About 700 installations as of November 1973. ■

Fast Dump Restore
Innovation Data Processing, Inc.

MANAGEMENT SUMMARY

Dumping and restoring are two of those essential utility tasks that must be done by nearly every computer installation a number of times each day. The larger and/or more complex the operational environment, the more frequently these functions must be performed. Typically, in a medium-to-large System/360 or 370 OS environment, about two hours per day are devoted to dumping and restoring, and significantly more time may be required in an on-line environment with remote users.

Innovation advertises that its Fast Dump Restore (FDR) routines can dump and/or restore IBM 3330 and 2314-type disk packs up to five times as fast as the standard OS utility. Also, FDR uses about 40 percent less tape than the standard utility, thus allowing the operator to back up files with fewer reels and correspondingly reducing tape handling time. Also advertised is a 30 to 80 percent reduction in tape EXCP's and a 40 to 90 percent reduction in disk EXCP's. These reduced I/O operations are due primarily to increased blocking factors, to the elimination of redundant and unnecessary data (CCW's, improved null-track handling, etc.), and to the use of cylinder orientation (versus the standard OS utilities' track orientation) to minimize delays due to physical rotation, arm movement, etc.

The users contacted by Datapro support Innovation's performance claims and report a high level of satisfaction with the package. In April 1973 FDR was improved by the addition of a self-loading stand-alone restore facility, thus clearing up the most common user complaint: initially, FDR could run only under OS and thus could not be used to restore the operating system in the event of a full system failure. The only remaining complaint voiced by FDR's users concerns its present inability to handle partial packs, but FDR's vendor promises to overcome that problem during the summer of 1973.

In the opinion of the Datapro staff this latter limitation, while true, is far from serious; the partial-pack capability of the standard IBM utility is infrequently used and is likely to create more problems than it solves. For instance, partial-pack copying by the standard IBM utility requires that the tracks be returned to their original specific locations upon volume restoration. Such a requirement creates a complex disk space management environment, and one that is wholly under manual control.

In all, FDR is an eminently worthwhile program, applicable to any OS environment, and a package that should be investigated by cost-conscious data processing managers. ▷

This handy little OS system utility can greatly reduce disk pack dumping and restoration time – which consumes about two hours of computer time per day in typical medium-to-large-size OS installations. FDR users typically report savings in excess of 50% of overall dump/restore time.

CHARACTERISTICS

SUPPLIER: Innovation Data Processing, Inc., Clifton Executive Plaza II, 925 Clifton Ave., Clifton, New Jersey 07013. Telephone (201) 777-1940.

Marketing activities are conducted by Computer Software Trading (Oslo) in Scandinavia, and by Westinghouse Management Systems (Paris) in the European Economic Community (Common Market) countries.

BASIC FUNCTIONS: Fast Dump Restore (FDR) is a high-speed System/360 or 370 OS or OS/VS utility used to dump/restore 3330 and/or 2314/2319-type disk packs to magnetic tape.

OPERATION: FDR handles both the IBM 2314 and 3330, and must be link-edited into one of the System/360 or 370's load libraries.

Specifically, the FDR utility programs are designed to replace the standard IBM OS supervisor component that provides the dump and restore functions.

FDR is not a supervisor component, but rather is a utility program that operates as a user application in a partition or region.

The following capabilities are supported by FDR:

- All allocated tracks on disk are dumped. If a VTOC (Volume Table of Contents) error or potential error is disclosed, the entire pack is dumped, thus preventing the possible loss of data due to invalid VTOC codes.
- FDR places the label or volume serial number of the disk pack on tape. Under "parameter" card control, this disk ID can be restored or discarded when restoring the disk file at a subsequent time.
- All allocated alternate tracks on disk are dumped. At restoration time, the alternate tracks are respecified depending upon the new pack's initialization parameters. (Note, the standard OS facility will specify the same alternate track addresses on this new pack as on the old one – potentially using a bad track.)
- FDR accepts backup tapes dumped not only by FDR itself, but also by the standard OS disk dump facility.
- DOS-format packs can also be handled by FDR (e.g., when running DOS under OS emulation). ➤

Fast Dump Restore
Innovation Data Processing, Inc.

▷ Of particular significance in considering the value of an improved dump/restore utility is the potential to do more dumps for the same processing dollar. Unfortunately, most installations don't dump as often as they should, resulting in more cumulative alterations to the data sets contained on disk volumes between backups. Thus, when restoration from tape is required for "live" volumes, a more costly process is encountered. With FDR's higher performance, dumps can be taken at more frequent intervals for the same cost in time, thus reducing the recovery expense over the course of a month's processing. This improved backup situation is especially important to on-line, transaction-oriented users with high-volume data base updating activity. Also, with a less time-consuming way to dump files, operations management personnel are less likely to skip a dump, thus guaranteeing better overall recovery capability. □

➤ In addition to the above FDR capabilities, more than 20 diagnostic messages are provided, including a completion code of zero to indicate a completely clean run.

FDR can dump up to nine 2314 or 3330 packs serially or concurrently in a single job step, and/or it can restore one pack at a time, including the old volume serial number or retaining the volume serial number of the new pack.

PERFORMANCE: Depending upon the CPU and magnetic tape models, FDR typically runs from slightly more than 3 to as much as 5 times as fast as the standard 2314 dump/restore facility for either dumping or restoring; from slightly less than 3 to slightly more than 4 times as fast as the standard 3330 dump facility; and from about 2.5 to 3 times as fast as the standard 3330 restore facility. The above ranges are based on full disk packs and "typical" multiprogramming environments.

HARDWARE/SOFTWARE REQUIREMENTS: FDR runs on an IBM System/360 or 370 computer under OS PCP, MFT, or MVT (or their VS counterparts) in a separate partition or region. Main memory requirements, in bytes, are as follows:

	FDR 2314	FDR 3330
Dump up to nine packs serially	64K-68K	100K
Dump up to nine packs concurrently	64K-68K for first pack, plus 60K for each additional pack.	N/A
Restore one pack	54K-56K	100K

PRICING: FDR is available for unlimited use on a single CPU upon purchase of a permanent license for $1,190. The price includes object deck, user documentation, and a hard-copy source listing. Additional CPU's (up to six) can be licensed for $595. Additional quantity discounts are available for more than six CPU's. A copy of the FDR source code on tape costs $50.

INITIAL INSTALLATION: July 1972.

CURRENT USERS: Over 100 in the United States. ■

PAC I
International Systems, Inc.

MANAGEMENT SUMMARY

Controlling EDP projects requires two things: accurate; and timely information and the ability to systematically use that information. PAC I (Project Analysis and Control system) is a package developed and marketed by International Systems, Inc. that is designed to facilitate the gathering, compiling, and presenting of the required project information. In addition to giving presentations on the PAC I system itself, the vendor routinely presents workshops targeted for project coordinators who use the PAC I system as a tool, and conferences for user management personnel who receive the system's output.

International Systems began operations in King of Prussia, Pennsylvania, in June 1969. The company remains located within a stone's throw of its original site, near famed Valley Forge. International Systems first installed PAC I in October 1969. By January 1973, 130 PAC I systems were installed, and at least 50 additional installations have been made since then. These installations are utilizing a wide range of computer systems from nearly all of the major U.S. and European mainframe manufacturers. PAC I installations can be found in North America (about 120), Europe (most of the remainder), and scattered throughout Africa, the Orient, and Australia. The company's four founders are still with the firm, but the president, who is European-born, resides and maintains an office in Europe.

International Systems cites the following benefits from use of PAC I:

- Controlling the high cost of systems development and programming.
- Controlling all projects by task, regardless of the sizes of the projects.
- Reporting progress on a timely basis, highlighting problem areas.
- Comparing actual progress and costs to estimates.
- Maintaining a project history of cost, manpower, etc.
- Determining departmental loads and employee availability.

The foregoing compares well with a report from the executive vice president in charge of operations at a large Eastern bank, who told an American Bankers Association National Operations and Automation Conference that PAC I provided all the data he needed to accomplish Steps 3, 5, 6, and 7 of the following 7-point project management system:

(1) Prepare a 5-year application (overview) plan.

(2) Outline more detailed 1-year plan segments.

▷

PAC I is a project analysis and control system that provides the tools needed to manage EDP projects by giving the required information on a consistent, clear, and timely basis. It runs on virtually any COBOL computer system.

CHARACTERISTICS

SUPPLIER: International Systems, Inc., 150 Allendale Road, King of Prussia, Pennsylvania 19406. Telephone (215) 265-1550.

International Systems has foreign representatives in Toronto, Ontario; Paris, France; Munich, Germany; London, England; Stockholm, Sweden; Amsterdam, Netherlands; Johannesburg, South Africa; and Tokyo, Japan.

BASIC FUNCTION: PAC I is a software system, executable on virtually any COBOL configuration, that is designed to assist in project management, control, costing, and planning, and in cost and resource allocation and control. It accepts input at detail levels, computes and reports total project estimates, computes and/or converts between estimates of percentage completion and hours remaining, generates up to 20 reports, and compares estimates of time and cost to actual figures. Concisely and basically, PAC I is a time and information gathering system that reports performance to date against initial estimates and projects future performance in terms of hours required to complete and probable completion date.

OPERATION: Projects are broken down according to the discretion of the user. Information can be gathered at any level and summarized to higher levels. The newest PAC I system input forms are designed for use as turnaround documents, thus presenting last week's progress in review to each person supplying input to the system about his assignments. The form also provides a minimum input recording facility.

The user must determine task objectives and constraints, manpower resources and rates, resource time and status reporting frequencies and techniques, report types, structure, frequency, exception constraints, summary and detail levels, distribution, and project and task numbering system. The system maintains a project information data base, provides an interface for computer usage and job accounting reports, has facilities for priority resource planning, and automatically simulates project scheduling and rescheduling. It produces up to 13 (soon, 20) reports on such factors as: project and task status, cost/time accounting and forecasting, project and nonproject expense data, departmental staff load analysis, individual resource planning and load reporting, task performance analysis and history, individual performance profiles, staff information reporting, and turnaround input documents. Recent enhancements have added a charge-back report and ability to create a priority network for each task.

The system has several algorithms to project future loading of resources. These algorithms can be used in combination, based on a target or upon availability, or modified by up to five types of priorities. Reports can account for such items as an individual's vacations, illnesses, military leaves, absences due to education, and so on, extrapolating the effect of the absences on the project's progress. The system can enable the user to reschedule key tasks as appropriate.

➤

(3) Gather for input detailed projections by programmer and application for each successive quarter as it approaches.

(4) Utilize a formal, albeit flexible, approach system that sees all plans approved by senior management, with management approving final justifications on major applications before they are started. The head of DP operations also has input to this phase.

(5) Report weekly on open applications, presenting status and total project outlook.

(6) Evaluate and compare estimates versus actual figures when an application is closed out; also present a history of the application.

(7) Present quarterly report summaries on department performance as a whole with respect to initial objectives.

One can see that most operations managers would indeed be grateful for a system that accomplishes four out of seven of these points, and the most paperwork-laden ones (e.g., reports, projections, etc.) at that.

The number of reports available from PAC I has steadily grown since the system's inception, and now stands at 20. The basic price of the system includes the user's choice of any 10 reports, with the remainder available on a per-report basis for an extra charge.

The system is undergoing constant enhancement; the most recent (in fact, at this writing it hasn't yet reached the users) includes simplified input forms.

One of the most desirable features of the package is that it projects total project estimates from its own calculations which are based on input gathered at the most basic levels. Thus, the project director is spared the arduous task of performing these calculations. Moreover, the system accepts either time to completion or percentage completed as input (whichever is easier for the reporter), and calculates the other while compiling sums for reports on both.

USER REACTION

Datapro's inquiries turned up some very positive comments by users of PAC I. One user said that in his 19 years of experience, the package is the best tool he's seen for managing systems and programming. Other customers stated that the system "does everything that International Systems says it will," "helped us (two companies) understand what we needed to complete projects," and "doesn't let programmers 'fudge' their reports or estimates."

Apparently it was Datapro's good fortune to talk to users who knew how to apply this management tool, because, as the vendor cautioned, the system can be no better than the use to which it is put. There can be little question that the package performs as advertised, and users also emphatically praise its installation ease, quality of documentation, and vendor support.

The users contacted had experience with PAC I ranging from six months to two years, on IBM and UNIVAC systems. It was reported by one user that his company brought the system in on a Monday, had test programs running that Wednesday, and was running production jobs with the system on the following Monday. A banking user credits PAC I with moving his operations department into a well-controlled environment from a past of unmanageable workloads, inaccurate and missed deadlines, larger and larger non-understandable reports, more frequent and longer-lasting (and, we're sure, more harrowing) meetings with management, and uncertainty in project orchestration and cost and manpower allocation.

In the first annual Datapro survey of proprietary software users, PAC I had two users reporting. These users both rated the package "good" in terms of their overall satisfaction, and both reported that the product performed as advertised immediately. All other responses to Datapro's questions regarding the package were either "good" or "excellent." In conclusion, Datapro can confidently recommend the package. □

PERFORMANCE: As with any program of this type, PAC I itself is merely a tool. It is impossible to state whether it can perform properly in meeting its objective—cleaning up a user's project planning and scheduling— since that is a function of how the user uses the tool. It is possible, though, to state how the system performs its information gethering, calculating, and reporting tasks.

According to users, the PAC I system is easily installed, is becoming easier to use with each enhancement, and provides all the reports and calculations with advertised speed and accuracy. Users report satisfaction with the package running in from 44K to 85K bytes (depending on the number of reports desired) and typically using 1.5 to 2 hours per week of computer time.

HARDWARE/SOFTWARE REQUIREMENTS: PAC I will run on most COBOL systems. It has operated successfully on Burroughs (B 2500, B 3500, B 5700, and B 6700), Control Data, Honeywell, IBM (360 and 370 from Model 30 upward, under DOS or OS), ICL, RCA Spectra (now Univac), Siemens, and UNIVAC (1100 Series) systems.

PRICING: PAC I is available fully installed at a purchase price of $9,250. This includes 10 reports, first-year maintenance, and first-year workshops and conferences. Maintenance is optional after the first year for $400 per year, and includes workshops and conferences. The vendor provides advice, consultation, and an initial 40 hours of training and support. Also provided is a source-code tape in installation-standard COBOL and documentation manuals.

INITIAL DELIVERY: The first PAC I system was installed in October 1969.

CURRENT USERS: 180 as of August 1973. A high percentage of these are in banking, government, and insurance. Many users are U.S. and foreign blue-chip companies, such as major banks, automobile manufacturers, airlines, petroleum companies, etc., as well as agencies of governments, utilities, and railroads. ■

Sprint
Jason Data Services

Sprint is a money-saving spooling system that supports basic System/360 DOS installations and extends to sophisticated System/370 DOS/VS users spooling three batched job partitions. Its reputation for cost-effectiveness has yielded a sixfold increase in installations within 18 months.

MANAGEMENT SUMMARY

Sprint is the lowest-priced non-IBM spooling supplement currently available for System/360 or 370 DOS or DOS/VS. At prices starting at $95 per month on a month-to-month rental basis, Sprint offers many of the most vital spooling functions available in systems costing up to half a dozen times as much.

In common with other DOS spooling supplements, Sprint can improve the throughput and greatly extend the power of DOS or DOS/VS. In fact, it is reasonable to anticipate an overall throughput improvement of 30 to 40 percent for typical installations with reasonably well-balanced (CPU-bound versus I/O-bound) workloads. This enhanced operation may result in reduced expenditures for additional equipment or overtime charges, and may permit postponement of an operating system upgrade that would otherwise be required to handle an increased workload.

System throughput improvements provided by either version of IBM's POWER are substantial, yet they leave a wide margin for further improvement by independent packages such as Sprint, which Jason claims can increase system throughput by 10 to 20 percent more than the IBM spooler can.

An interesting point about Jason Data Services that deserves special mention – and is a major factor in any decision to do business with the firm – is that Jason is a one-man company. The man is Mr. Jay Hanson. Sprint's low cost is due in no small measure to Jason's low overhead, but let's explore the potential hazards of using a spooling routine from a one-man software company.

A spooling package is a supplement to the operating system. As a system control program, a spooler services all users on an equal basis and interfaces primarily with the operating system for the host computer. Thus, individual user program changes do not require corresponding changes in the spooler. Of course, the addition of new types of peripherals or terminal devices to the system will require new device interface capabilities in the spooler and code for unique block lengths, control characters, etc.; but support for a wide variety of devices can be built into the spooler to handle many configuration changes based upon known types of devices.

Also, any changes in the operating system can require corresponding changes in the spooler. Sprint is designed to work with DOS or DOS/VS on System/360 or 370 computers. With IBM's announced stabilization of System/360 DOS at Release 26, no further significant changes will be made to that operating system for System/360 users. System/370 DOS or DOS/VS users, however, may be subjected to continuing developmental work that could require Sprint modifications for continued spooling operations. From a hardware point of view, most System/360 and 370 peripherals are currently supported under Sprint, although future IBM peripheral releases could cause a corresponding need for Sprint modification in order to support the new devices. Over the past year, Jason has made the transition to several newer IBM System/370 devices, including the 3330 Disk Drives and 3211 Printer.

▷

CHARACTERISTICS

SUPPLIER: Jason Data Services, 22511 Woodcrest Circle, El Toro, California 92630. Telephone (714) 581-0640.

FUNCTION: Sprint is a spooling supplement to IBM's real and virtual storage Disk Operating Systems. It runs either by itself in a foreground partition or transparently in upper main memory to provide spooling support for most standard IBM peripherals, as well as for imaginary devices which need not be either physically attached to the system or available to a specific partition. Input job streams can be processed by Sprint and are placed in intermediate disk buffers prior to loading into main memory for execution. Output files begin direct media conversion to cards or paper from intermediate storage before completion of the job producing the output. A tape version, for output printer spooling only, is also available.

Also available is a sophisticated relocating loader facility that makes all user programs automatically self-relocatable, and thus able to run in any partition. Another feature, device address independence, allows job-stream portability between partitions. For multiprogramming systems, an efficient partition balancer polices the system to ensure that no one partition monopolizes the CPU. Job accounting routines are standard with Sprint.

OPERATION: Sprint is initiated as a user job in the input stream through job control statements, and is available either in a disk-oriented input and output version or in a more primitive tape-oriented version that handles printer output only. Both versions are available for IBM DOS or DOS/VS.

Features available with Sprint include the following:

- Early printer start, enabling output to be printed while the job creating the output is still running.
- No required changes to the operating system or previously set up JCL for input job streams.
- Simultaneous card punch and printer output support (although only one printer and one card punch can be driven at the same time without an asynchronous processing supervisor).
- Support of IBM 1400 Series emulation.
- Semi-automatic restart from the beginning of a job specified by the user.
- Independent partition support for up to three batch job streams while running Sprint in high main memory. (Even under DOS/VS, Sprint handles a maximum of three batch job partitions.)
- Job accounting routines to itemize CPU, device, and forms usage by job step within job stream.
- Support of UCS for printer output.
- Multiple output copies.
- Ability to backspace the printer file for recovery.

➤

Sprint
Jason Data Services

And as this was written, the firm had just completed testing 3340 disk support.

Therefore, for the System/360 or 370 DOS or DOS/VS user, especially in the medium-scale or smaller range, the maintenance requirement for a spooling supplement is small. Jason has demonstrated that the typical Sprint installation can be made in about 15 minutes, with no changes to the operating system or the user's programs. Furthermore, the terms and conditions of Sprint usage are very easy on the user, with no lease to sign. Thus, a user whose spooling requirements can be satisfied by Sprint can look forward to saving up to $1000 per month, or even more, for as long as Sprint satisfies his needs, without undue concern about the fortunes of Jason Data Services.

USER REACTION

Every one of the Sprint users contacted for the purposes of this report indicated complete satisfaction with Sprint and Jason Data Services. No problems of any consequence were uncovered in the investigation; the worst difficulty mentioned was one user's report that he had to have Sprint modified slightly in order to handle mid-program form changes. On the positive side, all users interviewed cited Sprint's low cost and high performance as reasons for their satisfaction. Most of these users employ around 8K to 10K bytes per spooled partition.

Jason's technical support was called "excellent" or "good" by all users interviewed, except for those who happily reported that they had never needed service. The Sprint users all reported that the vendor is responsive, and many were grateful for the continuing series of enhancements (such as procedure libraries) to the package, all delivered free of charge.

In summary, Sprint is the most attractively priced spooling supplement to IBM's DOS or DOS/VS now available, and it includes enough of the important spooling functions to more than compensate for the lack of certain flexibilities. Every prospective user should carefully investigate the package on a trial basis in-house. The satisfied users contacted by Datapro included several who had also tried IBM's POWER as well as a number of non-IBM alternatives. For mature DOS users, not frightened by the small size of Jason Data Services, Sprint can well be a "best buy" and should receive serious consideration. □

- Automatic forms alignment routine, support for printer/punch forms changes, and support for special printer carriage tapes.
- Shared spool file ability, allowing output for both the card punch and the printer to share the same disk extent for more efficient space utilization as well as reduced arm movement.
- Priority interrupt feature, allowing output from a priority job to be spooled while holding output from prior jobs.
- Device address independence, which allows users to assign logical addresses to devices instead of having to punch new / / ASSIGN cards in order to run a job in a different partition. Sprint automatically substitutes actual addresses for logical addresses at execution time, dynamically based on the partition being used.
- A relocating loader that has features seldom found in "relo" packages. It does not cause the user program to increase in size (and thus render overlays inoperable); it handles the ANS COBOL SORT verb; it requires no changes in user JCL; it is faster than "straight"DOS; it will tell you when your program was last used, when it was cataloged, and more; and it is transparent in operation.
- An effective partition balancing algorithm that samples the system approximately every five seconds and adjusts priorities so that users get equal CPU time.

PERFORMANCE: Although Sprint can run in a minimum of memory (see HARDWARE/SOFTWARE REQUIREMENTS, below), its performance improves dramatically with allocation of more main storage. For example, Sprint Option D (see PRICING, below) using 6K bytes can process about 5500 lines per minute spooled to disk, but with 12K bytes allocated, the same workload runs at up to 10,000 lines per minute spooled to disk. Please refer to the USER REACTION section of this report for additional commentary on Sprint performance.

HARDWARE/SOFTWARE REQUIREMENTS: An IBM System/360 or 370 operating under DOS or DOS/VS with a multi-programming supervisor, command chain retry, timer support, multiple wait, and a minimum of 10 logical unit blocks (LUB's) in the Sprint partition. If multiple independent partition spooling is used, asynchronous processing support must also be included in the supervisor. Tape-oriented Sprint requires a minimum of 2K bytes of main storage, and disk-oriented Sprint with only printer and punch spooling (output-only spooling) uses a minimum of 4K bytes. Multiple-partition Sprint requires 6K bytes per partition to be spooled plus 2K bytes for Marvel, the multiple-partition feature. The foregoing are requirements in a real-memory DOS system. Under DOS/VS, 2K bytes of real memory are required per partition spooled, plus 2K bytes for Marvel, if used. In all Sprint systems, the optimum amount of main memory to use is 12K bytes of real main storage per partition to be spooled.

Many Sprint users are running the full system in about 8K bytes, as this improves the throughput greatly while still holding the size down to a reasonable figure. The minimum Sprint card reader file requirement is four 2314-type cylinders; minimum for the printer/punch file is 10 cylinders; and at least 2 tracks are required for job accounting.

PRICING: For a monthly rental of $95 or a perpetual license fee of $2,200 plus $150 for annual maintenance, Sprint can be obtained in one of the following configurations: Option A – DOS reader, printer, and punch spooling (min. 6K); Option D – DOS printer and punch spooling (min. 4K); or Option K – DOS/VS single-partition reader, printer, and punch spooling (min. 2K). For a monthly rental of $180 or a perpetual license fee of $3,000 plus $150 for annual maintenance, the following multiple-partition versions of Sprint can be obtained: Option F – DOS reader, printer, and punch spooling (min. 6K per partition spooled), device address independence, and Marvel (2K); Option H – same as Option F, plus a relocating loader (2K); or Option L – same as Option F, but for DOS/VS, and without the device address independence feature, since it's not needed (requires 2K bytes of real memory per partition spooled, plus 2K bytes for Marvel). The monthly rental prices include maintenance. Job accounting routines normally included in Sprint at no charge. can be purchased separately for $250. Jason's policy on any price increases is that only new accounts are affected.

FIRST INSTALLATION: October 1971.

CURRENT USERS: 120 as of December 1973. ■

OS/DOS Job Accounting Report System
Johnson Systems, Inc.

MANAGEMENT SUMMARY

Johnson Systems' Job Accounting Report System is one of the most effective of a considerable number of computer utilization reporting systems currently available. Much of this effectiveness is due to the system's flexibility – not just to accept System/360 or 370 time-accounting data from all of the principal recording systems in use today, but also because of its powerful Report Generator.

In fact, the Johnson system differs from its numerous competitors mainly in allowing the user full freedom to design his own output reports. The potential value offered by this capability, unfortunately, is often lost upon users who are selecting a job accounting system to keep system utilization records for the first time. At that evolutionary point in an installation's development, users seldom appreciate the full range of benefits that can accrue from properly displayed and interpreted time accounting data. What most first-time job accounting system users want is simply enough billing information to prorate computer charges to respective user departments – a fundamental but nonetheless merely preliminary application of the Johnson system.

It should be pointed out, however, that the control cards supplied by the user to specify the reports use a simple, fixed format. This format is considerably less complex than the RPG statements required to produce comparable reports.

In addition to sporting one of the most comprehensive cost distribution schemes in the industry, the Johnson system also supports the following types of activity:

- System resource utilization analysis, in which summary information regarding individual hardware component utilization is presented. This information can be used to project the impact of anticipated hardware changes, bottlenecks due to conflicting I/O assignments, etc.
- Throughput analysis, in which the characteristics of individual programs are identified (I/O or CPU "boundedness", channel utilization, etc.) for subsequent re-evaluation of resource management procedures.
- Program development monitoring, in which test shots, data set access, and other program development/testing criteria specific to a programmer are examined in depth.

Another rather surprising use of the Johnson system is to provide a solid first cut at true system performance ▷

The Johnson Job Accounting Report System is one of the most versatile and powerful of its type. A flexible report generator equips the package for a dual role as a system performance measurement aid as well as a basic accounting and cost distribution system.

CHARACTERISTICS

SUPPLIER: Johnson Systems, Inc., The Grant Building, Westgate Research Park, McLean, Virginia, 22101. Telephone (703) 893-2222. (Note that the Johnson Job Accounting Report System should not be confused with "DOSJARS", a competitive DOS job accounting system from Compunetics Corporation.)

BASIC FUNCTION: The Johnson Job Accounting Report System is a full-fledged job accounting system written in COBOL and Assembler language for IBM System/360 or 370 computers. A comprehensive report generator is incorporated into the system, allowing extensive flexibility for output report format and content. Both OS and DOS time accounting data can be entered in a single run. In addition to job accounting, the system can also be used as a first-level tool to measure overall system performance and individual peripheral device utilization.

OPERATION: The Johnson system is available in two basic versions: one for DOS only, and one to handle both standard DOS and OS System Management Facility (SMF) data. Two types of DOS input can be accommodated: Grasp's Accounting Macro data (Report 70E-760-01), and basic DOS Job Accounting data. The Johnson system can also operate in harmony with IBM's POWER and Universal Software's ASAP (Report 70E-879-01) spooling systems through the DOS interface routine.

The Johnson system also provides a DOS subsystem to edit the time accounting data and/or combine "foreign" data into the accounting data base (e.g., keypunch charges associated with specific jobs, programming man-hours, decollating costs). A combination of five different billing algorithms (each applicable to one or more computers and/or types of foreign data) can be supported. An extension of the number of algorithms to about 25 is planned. This "Data Base Maintenance Capability" is accomplished through a Johnson Data Management Utility program that accepts add, change, and delete cards for manipulation of the basic DOS input time accounting data, including the removal of user-specified records that are not wanted in the output reports.

A powerful Report Generator is provided to produce up to seven different reports on a single pass of the input data, each with unique format and content specifications. Additional report specifications can be entered during subsequent runs. All detailed data editing, sorting, and formatting functions are performed in the same job step as the actual report generation. Data from up to five differently priced computer systems can be entered into the Report Generator in a single run, and the resulting accounting information can be combined or reported separately. In the OS version, for example, where several DOS systems ►

▷ measuring. This use is made practical by the report generator. For example, standard time accounting data can successfully be manipulated to produce a daily utilization snapshot, a core utilization summary by job step, hourly turnaround statistics highlighting peak load conditions and unusual turnaround times, and a queue time summary that provides a quick picture of average queue and turnaround times in each job class for a given reporting period. Ordinarily most of the above reports require a hardware probe device plus sophisticated software analysis; but with the Johnson system, such reports can be produced in the same processing run as the mundane billing information.

Users contacted by Datapro report a high level of satisfaction with both versions of the Johnson system, especially after using the system for a while and developing a full appreciation for the advantages of the flexible output reports. In particular, the OS version (which competes against a $350-per-month, 12-month-payment IBM Field-Developed Program, the SMF Selectable Analyzer, 5798-AAR) is a particularly strong job accounting system in view of its system measurement capabilities.

Users report that the DOS version, on the other hand, is strongly enhanced by its "foreign" data for total data processing cost chargeback, in addition to both the flexible report and system measurement capabilities. The "starter" report formats and easy-to-understand documentation were also described as very helpful.

No modifications to IBM software or operating procedures are required for either version.

The Johnson Job Accounting System is available for VS (virtual storage) operation, and can be used to produce summary reports of paging activity, including associated overhead time, etc. Thus, DOS/VS or OS/VS users should be able to employ the system measurement capabilities of the package to a significantly greater degree in a VS environment than has been the case in a "real" environment. □

➤ and one or more OS systems are being accounted for, all charges to a given customer can be lumped together for billing or kept separate for detailed system utilization studies.

A number of specific capabilities are provided to help the user in the following applications:

- Billing – Flexible billing algorithms adjust to priority level, "real" memory utilization, partition/region occupied, elapsed time, CPU time, overhead time, tape and disk operator set-up charges, wait time, EXCP count per device equated to seconds, and unit record I/O data. Each of these elements can be assigned a user-specified weighted rate or excluded from billing. Charges can be broken down by job, job step, or other summarization level as specified by the user. Each computer can be assigned a unique billing algorithm within a single report. All billing parameters are specified on control cards.

- Cost Distribution – A rate determination analysis report can be produced that allows the total operating cost of the computer installation to be spread across all programs, depending upon resource utilization. This analysis permits the billing algorithm and parameters to be adjusted manually, so that a fair distribution of costs can be made and a given profit level can be assured if the computing department is a cost center.

- Throughput Analysis – Identifies inefficient or improper I/O device assignment for given programs; assists in scheduling by identifying relative component utilization to avoid inefficient setups and conflicting I/O assignments under multiprogramming.

- Device Utilization – Allows reassignment of data sets to provide more effective use of available tape or disk drives, etc. In several cases, the device utilization study has revealed to Johnson's users that a given peripheral could be eliminated following a minor data set redefinition.

For OS SMF data, Johnson Systems provides a "starter set" of report control cards that define 14 standard reports to help the user analyze his accounting data. For DOS, basic report control cards for eight standard reports are provided.

For both the DOS and OS versions, Johnson Systems has unusually clear and easy-to-comprehend documentation, ensuring good understanding of the systems early in the game.

HARDWARE/SOFTWARE REQUIREMENTS: Both versions of the system will run on an IBM System/360 or 370 computer under DOS or OS (or their VS counterparts). The Report Generator runs in a partition of 38K to 54K bytes under DOS and in a partition/region of 74K to 100K bytes under OS, depending upon input file blocking factors. For standard DOS accounting data, an Interface Routine of 3500 bytes is read into the partition being reported upon after the phase ends; thus, no additional memory is required for the Interface Routine. Also, for standard DOS accounting data, the JSI Dump Utility runs as a standard batch job in a 2K-byte partition to transfer data from the disk accounting file to an accounting tape; and the JSI DOS Data Management Utility requires a 38KB partition to edit the input and augment it with foreign data if required.

PRICING: The combined OS and DOS version is available on a purchase basis for $3,450 or on a rental basis for $245/month. The DOS version costs $1,950 on a purchase basis, $125/month on a rental basis, and $195/month for a cancellable lease/purchase agreement. The lease/purchase plan is cancellable on 30 days notice and applies 2/3 of the payments over a 15-month term toward purchase, thus providing a full payout of 150% of the purchase price at the end of the term.

INITIAL INSTALLATION: December 1971.

CURRENT USERS: About 65 for the DOS version and 8 for OS. ■

Accounts Payable System
Keane Associates, Inc.

MANAGEMENT SUMMARY

The Keane Associates Accounts Payable System is a good choice for users of small IBM System/360 or 370 computers who carry accounts with a large number of vendors.

The system runs in 32K bytes of core and requires only one disk and two tape units in addition to DOS requirements. The Vendor Master File is maintained on disk in an indexed sequential organization, permitting efficient processing of a low-activity file (large number of vendors compared to the daily number of invoices).

The operation of the system is built around multiple files and 16 COBOL processing programs. A total of five files are permanently maintained, with the generation of several temporary files on disk as the need arises in the operation of the system. This is a common consequence of limiting the amount of core storage used, but it does lead to some awkwardness in operations.

Running the system can be broken up into ten separate steps. Typically, half of the steps are performed daily, and half monthly.

All the normal reports you would expect to see are produced, including Cash Requirements, Check Register, Expense Distribution, and Trial Balance. A better-than-average complement of registers, listings, and control reports are provided, probably as a result of the ease with which they can be obtained due to breaking up the system into so many operational modules.

There are also a couple of reports you might not expect to see. The Accounts Payable Activities Report summarizes in one place new invoices, paid invoices, and open invoices for a particular accounting period. The vendor is identified along with any checks that were written. Invoices containing items suitable for reporting on 1099 forms (outside services for which no income tax was deducted) can be identified and a separate listing made.

The system provides a reasonable degree of flexibility in handling accounts payable. Pre-paid items can be identified and included for distribution but suppressed from the check-writing function. Separate vouchers are prepared for each invoice; invoices can be separated into individual items for cost distributions by the simple expedient of preparing separate vouchers, a non-elegant but workable method. The only flaw is that there is no automatic check on total invoice amounts.

There are several other unusual features in this package. For example, the Vendor Master File contains space for ▷

This package is a good bet for the small computer user because it requires only 32K bytes of core storage. It also has some unusual features oriented toward payment for services by individual consultants and foreign nationals.

CHARACTERISTICS

SUPPLIER: Keane Associates, Inc., 36 Washington Street, Wellesley Hills, Massachusetts 02181. Telephone (617) 237-2621.

BASIC FUNCTION: To automate the writing of checks due for accounts payable and produce a series of related reports. The system is written in COBOL.

OPERATION: The complete system can be broken down into 10 operations. Typically, half of these operations are preformed daily, and half monthly.

The system is based upon three principal master files: Vendor, Voucher (open invoices), and Distribution (assignment of costs to particular accounts). Three other files are also maintained. A Daily Transaction File is created to collect data from multiple input batches during the day. A Distribution History File is maintained to allow the generation of Expense Journal Reports over any back period. An Outstanding Check File permits check reconciliation.

The system includes 16 COBOL processing programs and 13 sort and utility programs. All files except the Vendor Master File are sequentially organized and may reside on either magnetic tape or disk. The Vendor File is indexed sequential and must reside on disk. Typically, the five files other than the Vendor File reside on tape. Interim files, mostly resequencings, are typically held on disk. All sorts are disk-based.

The first three operations involve editing the input and updating the Voucher File and Distribution File. The next two operations involve preparation of the Cash Requirements Report and printing of the checks. These two operations are separated by a manual check of the cash requirements and decisions about holding back payment on certain invoices or releasing for payment invoices previously held.

An unusual technique is used for the hold/release mechanism. As part of the operation that prepares the Cash Requirements Report, a set of cards is punched for each invoice. A Hold card is punched for each invoice not presently being held, and a Release card is punched for each held invoice. Implementing the decisions about holds and releases then merely involves selecting the appropriate cards and feeding them into the check-writing operation; no additional manual keypunching is required.

The five monthly operations include preparing the Expense Distribution Report, check reconciliation, preparing the Accounts Payable Activity Report, preparing the Trial Balance, and preparing the 1099 listing. ➤

Accounts Payable System
Keane Associates, Inc.

accommodating a payee name and address and a depositor name and account number in addition to the vendor name and address. This is helpful for paying foreign nationals who want their checks deposited directly in a foreign bank, or to make payments to a company being held in receivership. (Most small companies will not make use of this feature.) Another example involves holding back payment on selected invoices or releasing invoices previously held. As a part of the operation that prepares the Cash Requirements Repot, a card is punched for each invoice to be paid. A hold or release can be put on any invoice by selecting the appropriate card and inputting it to the check-writing operation. Still another unusual feature is the standard procedure of keeping non-negotiable copies of the checks written as proof of payment. (Most businesses are satisfied with the cancelled check.)

These unconventional features are present because the Keane Accounts Payable System was developed for a specific user and generalized somewhat into a commercial package. (This is not necessarily a criticism, because most proprietary packages originate this way.)

All in all, the Keane system looks like a good bet for small System/360 or 370 installations. Users contacted by DATAPRO 70 report that the system is generally easy to work with, although some awkwardness is introduced because of the need to reduce storage requirements. This, however, is a familiar state of affairs for users of small computers, and the excellent quality of the system tends to compensate for any slight inconvenience. □

INPUT: All input is prepared on punched cards in batches. File maintenance entries to correct errors or make changes are included in this input. Transactions are sorted by batch and accumulated on tape. When all batches have been entered for the day, the tape Transaction File is sorted to disk and the accounts payable cycle can begin. Cashed check records are prepared on cards as they return for input into the check reconciliation cycle.

MASTER FILES: The master files carry extensive information.

The Vendor File carries vendor name and address plus a short name (used to conserve line space in some reports), a payee name and address, a depositor name and account number, and year-to-date totals for number of items paid, discounts taken, and credits received, but not net amounts paid. Four lines, each 25 characters, are provided for names and addresses.

The payee and depositor names and addresses facilitate payment of foreign nationals by deposits to an out-of-country bank, and payment to companies being held in receivership.

Each Vendor File record occupies 300 characters, which also allows a modest space for expansion of the information carried.

The Voucher File carries the usual entries for invoice number, payment due date, payable amount, discount information, hold/release code, vendor identification, company and division identification, purchase order number, and the identifying internal voucher number. In addition, check information including number, date, and amount is also included. An unusual entry is a provision for including information for a check issued against this invoice that has been voided. The record for each voucher (invoice) totals 155 characters, including a small space for expansion.

The Distribution File includes the information necessary to identify each invoice, vendor, amount, service or product, and account to be charged. The account number is 15 digits, with no standard provision to identify subaccounts separately. Each record in the Distribution File occupies 100 characters, with room for expansion beyond standard entries, and pertains to one invoice.

OUTPUT: A total of 15 different reports can be generated. Seven of these are registers, listings, and control reports showing all items processed at various stages and/or control totals; these reports form comprehensive audit trails. The other eight reports, including the checks as one type, form the information available to management for showing a picture of the cash position of the company in regard to accounts payable. Five of these reports are normally associated with an accounts payable operation: Cash Requirements, Checks, Expense Distribution, Journal of Outstanding Checks, and Trial Balance.

The Accounts Payable Report is actually a summary of information found in other reports and listings. For a specified accounting period, newly opened vouchers (invoices), paid vouchers, and vouchers remaining open are presented for each division and company. Complete identifying information is shown for each item, including check number and date.

The remaining report is a listing of items on invoices applicable for reporting on a 1099 form. These items are identified during the original transaction input.

HARDWARE/SOFTWARE REQUIREMENTS: A minimum operational system consists of an IBM System/360, Model 25 or larger, with 32K bytes of core storage, a 1403 Printer, a 2540 Card Read/Punch, one 2311 Disk Storage Drive, and two 2400 Series Magnetic Tape Drives. This configuration does not include the requirements for DOS, under which the Accounts Payable System runs. Use of additional tapes or disk drives to accommodate the various files is recommended to speed processing and minimize operator intervention.

PRICING: The system sells for a flat $5,000. The price includes one-man week of effort for installation and training. The system is delivered in the form of source-code decks (COBOL), and installation includes valid compilation of the user's system.

The company states that updates and any necessary program corrections are furnished to users as a matter of course.

INITIAL DELIVERY: January 1970.

CURRENT USERS: About 10 as of May 1972. ■

DEEP/360 (Data Exception Error Protection) Macro Services Corporation

MANAGEMENT SUMMARY

DEEP/360 is a simple $225 program for the IBM System/360 and 370 that avoids some types of aborts during program testing — thereby saving programmer time, machine time, turnaround time, and programmer frustration.

Programming languages which enable programmers to work at high levels allow most computer programs to be written without getting too deeply involved in the complexities of object code. However, while the languages can avoid the complexities of the various instructions, they cannot so easily avoid the similar complexity of the various forms of data.

In IBM System/360 and 370 computers, there are actually five different forms of data. The same item of data is often wanted in more than one form. For instance, if gross pay is to be calculated, it must be in a form that can be used for computation. But if it is to be printed, then it must be in a form suitable for printing. As the two are not the same, then not only must a programmer give two names and descriptions to the same item of data; he must also remember to arrange for all the necessary moves and conversions between the two forms.

However, programmers, who are naturally inclined to concentrate on the work that has to be done to produce the basic solutions to their problems, can easily omit a few of these moves or descriptions. When this happens, the program – whether written in detail by the programmer or, more likely, by a compiler – will use, say, the add instruction that applies to a binary representation of a number which is ready for printing. The machine notices the mismatch – and stops processing the program. It issues a "Data Exception", saying that the data it was given to work with simply did not match the instruction.

There are a number of other frequently encountered cases where the data simply does not match the requirements of the instruction. One is where an attempt is made to divide by zero. Another is where the multiplication operands are so large that the resulting product cannot be fitted into the system. Others can occur when blanks are used where zeros are expected, or when work areas are not cleared out completely.

On the System/360, the unavoidable penalty for these frequent errors and omissions is that the program is aborted. During production runs, this action is usually desirable. However, during testing it is often desirable to allow the program to continue in operation, even with dubious data, provided that the programmer knows that a data error has occurred. This way, what might otherwise be a series of non-productive tests, with each one aborting when the first data exception is found, can be compressed into one test that uncovers a number of such errors.

Although the System/370 hardware permits users to ignore data exceptions, some users question the wisdom ▷

CHARACTERISTICS

SUPPLIER: Macro Services Corporation, 373 Park Ave. South, New York, New York 10016. Telephone (212) 697-6711.

BASIC FUNCTION: DEEP/360 (Data Exception Error Protection) provides automatic continuation of operation of COBOL and BAL programs under test on a System/360 or 370 when data exception conditions occur. Arbitrary correction methods are applied in some cases; in others the instructions are simply bypassed. In all cases, console reports are made for the first error at a given location.

OPERATION: DEEP/360 is a BAL program which uses the STXIT macro to trap data exception errors. The program prints the core location of the instruction at fault, "corrects" the data which caused the data exception, notifies the user of what corrective action has been taken, re-executes the instruction at fault, and proceeds with the user's program. DEEP/360 is a relocatable module which must be linked to the user's program. Main storage required is approximately 3,000 bytes.

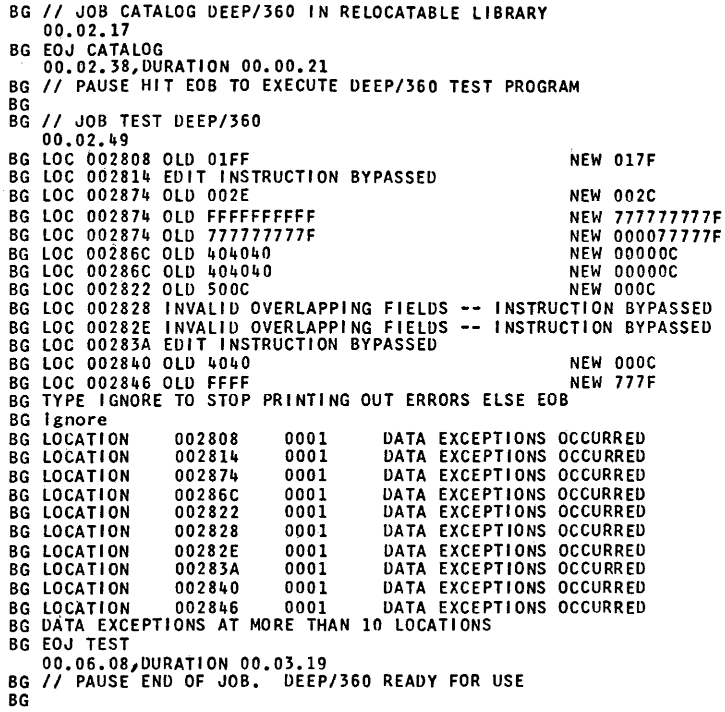
DEEP/360 must be called twice whenever it is used, once as a beginning housekeeping step and once as a closing step. The first call sets up the supervisor linkage, and the second call causes end-of-job error statistics to be displayed.

Each time a data exception occurs, DEEP/360 will assume control and examine the data which caused the check. If DEEP/360 cannot find the error or cannot fix it, a message to that effect is printed and the offending instruction is bypassed. When possible, however, the data is altered and the offending instruction re-executed. In this case, both "before" and "after" data fields are displayed. Rules for "fixing" data can be summarized as follows:

- Packed fields missing signs assume positive zones.
- Packed fields containing digits greater than 9 have the 8-bit stripped from such digits.
- Packed fields with hexadecimal "40" in the last byte are changed to positive zero.
- High-order digits of a multiplicand field are changed to zero as necessary to prevent product overflow (Multiply instruction only).

For the first ten unique core locations at which data exceptions occur, one and only one console message per location will appear; recurring errors in those ten locations will be counted but not printed. Should errors occur in more than ten different locations, messages for the eleventh, twelfth, etc. location will be printed every time an error occurs there. After any ten console messages, the user has the option to suppress all further messages. At end-of-job, the error counts from the first ten error locations are displayed. ➤

DEEP/360 (Data Exception Error Protection)
Macro Services Corporation



```
BG // JOB CATALOG DEEP/360 IN RELOCATABLE LIBRARY
   00.02.17
BG EOJ CATALOG
   00.02.38,DURATION 00.00.21
BG // PAUSE HIT EOB TO EXECUTE DEEP/360 TEST PROGRAM
BG
BG // JOB TEST DEEP/360
   00.02.49
BG LOC 002808 OLD 01FF                            NEW 017F
BG LOC 002814 EDIT INSTRUCTION BYPASSED
BG LOC 002874 OLD 002E                            NEW 002C
BG LOC 002874 OLD FFFFFFFFFF                      NEW 777777777F
BG LOC 002874 OLD 777777777F                      NEW 000077777F
BG LOC 00286C OLD 404040                          NEW 00000C
BG LOC 00286C OLD 404040                          NEW 00000C
BG LOC 002822 OLD 500C                            NEW 000C
BG LOC 002828 INVALID OVERLAPPING FIELDS -- INSTRUCTION BYPASSED
BG LOC 00282E INVALID OVERLAPPING FIELDS -- INSTRUCTION BYPASSED
BG LOC 00283A EDIT INSTRUCTION BYPASSED
BG LOC 002840 OLD 4040                            NEW 000C
BG LOC 002846 OLD FFFF                            NEW 777F
BG TYPE IGNORE TO STOP PRINTING OUT ERRORS ELSE EOB
BG Ignore
BG LOCATION    002808    0001     DATA EXCEPTIONS OCCURRED
BG LOCATION    002814    0001     DATA EXCEPTIONS OCCURRED
BG LOCATION    002874    0001     DATA EXCEPTIONS OCCURRED
BG LOCATION    00286C    0001     DATA EXCEPTIONS OCCURRED
BG LOCATION    002822    0001     DATA EXCEPTIONS OCCURRED
BG LOCATION    002828    0001     DATA EXCEPTIONS OCCURRED
BG LOCATION    00282E    0001     DATA EXCEPTIONS OCCURRED
BG LOCATION    00283A    0001     DATA EXCEPTIONS OCCURRED
BG LOCATION    002840    0001     DATA EXCEPTIONS OCCURRED
BG LOCATION    002846    0001     DATA EXCEPTIONS OCCURRED
BG DATA EXCEPTIONS AT MORE THAN 10 LOCATIONS
BG EOJ TEST
   00.06.08,DURATION 00.03.19
BG // PAUSE END OF JOB.  DEEP/360 READY FOR USE
BG
```

Console printout illustrating the use of DEEP/360 to locate a number of data exceptions during a single run. Note that the action taken in each case was printed as it happened until printing was suppressed, and that a list of 10 locations was printed at the end along with an indication that more than 10 data exceptions occurred.

▷ of doing so, preferring to know of such problems via an aid such as DEEP/360.

Evaluation of DEEP/360 should start with its price. At $225, the question is not "Is it worth it?" but instead, "Will anyone use it?" The next step is to find out whether data exceptions are aborting any of your test runs. If they are – even if there are only novice programmers concerned – then DEEP/360 is likely to be well worth its modest price.

USER REACTION

Users of this inexpensive 3K program consistently rate their overall satisfaction with it as "good" or "excellent." They rate the vendor's service and support in the same way. A user points out that the package eliminates data exception cancels during test runs, thus permitting the user to gain more information from a test run and to save machine time. Although sales of the package have slowed since the System/370 was introduced, there are System/370 users who find merit in DEEP/360, preferring to know about data exceptions that the hardware can let them ignore. □

➤ HARDWARE/SOFTWARE REQUIREMENTS: DEEP/360 operates on any BAL or COBOL System/360 or 370. About 3K bytes of memory are required by the program. PL/1 and FORTRAN programs are not compatible with DEEP/360.

PRICING: $225 buys a single copy of the routine for an installation. A corporation wanting unlimited use is charged $1,500. Support consists of a user's manual and complete listing of the program.

INITIAL DELIVERY: The program came into use internally in May 1969, and was first used by an outside installation in September 1969.

CURRENT USERS: 500 as of mid-1973. ■

DATAMACS
Management and Computer Services, Inc.

DATAMACS facilitates the creation of test files on any IBM System/360 or 370 computer for programs written in COBOL or any other source language. Record and file creation can be quite flexible, and user preparation is minimal with COBOL.

MANAGEMENT SUMMARY

Program testing isn't nearly as interesting as coding, and a programmer is likely to be pretty sick of his program by the time the test phase begins. Furthermore, programmers don't seem to enjoy checking out the work of others.

A major program testing headache is generating test files. There are two basic approaches to this problem, both of which have been automated by one or more manufacturers' or proprietary packages: (1) create an imaginary set of test records, or (2) select an appropriate set of records from a live file. DATAMACS takes the first approach. It is most conveniently used with COBOL programs, but creation of test files for programs in other languages can be managed.

Basically, DATAMACS control statements, which are COBOL-like directives, are interspersed in the Environment and Data Divisions of a COBOL source program. The source program is temporarily segregated onto tape or disk. Based on the DATAMACS specifications and the COBOL source-program file and record descriptions, data values are inserted into field locations within records which are grouped in files. Normally, the program is then compiled and executed using the test files.

The DATAMACS statements are powerful and easy to formulate. However, it is just a little difficult to readily visualize the results once the coding has been written. The strength of DATAMACS is the convenience with which it is possible to set up a complete test file. Different types of records to exercise all program procedures, as well as incorrect records to test the error control procedures, can easily be coded.

Using DATAMACS has advantages whether your programmers are beginners or real hot-shots. Because of the ease with which test files can be created, managers can feel more confident that programs will actually be tested—a very real cause for concern in many installations. If your programmers are the conscientious type and will test their programs thoroughly, no matter what, then use of DATAMACS will free them from some of the drudgery of programming for more creative tasks (such as figuring out why the program didn't work, even with good test files).

MACS has developed a very strong selling point in some installations by creating test files for existing production programs and then finding bugs in these programs—clear evidence that a more rigorous testing program is needed.

The program is written in such a way that new modules can be added without a great deal of difficulty to take

CHARACTERISTICS

SUPPLIER: Management and Computer Services, Inc., Suite 790, Valley Forge Plaza, Box 826, Valley Forge, Pennsylvania 19482. Telephone (215) 265-2910. MACS also has sales/technical representatives in Brazil, England, and France.

BASIC FUNCTION: Generates data files for testing COBOL programs according to specifications set up by the programmer. The program is written in BAL and designed for "load-and-go" COBOL operations on an IBM System/360 or 370 operating under DOS, OS, or their virtual storage counterparts, but free-standing files and test files for non-COBOL programs can also be generated.

OPERATION: Normal operation of the DATAMACS program is to read a COBOL source program that includes special control cards interspersed in the Environment and Data Divisions, to segregate the control cards and COBOL source code, to generate data files based on the specifications contained in the control cards, and then to transfer control to the COBOL compiler for compilation and execution. Any type of file can be created because DATAMACS uses the IBM access methods to write the files. The process can be halted following test file generation to create free-standing files.

DATAMACS can be used to create test files for programs not written in COBOL by using a skelton outline of the Environment and Data Divisions, along with the DATAMACS control cards, and halting the process after the generation of test files.

Specification of the data to be created for the test files is effected by three types of control sentences: file, data, and record.

A DATAMACS File-Control sentence follows the SELECT sentence for each file defined in the Input/Output Section and identifies the number of records to be generated for that file. The number of test records to be printed can also be specified, and printing can be in either hexadecimal or EBCDIC form.

Data-Control sentences follow the data names and pictures for each field within a record. A variety of generator clauses provide a great deal of flexibility in setting up data values. Other clauses facilitate setting up control-break conditions and multiple-use areas (redefining). When no control card is submitted, standard defaults are placed in the field. The user can also specify random values for defaults in a GENERATE clause.

The nine types of generator clauses are:

- Constant—data items inserted in a data field are literals supplied in the specifications.
- Sequence—data items inserted in the specified field in successive records are incremented; ascending or descending order, beginning and ending limits, and size of increment can be specified.

DATAMACS
Management and Computer Services, Inc.

▷ care of special record structures. This has been a strong selling point—the willingness of MACS to modify and add to the program. First released at the end of 1969, DATAMACS is now in its sixth release. New releases are automatically distributed to existing customers at no extra charge. Generally, new releases add to the flexibility of record creation as well as correcting bugs that have been found. Bugs are corrected immediately for those users who find them on their own.

USER REACTION

Users contacted by Datapro for the June 1972 DATAMACS report all reported satisfaction with the product, and found that it performed as claimed by the vendor. One user complaint, non-relocatability of DATAMACS, was cleaned up in the sixth release.

Rechecking the product's reputation now with users of virtual storage systems, Datapro found the same high degree of satisfaction with DATAMACS and its vendor. It was said that DATAMACS under VS runs properly, required no changes to do so, and shows only slight performance degradation due to paging. The package is still most popular with COBOL users, who find that most of the DATAMACS coding is automatically handled by COBOL. Users also like the way the test data stays with the source code.

One user rates DATAMACS as the best test data generator available at this time. The only complaint that was voiced is a feeling that more OS orientation is needed in the documentation. DATAMACS should be considered by all 360/370 COBOL users desirous of improving their program testing. □

➤

- Random—the value of data items inserted in successive records is generated randomly; numeric items are confined within an upper and lower bound, while alphabetic and alphanumeric items are drawn from the respective character sets.
- Cluster—data values inserted in successive records are grouped about a specified numeric value; useful for situations where a known ordering is in effect (as, for example, in clustering weekly hours worked about the value 40 when testing a payroll program).
- Compute—data values are generated by adding, subtracting, multiplying, and/or dividing two or more data items together; handy for automatically generating extensions or other derived results.
- Mix—allows any combination of the first five generator verbs to be used to control the generation of data items; normally, different records would contain data items generated by different methods; useful for generating "incorrect" fields to test error detection and recovery procedures within a program.
- Date—generated in specified format, with default to MM/DD/YY.
- Default—standard or random values generated.
- If—identical in format to the COBOL IF; allows conditioned generator statements to replace COBOL imperatives.

An EVERY clause can be used with any of the generator clauses to cause a designated field to assume a new value only at specified intervals. In general, a Times and a Repeat specification are available to allow convenient repetition of a specific data item or groups of data items. Another clause within the Data Control generator sentences permits check-digit generation for numeric items generated by one of the generator clauses; the standard used is modulo-10, but others could be implemented rather easily.

The Record Control sentence follows the FD (file description) sentence and precedes the record description. This sentence controls which records are to be written into the file and in what order. Again, Times and Repeat specifications facilitate coding.

To facilitate the interpretation of the source-program output during testing, a Total-Control sentence can be used to generate and print the totals of the data values used in all occurrences of selected fields. Values can be summed, batched, or moved and saved.

COPY and BASIS specifications used in the COBOL source program are interpreted by the DATAMACS program. This allows DATAMACS control statements to be included in coding stored in the library.

Two users exits are provided. One is intended to allow writing of non-standard header and trailer records for each generated file; the other allows modification or deletion of a generated record prior to output to tape.

Test files are generated on tape or disk storage. The source program and control statements are typically entered via punched cards. The source COBOL program is transferred to a work file, which can be on magnetic tape or disk. Diagnostic messages appear in the source-code and control-statement listing printed by DATAMACS. The diagnostics have been improved and increased in number from 12 to 112 in Version 6.

HARDWARE/SOFTWARE REQUIREMENTS: Versions of DATAMACS are available for IBM System/360 or 370 computers under DOS, DOS/VS, OS/MFT, OS/VS1, or OS/VS2. The smallest DOS (4-file) version can run in a minimum 24K-byte partition. There is a standard (10-file) DOS version that can run in a 48K partition. OS versions (10-file) will operate in as little as a 65K-byte partition or region. Any of IBM's COBOL compilers can be used under any version.

PRICING: DATAMACS is provided to users on a 99-year license plan for a one-time charge of $5,500 (DOS or DOS/VS) or $6,500 (OS or OS/VS). This includes program maintenance for the first year, training, and such updates as are included in future releases of the basic package. Thereafter, annual maintenance is priced at 10% of the original selling price. The Check Digit Routine costs $600 extra. A free trial period is available to those who wish to look before leaping. Discounts are available for multiple users.

Government users can obtain DATAMACS (GSA number GS-00C-00164) for 12 monthly payments of $435 (DOS or DOS/VS) or $515 (OS or OS/VS). This arrangement includes a 30-day cancellation provision.

INITIAL DELIVERY: December 1969.

CURRENT USERS: Approximately 300 worldwide as of July 1973. ■

PULMACS
Management and Computer Services, Inc.

MANAGEMENT SUMMARY

Keeping track of source programs seems a simple enough task to do manually—until you start talking about several hundred programs and/or very large programs. The mechanics of storing and handling card decks are familiar operations in almost all EDP installations. But this approach is not particularly efficient, convenient, or neat. A much better approach is to use the computer to manage the handling of this data—that's what it's for, after all.

With PULMACS (Program Update Librarian) from MACS (Management and Computer Services, Inc.), the System/360 or 370 DOS or DOS/VS user with a small to medium tape- or disk-oriented system can efficiently manage a catalog of a few hundred programs. What's more, PULMACS is about the least expensive program available that is designed for this function.

Although the program has few frills, it permits surprisingly flexible operation. Any type of 80-column data can be cataloged, from COBOL source programs to JCL card sets to program data. Special accommodation has been made to provide more compact storage through the elimination of sequence numbers and identification fields in a variety of source languages, as well as through the elimination of blanks in all types of data.

A number of auxiliary functions can be performed, such as copying programs to punched cards or to a printer or building a file of programs to be compiled.

The facilities provided for modifying a program are simple to specify and very straightforward to use. However, several verbs are supplied that differ only slightly in use; e.g., INSERT causes card images to be inserted after a particular statement, while CHANGE causes a statement to be replaced and card images to be inserted—and there are two more verbs that cause changes. All these operations are necessary to handle various modification problems, and it should take only a short time for a programmer to become familiar with the facilities.

As mentioned, PULMACS contains few frills. Only one version of a program can be maintained in a file unless the program is recataloged under a different name. Source records are automatically resequenced each time a change in the source program (or other data set) is made. This assures complete and unambiguous numbering of each source record so that any statement can be identified and referenced in a later modification. However, this approach means that a printed listing should be made each time a program is modified so that later references for additional modifications can be made accurately. This will add to the time required for running PULMACS. The direct-access version of PULMACS also has a data compaction capability to make maximum use of available disk space.

Among users of the sequential version, MACS states that the number of programs maintained in the library averages about 200 to 300, with the maximum being about 400. ⊳

PULMACS is a simple, straightforward, and inexpensive package that handles source program library maintenance in System/360 and 370 DOS and DOS/VS systems. It is available in sequential and direct-access versions.

CHARACTERISTICS

SUPPLIER: Management and Computer Services, Inc., Suite 790, Valley Forge Plaza, Box 826, Valley Forge, Pennsylvania 19482. Telephone (215) 265-2910. MACS also has sales/technical representatives in Brazil, England, and France.

BASIC FUNCTION: PULMACS, written in BAL, is used to create and maintain a file of programs in source language or other 80-column data for an IBM System/360 or 370 computer operating under DOS or DOS/VS.

OPERATION: PULMACS permits a user to maintain either a sequential or direct-access file of data sets. Each data set can be a source program, program data, JCL set, or object program. The only restriction is that the information to be filed must be in 80-column card image form. Once a file is created, various manipulations can be performed on each data set. Multiple additions and changes can be made in one pass of the file. For the sequential version, all input must be ordered in the same sequence as the file, which is alphabetical by program (data set) name, and each run causes the entire file to be passed.

A direct-access version, which accepts inputs in any order, operates from disk storage and permits random access to data sets using MACS's own access method (not one of IBM's). The method of accessing data sets is the only functional difference between the direct-access version and the sequential version.

Input to PULMACS consists of two data streams. One is the set of new programs to be cataloged along with changes to existing entries, and the other is the previous copy of the program file. New input can be from a card reader, magnetic tape drive, or a 2311, 2314 or 3330 disk drive. The program file can be held on magnetic tape or a disk drive.

Possible outputs from PULMACS include the updated program file, an error list and program directory, a detailed directory, a job stream file (called the Assembly file) of programs ready for a compilation run, and data sets output on punched cards or listed on a printer. All outputs can be to magnetic tape or to a disk drive. (Printed or punched card output can, of course, use the native device instead.) Only one print file is allowed for, so the detailed directory must be run by itself because the error list and directory is a standard output from all other runs. Outputs can be mixed to a large extent. Temporary changes can be inserted in the Assembly file without causing changes in the master file.

INPUT: Only three types of control cards are required to specify a complete PULMACS run: the Control Statement precedes the entire run and sets certain parameters that hold throughout the run; a Program Statement identifies each data set; and an Update Statement precedes each set of changes for a program.

The Control Statement identifies the type of peripheral device (magnetic tape or disk drive) to be used for the program file and the Assembly file. ➤

PULMACS
Management and Computer Services, Inc.

▷ For users with many more programs to maintain, MACS recommends the direct-access version.

USER REACTION

Of the users contacted by Datapro for the June 1972 report on PULMACS, six had already converted from the sequential version to the direct-access version; all reported complete satisfaction with both versions, even though their needs now call for the direct-access system. PULMACS users contacted for this updated report feel the same way about the product. There are still a few users of the sequential version around, too.

PULMACS offers many of the features often found in more expensive program library packages, and at a fraction of the cost. Naturally, you get what you pay for, and the full facilities of the more costly packages are not presently available in this product. However, with its nice balance of facilities and low cost, PULMACS definitely merits investigation by small to medium DOS and DOS/VS installations with program library needs. □

➤ Assignments, once made, hold for the entire run. The old and new program files must use the same type of device. Three mutually exclusive modes of running can be specifically defined: Create, Compiles-only, and Directory. The Create mode does not permit an input program file. The Compiles-only mode does not include an output program file. The Directory mode ignores everything but the detailed listing of the program file. If none of these modes is identified, then a regular updating run is indicated with both old and new program files, program changes, an Assembly file, and punched or printed listings as indicated by the Program Statements.

The Program Statement that precedes each data set to be catalogued or updated consists of three parts: the control word (PROGRAM) and program name (up to 8 characters), source language identification, and options.

A wide variety of languages can be identified, including BAL (or ALC or ASSEMBLY), COBOL, FORTRAN, PL/I (or PL/I), AUTOCODER, RPG, JCL, OBJECT, and DATA. This parameter is used to control condensation of the source records and is printed on the detailed directory; it can also be used to indicate that a data set is to be eliminated.

The options include NEW, UPDATE, COMPILE, PRINT, PUNCH, NEWNAME, SECURITY, TEMP, GEN, and DATAMACS (Report 70E-593-01). The options permit adding or changing data sets, copying a data set to the Assembly file, and listing data sets on punched cards or the printer. The DATAMACS option (DATAMACS is a test data generator) causes DATAMACS statements contained in COBOL source programs to be deleted when writing a program to the Assembly file. NEWNAME allows changing the name of a data set. All combinations of the options are permitted except obvious incompatibilities such as NEW and UPDATE.

Associated with the UPDATE option are Update control cards which precede individual changes. Five different types of maintenance operations can be performed: INSERT, REPLACE, DELETE, CHANGE, DELADD, and PATCH. Basically these functions allow inserting a number of card images following a specified statement, deleting a single statement, replacing a single statement, changing a single statement and inserting additional card images, and patching by replacing existing statements with the card images in the input stream on a one-for-one basis. In the case of DELADD, multiple statements can be replaced by one or multiple changes. In addition, the INSERT and REPLACE functions can be applied to the output to the Assembly file without changing the permanent program file. Using the Update control cards, a particular group of statements can be output on the card punch or printer.

PROGRAM FILE: The program file is recorded at 1740 characters per block on tape or disk. Within the block, variable-length records are recorded due to compaction. Source records are condensed by eliminating blanks, sequence numbers, and identification fields. Elimination of sequence number and identification fields is predicted on the source language, and will therefore vary depending on the language identification in the Program Statement. Source Information identified as data will be checked for space elimination only. A 5-character internal sequence number is assigned when the program or other data is first cataloged. Every time a change is made, sequence numbers are automatically reassigned.

ERROR LISTING AND DIRECTORIES: The parameters from the control cards are checked for errors and inconsistencies, such as specifying the addition of a source program to the library when a program with the same name already exists or an invalid sequence number in one of the Update control cards. At the end of each run that involves changes to the program file, the errors are listed along with a directory which lists each data set by name and number of records.

The detailed directory includes the program name, source language, date of last action, version/level number, record count, and the options last performed on each data set. The version/level number is incremented each time a change is made to a date set. (Only the most recent version is saved.)

PERFORMANCE: The basic PULMACS program is a sequential processor. Performance should be consistent with the time required to read and write the files connected with the various maintenance and copying functions. Inclusion of many card punch and/or printer copy functions could really slow things down. (Note that printed copies of programs should be requested at least occasionally, because of the automatic resequencing of statement numbers, to simplify correctly identifying sequence numbers in subsequent update operations.) One user with 400 programs in his file finds that it takes about 15 to 20 minutes to make one pass with 2415 tape drives. Faster tape drives would improve performance substantially.

The direct-access version operates at nearly device speed, with significantly faster run times than the sequential version. The company estimates that the break-even point between sequential and direct-access processing for PULMACS would lie at a library size of about 500 programs for the average installation.

HARDWARE/SOFTWARE REQUIREMENTS: PULMACS runs on essentially any System/360 or 370 under DOS or DOS/VS. The program can run in any partition of 17K bytes or more. The Decimal Arithmetic option is required. The direct-access version runs in a partition of less than 16K; the core saving over the sequential version is due to the elimination of the need to store the table for the directory in main memory. Current versions of PULMACS are relocatable.

PRICING: The sequential version of PULMACS sells for a one-time charge of $995 including 3 years of maintenance, and the direct-access version sells for $1495 with 3 years of maintenance. Multiple-installation discounts are available.

INITIAL DELIVERY: Sequential version, March 1971; direct-access version, July 1971.

CURRENT USERS: About 45 as of July 1973. ■

ALLTAX
Management Information Service

MANAGEMENT SUMMARY

ALLTAX is a COBOL subroutine, suitable for incorporation into most payroll programs, that performs the burdensome withholding tax calculations required by the various taxing authorities. It can save considerable programming and maintenance time for companies with employees in many different locations.

Payroll withholding is an essential part of any payroll system. Once upon a time it was a fairly simple part of the computer operations, because the various authorities were forgiving. But now that computers have become familiar objects, it is expected that they make their calculations correctly.

Payroll calculations are not particularly difficult. Indeed, many of the various governmental authorities even provide, for the asking, a set of formulas to use for the operations. Even the developers of the ALLTAX routine do not really claim that there is anything so special about these withholding calculations that it is impractical for you to have your own programmers write them. Their argument is "Release your programmers for more productive work."

To evaluate whether or not it is worth releasing your programmers, therefore, depends on just how much work they are being released from. This is really a matter of history. How often have people needed to change the payroll program? How much time has been spent obtaining and testing the latest changes? How often does some change in tax rates have to be integrated?

Management Information Service, the developer of ALLTAX, points to a long list of major companies which are users of the system. This can be looked at, quite validly, as showing that the system operates effectively. However, to some extent it can also be looked upon as a statement that the type of major company which has branches in many different areas is likely to benefit most from ALLTAX.

This is clearly valid. If in a corporation there is a centralized programming or program development department which handles the payrolls for a number of different locations, then clearly the cost of constantly being on top of what is happening in each of the various localities can be considerable. Here the provision of updated and checked information (state approval letters are included with each package) is valuable in its own right. Probably this is the best justification for investigating the ALLTAX service. (An even better one is being too short of programmers to produce those vitally needed payroll changes.)

A measure of ALLTAX's acceptance is that the developers of several proprietary payroll packages have elected to include ALLTAX rather than develop their own tax package. Included in this group are Management Science America and United California Bank. Users contacted by Datapro report a high level of satisfaction with the system.□

CHARACTERISTICS

SUPPLIER: Management Information Service, P.O. Box 336, Ramsey, New Jersey 07446. Telephone (201) 327-8510.

BASIC FUNCTION: ALLTAX is a subroutine that calculates federal, state, city, and county taxes, written in COBOL for integration into the payroll programs of individual companies. ALLTAX can also handle Disability, FICA, FUT and SUT taxes, and taxes on advance payments.

OPERATION: ALLTAX is entered each time a calculation of tax is desired. It can be entered as often as desired for reciprocal or multiple state tax situations. In this way, all parameters can be changed between the federal and state calculation or between the state and city calculation, if necessary.

Basic parameters which are described below are passed to specific work areas prior to entering the routine. Linkage to the main program is accomplished by an ALTER END-STD-TAX ROUTINE TO PROCEED TO (INSERT RETURN PARAGRAPH LABEL) followed by GO TO STD-TAX-FORMULA statements.

Prior to returning to the main program, the tax amounts to be deducted or accrued are placed in up to four specific work areas for further processing by the main program. The tax amounts furnished by the package can be added to, reduced, or ignored as desired to accommodate additional withholding requested by the employee, percentage-of-time-in-the-state situations, and any other processing exceptions which might be unique to your company.

ALLTAX consists of three distinct segments which are internally linked to form one subroutine: SWIFT-E (Standard WIthholding Formula and TablEs), DISCATAX (DISability, fiCA, fut and sut TAX calculation), and TAXADVANCE (which simply calculates all taxes on vacation advance payments).

SWIFT-E: SWIFT-E calculates all federal, state, city, and county withholding taxes and has three main components: the standard withholding formula which calculates the tax; a table lookup routine which determines the table to be used in the calculation; and a Standard Table for federal, state, city, and county taxes containing the current variable factors required for the calculation.

The Standard Table is variable in size depending on the number of tax brackets specified by the taxing authority. Minimum table size is 41 positions (32 packed-decimal bytes), and average table size is 120 positions (100 packed-decimal bytes) per state.

Each Standard Table has a unique 01-level label. Based on the State Code encountered by the table lookup routine, ➤

ALLTAX

Management Information Service

the Standard Table associated with that code is moved into a common table work area. Exceptions are processed, and the tax is then calculated based on the factors contained in the table work area.

Both the table lookup routine and Standard Tables are completely modular and can be stored off-line on tape or disk if main storage is limited. Standard tables and associated table lookup instructions can be removed entirely to reduce storage requirements if you are not withholding for every state.

The following parameters are required for the SWIFT-E segment: 1-digit pay frequency code; 1-digit marital status code; 2-digit total number of dependents; 8-digit (6.2) gross pay; and 2-digit withholding state code.

For a supplemental pay cycle (separate bonus payments, etc.), an 8-digit (6.2) base pay and a 1-digit supplemental pay cycle indicator are also needed for the aggregate income calculation required by most states.

DISCATAX: DISCATAX calculates those taxes where a maximum annual tax amount is specified. Year-to-date gross or year-to-date tax amount is used for calculation purposes to eliminate penny rounding problems and to stop the tax automatically when the maximum amount has been reached. This segment, by itself, can be used either in the main program to calculate FICA and Disability withholding or in the monthly or quarterly accounting programs to calculate the employer FUT/SUT tax accruals on a monthly or quarterly basis.

Linkage between SWIFT-E and DISCATAX is automatic, so that Federal Income Tax and FICA are calculated before returning to the main program. In the case of Alabama, for example, the Alabama Withholding Tax, Disability Tax, Alabama FUT and SUT would be calculated and made available in four separate work areas for main program. In this manner, FUT/SUT can be accrued on a pay-period basis for cost accounting purposes.

The DISCATAX Segment requires the following additional parameters: 2-digit unemployment state code; 8-digit (6.2) year-to-date gross pay; 5-digit (3.2) year-to-date tax amount (optional except for N.Y. State).

TAXADVANCE: TAXADVANCE utilizes both SWIFT-E and DISCATAX to calculate all taxes on employee vacation advance payments before returning to the main program.

For an employee who lives and works in, say, Gadsden, Alabama, this segment calculates the total amount of tax attributable to the vacation advance for Federal Income Tax, FICA, Federal Unemployment Tax, Alabama Income Tax, Alabama Disability Tax, Alabama Unemployment Tax, and the Gadsden Local Tax and funishes these amounts in seven different work areas for further processing by the main program.

Additional parameters needed for this segment would be: 1-digit vacation advance indicator; 2-digit local tax code; 5-digit (3.2) number of weeks of vacation advance (6 days = 01.200); 2-digit number of local dependents; 8-digit (6.2) total amount of vacation advance.

TESTING: A special testing operation for checking out the system, with an emphasis on checking out new changes, is available. The name of the segment is EASY-RUN, and it consists of the necessary I/O operations to accept punched card input and print the output by state for verification purposes.

Another testing option, the TAXAUDITOR segment, allows verification of the computer formula by printing out a Wage Bracket Table that shows specified wage brackets and tax amounts for zero through twelve dependents. Each table is for a specific tax, marital status, payroll frequency, and residency. Beginning and end points can be specified for the wages to be shown, as well as the bracket increment.

HARDWARE/SOFTWARE REQUIREMENTS: The major core usage is for storage of the tables for the various states and cities. This is strictly related to the number to be handled by the particular payroll. With consideration for all options, including tax advance, the maximum System/360 requirement is 16,000 bytes. The corresponding figure for a Honeywell Series 200 is 21,000 characters; for a Burroughs B 5500, 4,500 words; for a Honeywell 425, 7,000 words. An operational COBOL compiler is also required.

PRICING: The rental price for a complete package including U.S. Federal, State, City, and County taxes is $150 per month for 24 months; program maintenance after 24 months is available at $30 per month.

Alternatively, the package can be purchased. A COBOL program listing only, COBOL source deck only, or both can be purchased. For an additional charge, BAL listings and source decks will be provided. Individual prices are set for each segment. A graduated set of prices applies to the tables for the SWIFT-E segment. The complete package, including program listing, source deck, and a complete set of U.S. tables, goes for $2,250. Logic and tables for Canadian withholding taxes can be added for $600. Maintenance service for new or changed taxes is scaled from $100 per year for less than 9 tables to $300 per year for all U.S. tables. Maintenance for Canadian taxes runs $175 per year.

Management Information Service commits itself to mailing tax additions or changes to users between 15 and 30 days prior to the effective date of the changes.

An optional one-day seminar for installation and training is available for a fee of $200 plus travel expenses. Additional help is available on a per-diem basis.

INITIAL DELIVERY: 1966.

CURRENT USERS: Over 275. ■

Management Accounting and Reporting System Management Science America, Inc.

MANAGEMENT SUMMARY

There are many schools of thought pertaining to how much and what information is useful to a manager. Management by exception, detail management, improvisation, limited responsibility all have their practitioners. Universally, however, the primary information source for management is the entries—revenues and expenses, assets, and liabilities—that go into a company's ledgers. Ideally, it would seem that this information should do more than just inform the stockholders whether or not the company made a profit and how much. The flow of money should, if properly reported in the correct perspective, tell a manager how his responsibility area performed and, therefore, identify the things that need improvement.

The Management Accounting and Reporting System, in its full configuration, can be a powerful aid in preparing good reports for management. It is particularly flexible in specifying groupings of information that cross the lines on the formal organization charts of a company. The management-type reports are heavily comparison-oriented; i.e., comparing this period's results to budgeted amounts and to the previous year's results and budget. Reports can be automatically generated on a daily, weekly, monthly, or other frequency or can be produced on an on-request basis.

The flexibility in designing reports comes from the dual numbering method employed. A seven-digit number is used to identify the account; a second seven-digit number identifies reporting centers. The MSA concept of a "reporting center" is very flexible; each center can represent a department or other formal organizational group, or it can represent a location, product, cost center, product line, etc. Basically, this type of numbering technique allows the convenient specification of accounts or areas for which balances are to be accumulated, groups that are to receive reports, and the linkages for various levels of reports.

MARS is available in several modules. The General Ledger subystem forms the basis for file creation and basic accounting. To this can be added modules for generating and storing budget information, for generating management reports, for cost accounting, and for deriving ratios among related account balances. Separate files are maintained for standard and allocated cost types of account balances. Again, the dual numbering simplifies setting up separate responsibility and profit centers.

Because each item in the chart of accounts needs to be handled separately when the system is installed, it will probably take a little more time and effort than a general ledger system that makes use of account number ranges to identify income and expense accounts. Indeed, MARS has little to recommend it over other packages for just the general ledger function. But, for setting up a comprehensive and relevant reporting system for management, MARS has a lot going for it. True, it doesn't really do

MARS starts with a general ledger accounting package and adds a comprehensive management reporting system that is particularly flexible in identifying groupings. It will run on an IBM System/360, Model 30 and above.

CHARACTERISTICS

SUPPLIER: Management Science America, Inc. (MSA), 1389 Peachtree Street, N.E., Atlanta, George 30309.

BASIC FUNCTION: To collect and organize financial and budget information as accounts. Reporting functions are oriented toward customized grouping of data for various levels of management, as well as conventional financial reports.

OPERATION: The MSA Management Accounting and Reporting System (MARS) consists of four subsystems tied together by a common data base of financial and other information. Each subsystem is composed of one or more modules. There are a total of 63 programs and 34 sort routines in the complete package.

The four subsystems are: General Ledger Accounting, Responsibility Reporting, Cost Accounting, and Management Analysis Reporting.

The General Ledger subsystem is composed of four modules: Basic General Ledger, Alert Reporting, Account Analysis and Reconcilement, and Transaction Explosion. Basically, these modules give the capability for posting accounts, flagging irregular entries or unusual balances, auditing specific accounts, and controlling automatic entry generation and multiple posting of entries.

The Responsibility Reporting subsystem is a pair of modules for generating a budget or plan for assets, liabilities, income, and expenses as well as generating comparative reports for multiple levels of management.

The Cost Accounting subsystem is a single module for allocating indirect costs and generating appropriate cost allocation and profitability reports.

The Management Analysis Reporting subsystem is also a single module and is intended for preparing reports showing ratios determined by the user.

In the full system, a total of 10 different files are maintained. Three are concerned with storing general ledger account information, account history, and transaction history. An additional file contains history information for account groupings specified in various reports. Two files store report specifications and alphanumeric descriptions (labels) for report lines. Another file stores cost allocation factors, The final two files store the budget and profit center data.

Three of the files are organized as indexed sequential (used with report generation and cost allocation functions); the others are all sequentally organized.

▷ anything a good report generator could not do for you, provided your master files retained entries in sufficient detail. However, the logical relationships among accounts for determining what goes into which report would be difficult, time-consuming, and very expensive to set up using a general-purpose report writer.

MARS is a slightly cut-down version of the Financial Information and Control System for Banks (abbreviated as FICS). FICS was jointly developed by MSA and nine banks. MSA estimates that the company itself invested about 37 man-years of effort and 1.3 million dollars in this effort, not to mention the contributions of its co-developers. While MARS stands as a complete package now, it is likely that additional features will be added in time to come. Probably the first expansion area will be to enhance the capability for interfacing with information generated by other applications packages. (MARS can now accommodate the output of other packages, but full access to such information is not available for management reporting.) □

➤ The concept of MARS is to collect data from a single input point for the purpose of maintaining general ledgers and to serve as a data base for management. The power of the account/center numbering technique and the report generation capabilities is to enable the generation of highly pertinent and readable reports that contain company, department, cost center, product, or other account grouping information required by different managers.

ACCOUNT NUMBERS: The disposition of each item entered into the system is controlled by a series of numeric codes.

Companies and subsidiaries are identified by a pair of two-digit codes. This permits data for multiple companies to be independently maintained within the files. Consolidated reporting for multiple sub-companies is also possible.

The account number proper is actually a pair of seven-digit numbers. This first corresponds to the numbers used in a company's chart of accounts. Different charts of accounts can be used by subsidiaries. The second seven-digit number identifies a reporting center, which can be a department, cost center, product, or other grouping.

Two one-digit codes identify classes (expense, income, capital account, assets, etc.) and groups (fixed, variable, etc.). This arrangement facilitates the generation of standard company balance sheets and income/expense reports and certain other company reports.

In addition, items coming in from other application packages, such as accounts receivable, accounts payable, and payroll, are identified with a two-digit code for the application source and an eight-digit transaction/invoice number.

One item can be associated with as many reporting centers as desired, either by assignment when entered or by the relationships among accounts stored in the system.

GENERAL LEDGER SUBSYSTEM: This subsystem forms the basis of the MARS package and is a complete implementation of the general ledger function.

The Basic General Ledger module provides account balances by company, account, and department. Historical balances are stored for up to 3 years (36 periods), and the oldest year's history is cleared when the first period of the current year is entered. Complete transaction data is stored by month; this data can be held for as long as desired. In addition, an active file of transaction data is maintained by day for 10 days, permitting a period to be closed up to 10 days after the calandar date passes.

The reports produced by this module include the normal complement of validation and proof listings plus a host of others. A trial balance, general ledger, and transaction journal by department are produced during each posting (input) cycle. Monthly trial balances, expense recaps, and journals are also produced. In addition, separate balance sheets and income and expense statements can be prepared on a monthly, daily, or on-request base; several different comparative formats are available for these reports.

The Alert Reporting module allows high and low limits to be set for transaction types and account balances. Reporting options range from flagging irregular entries and balances to separate listings.

Individual accounts can be examined through the Account Analysis and Reconcilement module. This is particularly useful for examing such things as employee travel expense accounts to identify unreconciled balances.

The Transaction Explosion module plays an important role in the overall MARS picture. It serves as the interface for linking other subsystems to the information input into the General Ledger subsystem. It also allows the generation of recurring entries such as rentals or other periodic payments. An important function of this module is the explosion of a single entry into multiple general ledger transactions through relationships specified in the entry or stored in the system.

RESPONSIBILITY REPORTING SUBSYSTEM: This is a key subsystem in MARS; it is what essentially makes MARS different from other general ledger packages. There are two key ideas. One is provision for planning a budget, and the other is the concept of responsibility reporting.

The Budget Generation module accepts budgeted amounts for revenues, expenses, assets, and liabilities. Amounts may be for a month, quarter, half-year, or a year; the amounts are automatically scaled to monthly periods. The reports produced by this module are overall summaries of assets, liabilities, expenses, and revenues for a company and couple actual performance with the planned amounts for the remainder of the year to produce projections.

Strong emphasis has been placed on easing the job of preparing a budget by making maximum use of a previous budget. The whole budget or portions can be copied from the previous budget. Other shortcuts that can be used include applying a uniform percentage increase or decrease to all items and tying accounts together to minimize the number of entries required. For example, fringe benefits can be made a percentage of salaries; then only salaries need be entered, which in turn can be the same or a certain percentage higher than last year's.

The Responsibility Reporting module is a specialized report generator. Columnar entries are restricted to one of

Management Accounting and Reporting System
Management Science America, Inc.

➤ 11 formats that present combinations of current period, last year's current period, year-to-date, and last year's year-to-date balance information for actual amounts, budgeted amounts, and variances (percent or dollars). Body entries are specified entirely by the user.

Each detail line can be composed of a single account balance or a group of accounts summed together. The dual account/reporting center numbering technique facilitates identifying exactly the entries desired. Formal organization boundaries are no hinderance in identifying entries because a reporting center can be a department, a location, an arbitrary cost center, or even a product. Inclusion of one item automatically includes all subsidiary items; for example, all sales locations would be included if the sales division were itemized, provided of course, that these were relationships previously set up.

Full provisions are included for associating a descriptive name for each line. Several options are provided for identifying exceptions by percentage or dollar amount variances between actual and budgeted; exception items can be flagged or reports including only exceptions can be produced. Up to eight hierarchies of totals can be incorporated into a report, with an unlimited number of subtotals within each hierarchy.

The report structure is stored on a separate file. Reports can be generated automatically at a specified frequency or generated on an on-requiest basis.

There are two important considerations in evaluating this capability. One is that any grouping of information that can be identified can be included in a report. The second is that similar reports for similar reporting centers are very easy to generate: just take the first one, change any account/center numbers that differ, and you have the report for the second manager. If handled right, standardized reporting is easy to accomplish. On the other hand, highly individualistic managers can also be accommodated without too much difficulty.

COST ACCOUNTING SUBSYSTEM: This module permits the spreading (allocation) of all revenues and expenses (costs) among profit centers and products. The Responsibility Reporting subsystem is intended for reporting to managers responsible for managing a particular phase of a company's business; many of these centers (such as personnel, library, research center, etc.) are not involved directly in profit making (as are manufacturing, sales, etc.). MARS distinguishes between and separates these two types of reporting. Separate files are maintained for profit center and responsibility accounting.

Individual methods can be applied for each allocated item, including standard costing and weighted averaging. Allocations can be based on simple rules such as head counts or square footage or on more sophisticated methods such as standard and unit costs.

Budget information is drawn from the plan produced in the Responsibility Reporting module, using the same basis as for the allocations.

Two methods are available to close out accounts. One is the conventional sequential approach, where amounts are allocated to accounts in order. Depending on the allocation methods, this may result in "round-off" errors or in errors due to the changing base of allocation as successive accounts are handled. To deal with this situation, the second close-out method can be used, which uses simultaneous linear equations to determine all allocations during the same calculation.

Several control reports are produced, including two listings by responsibility center that show where costs were allocated to and where they were allocated from.

The reports for profit centers and products produced from this module use the report generation facilities of the Responsibility Reporting module and are called profitability reports.

MANAGEMENT ANALYSIS REPORTING SUBSYSTEM: The final reporting module includes basic arithmetic capabilities for generating ratios between specified account groupings. Items such as cost of telephone service per employee, working capital ratio, building maintenance per square foot, etc. can be generated in summary fashion. (The details going into the make-up of these ratios cannot be shown.) The output is a comparative report showing current period and year-to-date data for this year and last year for actual, budgeted and variance amounts.

HARDWARE/SOFTWARE REQUIREMENTS: Except for one FORTRAN program in the Cost Accounting Subsystem, all programs are written in COBOL (IBM D Level). A total of 52K bytes is required exclusive of any operating system requirements. Thus, the system will run on an IBM System/360 Model 30 or larger configuration. (MSA states that at least two systems are now operating on Model 30's.) Peripheral requirements depend on file sizes. A typical configuration includes one 2311 Disk Drive and an additional four tape or disk drives. The system is operational under either OS or DOS.

To accommodate the use of simultaneous equations to close out accounts in the Cost Accounting subsystem, the floating-point arithmetic feature is required in the System/360 processor.

PRICING: The price includes 15 days of installation support, a one-year warranty including system updates, source code, and compilation of all programs on the users computer system.

Each subsystem is priced separately, and the basic prices are:

Basic General Ledger subsystem–	$20,000
Responsibility Reporting subsystem:	
Budget Generation module–	$10,000
Responsibility Reporting module–	$10,000
Cost Accounting subystem–	$15,000
Management Analysis Reporting Subsystem–	$5,000
Total–	$60,000

A user can be even more selective as to which portions of MARS he wishes to adopt, with appropriate price modifications. The Cost Accounting subsystem, however, cannot be added without the Responsibility Reporting module.

FIRST DELIVERY: MARS first became operational in a commercial account in August 1970.

CURRENT USERS: About 15 as of February 1971, including the 9 participating companies that assisted in the development. ■

Payroll Software System (64K) Management Science America, Inc.

MANAGEMENT SUMMARY

Payroll is one application that everybody uses. And there is a natural affinity between banking services and the issuance of paychecks. It is no surprise, then, that most of the successful commercial payroll packages are oriented toward bank service organizations. MSA's Payroll Software System is even more oriented toward bank services than most, with features for handling employee banking needs (such as checking and/or savings account deposits, payment of mortgage loans, and payment of personal loans) and facilities for producing invoices for payroll services.

Deductions and other earnings are handled together. Any combination of up ot 30 categories can be defined for each company represented in the master file, and up to 25 categories can be accumulated for each employee. Calculation of deductions and other earnings is quite flexible.

Taxes are computed using the Management Information Service ALLTAX program as an integral part of MSA's Payroll System, ensuring up-to-date tax rates.

The various outputs of the Payroll System include a host of listings and registers oriented toward providing audit trails and maintaining controls, plus other reports and files for the various extra features such as the employee bank services, check reconciliation, labor distribution, payroll service billing, and interfaces with other MSA accounting systems.

Of significant utility is the Special Report Generator, which provides access to the employee master file for individualized reports. The employee data can be expanded to include just about anything. In addition, a close interface exists between the Payroll System and MSA's Personnel Information System. The Report Generator has access to the files of both systems.

The Payroll System includes provisions for hourly and salaried employees and pay frequencies of weekly, biweekly, semimonthly, and monthly.

Pays based on piecework could probably be arranged, but the system is not oriented toward this type of payroll. For example, there is no provision for a minimum pay, even though fields exist in the master file for guaranteed hours and rates.

MSA has been highly successful marketing this package. The customers have been spread through many types of businesses, with about half of the 64 systems currently installed being manufacturers of one kind or another. Other customers include banks, life insurance companies, airlines, and motel chains. □

Capable of running on an IBM System/360, Model 30 or larger, the MSA Payroll System (64K) provides several features oriented toward bank service organizations. Also included in the system is a report generator that, coupled with the employee master file, provides an effective tool for extracting management reports.

CHARACTERISTICS

SUPPLIER: Management Science America, Inc. 1389 Peachtree Street, N.E., Atlanta, Georgia 30309.

BASIC FUNCTION: To generate salaried and hourly employee paychecks for multiple companies. Processing options include labor distribution reporting, Employee Master File report generator, check reconciliation, budget comparisons, payroll service billing, and various bank services such as deposits to a checking or savings account.

OPERATION: Once the master files have been established, all transaction data (timecards, file corrections, vacation pay, and totals) is input at the same time with no ordering required. The transaction data is sorted and edited; several audit reports are generated to identify errors and to provide an entry proof.

The master file is updated with any changes, pay is calculated, and totals are accumulated. The output is two files: checks and other reports. The checks are printed, and the two files are recombined to generate all other reports.

In addition to paychecks, the employee information report generator, labor distribution reports, and a conventional complement of reports and registers, a number of other capabilities are included in the MSA Payroll System.

Comparative reports showing percentage variances between actual labor costs and budgeted (planned) costs can be prepared that include month-, quarter-, and year-to-date totals. This report involves input of the budget data through cards; budget data is not stored.

Another feature is a check reconciliation module. Reports showing discrepancies between cashed amount and amount in the file and outstanding checks are produced. The outstanding checks register includes only those outstanding checks that were issued before a date entered by the user—a very helpful feature if handled properly.

For service orgainzations using this payroll system, a particularly useful service billing feature is included. Basic data about the number of changes processed, the number of records on file, the number of checks produced, etc., is output on cards. A billing program is included which uses these cards as input and produces invoices as output. Charge rates can be different for each customer, and fixed charges can be added in the billing program run.

Four permanent files are maintained by the system: Payroll Master File, Cost and Labor Distribution Master File, Earnings History Master File, and a Description File that contains names and other data required for bank services, W-2, and 941-A reports. The total package consists of 43 COBOL programs and 9 sort routines; the source language is IBM E-level COBOL. MSA plans conversion to ANS COBOL during 1971; previous users will continue to receive support.

Payroll Software System (64K)
Management Science America, Inc.

The Employee Master File can be organized in up to five levels. The first two levels are reserved for companies and subsidiaries, but the other three can be any organization desired (plant, department, section, etc.). These five levels are used, along with other data elements, to control sequencing of checks and various reports.

Both hourly and salaried employees can be paid automatically without submission of timecards. Only one standard labor distribution can be specified for automatic hourly pays, but up to three numbers with specified splits can be used for automatic salaried pays. Any number of labor distributions can be entered via timecards. Non-standard hourly rates and salaries can also be specified in the timecards.

Overtime is computed as a mutliplier (one-half to triple time in half-time increments) times the standard rate or rate entered in the card. Salaried workers can be paid overtime on an hourly basis. Up to four shift differentials can be specified for a company as either a percent or amount per hour.

Vacation pay is calculated separately based on year-to-date average of rate/salary, current rate/salary, fixed amount, or percent of rate/salary. Several options allow adding a percent, amount per hour, or fixed amount to the base. Individual options are available pertaining to taxes and deductions.

Deductions and other earnings are handled together. Each employee can have any combination of up to 25 deductions and other earnings (one must be sick pay) applied to his pay and also accumulated. Each company can set up its own group of 30 deduction and other pay categories. Several options are available for computing each deduction and other earnings. Deduction arrears can be handled automatically with several options for processing in the next pay cycle. Each type of other earnings can be taxed differently.

Taxes are computed through the Management Information Service ALLTAX program, an intergral part of MSA's Payroll System. This program (see Report 70E-594-01) includes all federal, state, and local taxes for the U.S., with updates as changes are made. Taxes for up to 10 different states or localities can be accumulated for each employee, to accommodate employees who move around a lot.

INPUT: A total of seven different card sets are used to accommodate input for payroll purposes. Four of these formats are for changes to files and the addition of new companies for new employees. Individual forms are used for timecards and vacation payments. The remaining card contains totals used for control.

All types of transactions are input at the same time and in any order. Transaction codes (2 alphabetic characters) are set up so that transactions can be sorted in appropriate order (e.g., employee file changes before timecards). This allows file maintenance and payroll to be processed in the same run, while ensuring that all changes have been made before pay calculations are performed.

Payroll information is transmitted to the computer via a Time Card Entry Form produced during the previous cycle. Sequencing of the employee listing can be varied, and multiple lines can be inserted for any employee expected to have multiple pay types. Other forms are used for file maintenance.

OUTPUT: A normal and very adequate number of listings, registers, tax reports, etc., are produced to provide audit trails and records of the processed payroll. In general, these reports are produced for each Level 2 entity (company or subsidiary). General options for all reports include totals and/or page breaks for Levels 3, 4, or 5, and selection of all, inactive, active, or terminated employees.

The standard deductions/other earnings (D/OE) report lists all 30 categories for all employees. The format is 15 doubled columns. This results in a lot of information on the page, with no room left over for accumulated amounts and arrears. This problem is solved with a specialized report generator for producing a separate report for each D/OE category. Only one card is required to specify such a report, and the options are very useful. Control breaks, report sequence, and accumulated amounts to be printed (current and month-, quarter-, and year-to-date) can be specified.

The Special Report Generator allows access to the Employee Master File for custom reports. All elements of the file have been assigned six-character names. A maximum of 22 data elements can be included in any one report.

The data stored for each employee on the master file can be extended in 208-byte segments. A file definition procedure is required to identify the structure and store it in a manner accesible to the Report Generator.

Selection of employees to be included in the report can be based on the value of up to six data elements compared to high and low limits contained in the report specifications. A total of eight cards are required to completely specify a report. Only one report per Level 2 entity (company or subsidiary) can be processed during a run, but a different report can be prepared for each Level 2 entity in the same run. Some special forms are used in addition to the checks.

A comprehensive program is also included for producing mailing labels. Employee home or internal company address can be used, and sequencing is very flexible.

HARDWARE/SOFTWARE REQUIREMENTS: The MSA Payroll System will run on an IBM System/360, Model 30 or larger, with at least 64K bytes of main memory and operating under DOS. (The system can be modified to run under OS if desired.) A COBOL Level E compiler is required to generate object code. All files are sequential and can be on magnetic tape or disc.

PRICING: The price for the complete package is $22,000 (DOS) or $25,000 (OS). If purchased with the Payroll System, the Personnel Information System goes for half of its regular $6,000 price.

The price includes a 1-year warranty, installation (normally 1 week), and training. The programs are provided in source code, with compilation being done on the user's computer system.

A maintenance agreement effective after the first year is currently being firmed up, with the cost expected to run between $1000 and $1800 per year. The agreement will include a 1-year extension to the warranty plus continuous system and tax updates.

An unusual arrangement exists for updating the Payroll System coding. MSA publishes an update bulletin which contains the coding changes for all fixes. Included with the Payroll System is a program, PAYMAINT, which facilitates updating the COBOL source code from the cards keypunched by users. (This program can also be used to maintain any COBOL program.)

INITIAL DELIVERY: January 1970.

CURRENT USERS: 64 as of February 1971. ■

Anyplace II
Marcus Powell Associates

MANAGEMENT SUMMARY

Anyplace II is one of the newest and least known among the numerous enhancements to IBM's DOS that were available prior to the announcement of DOS/VS by IBM in August, 1972. With that announcement, much of the steam was taken out of marketing efforts for DOS enhancements by independent software vendors.

DOS/VS (Release 28 of IBM's DOS) includes a self-relocation capability that supersedes the need for Anyplace II. Furthermore, the availability of DOS/VS at no additional charge on System/370 Models 135, 145, 155-II, and 158 tends to offer an almost irresistible incentive to current System/370 DOS users (who can make excellent use of Anyplace II at this time) to upgrade to DOS/VS. Coupled with the end of free support for DOS on System/360 or 370 computers by IBM as of March 31, 1973, a truly remarkable front of pressure is being applied to DOS users to give up DOS and/or the System/360 in favor of OS and/or DOS/VS on the System/370. (A full analysis of IBM's upgrade strategies can be found in the System/370 report, 70C-491-04).

On the other hand, for existing DOS installations, Anyplace II can provide a useful additional facility. Self-relocatability permits savings in the following areas:

- Machine time—Anyplace II eliminates the need to catalog the same program in the Core Image Library (CIL) more than once, for use in more than one partition, and similarly eliminates the need to re-catalog the program when the supervisor or partition size changes.
- Programming time—self-relocatable programming is a complex and time-consuming effort if not done automatically.
- Operations—only one set of Job Control Statements per program is required, and job scheduling is greatly simplified since any job can go into any partition (space and peripherals permitting).
- Disk storage—Anyplace II saves Core Image Library (CIL) space since the program need be cataloged only once.

As a concept, relocatability (whether by "self" or by an operating system agency) offers a great potential for increased system resource utilization. Under unaided DOS, programs are non-relocatable in the core image library; i.e., they have been assigned to specific main memory locations. Thus, if one of the three partitions in a DOS multiprogramming environment is free, but not the

▷

This System/360 or 370 DOS enhancement provides a self-relocation capability for both user programs and systems software. As one of the least expensive packages of its type, Anyplace II can be a good bet for the cost-conscious DOS user until DOS/VS or a more powerful operating system is installed.

CHARACTERISTICS

SUPPLIER: Marcus Powell Associates, 2694 Doidge Avenue, Pinole, California 94564. Telephone (415) 758-6080.

BASIC FUNCTION: Anyplace II is used to make IBM System/360 or 370 DOS problem programs in COBOL, PL/1, FORTRAN, RPG, or BAL language self-relocatable. As a Linkage Editor "pre-processor," Anyplace II appends a control section to each program phase and adds a segment to the supervisor to make either monolithic or multiphase programs relocatable.

OPERATION: Anyplace II is used to process Linkage Editor input on SYSLNK and the libraries to create a new Linkage Editor input file. The new SYSLNK file is then link edited to produce self-relocating programs.

At execution time, an Anyplace II segment incorporated as a macro at system generation examines each program for a flag indicating that it is self-relocatable. If the program is self-relocatable, it is loaded into main memory at the beginning of any suitable partition, with ADCON resolution done by the control section. After the program is loaded, the control section and relocation table are automatically stripped off, thus returning the user program to its original size and function.

Few limitations or restrictions are placed upon the usage of Anyplace II. Among the strong points of the package is the use of Blocked Fetch to speed loading. The Anyplace system provides 19 diagnostic messages, and the system itself consists of self-relocating programs.

PERFORMANCE: Anyplace II requires less than 1 second to load a phase—as little as one-fourth the time that some competitive independent self-relocation processors require.

HARDWARE/SOFTWARE REQUIREMENTS: Anyplace II can run in a minimum partition size of 10K bytes on an IBM System/360 or 370 under DOS. It adds a control section to each phase consisting of either 15 + A or 15 + 3A bytes, where A is the number of ADCONS in the phase. Each ADCON requires only one byte if the previous ADCON was no more than 17 bytes away; otherwise three bytes per ADCON are needed. Typical size of the control section is about 150 bytes. A segment is added to the DOS supervisor that typically is also about 150 bytes long.

PRICING: Anyplace II is available for purchase at $1,800, including source decks, program logic documentation, and maintenance of the package for one year. Additional installations are available for a charge of $100 each. A 30-day free trial period is provided.

FIRST INSTALLATION: May 1972.

CURRENT INSTALLATIONS: 9 as of September 1972.■

Anyplace II
Marcus Powell Associates

▷ one for which programs awaiting execution have been cataloged, that partition (and share of the system resources) must remain idle. Relocatability allows a program to run from any location in main memory, provided that enough space and peripheral devices are available. Larger systems with plenty of peripherals naturally have more to gain from relocatability, since there is more room to maneuver from a scheduling point of view, and also because the potential for wasted resources is greater.

In addition to these general self-relocatability considerations, Anyplace II requires no changes to the user programs or the Linkage Editor and supports the COBOL, RPG, PL/1, FORTRAN, and BAL languages. On the other hand, modification must be made to the DOS supervisor in one of two ways. At Sysgen time a separate macro (SGANP, for System Generation Anyplace) can be incorporated in the supervisor; or a separate "initializer" (ANYINIT, for Anyplace Initializer) can be used while the system is running to modify the supervisor dynamically. ANYINIT takes about 10 seconds to execute, and can patch the user program self-relocation capability into DOS at any time after IPL (Initial Program Load).

Users of Anyplace II contacted by Datapro report that the package performs as advertised and acknowledge one novel aspect of Anyplace II. A "retention" date is imbedded in Anyplace II for the duration of the no-charge test period. This date is set up to reflect the period of time for which the user is authorized to test the system. This period is typically 30 to 45 days. When the date (as determined from the IPL) is exceeded, neither Anyplace II nor the self-relocating versions of the user's problem programs created by Anyplace II can be used. (If the user does not advance to full paying status following the test period, and continues to use Anyplace II, considerable inconvenience can result. A series of hashed figures cleverly spread out through the logic flow of Anyplace II makes tampering with this Marcus Powell safeguard rather impractical, according to one chagrined user.)

Until mid-1973, DOS/VS will remain a "paper tiger," and DOS users of either a System/360 or 370 can well profit from a DOS self-relocatability enhancement until then. Subsequent to the free availability of self-relocatability in DOS/VS, the market for Anyplace II will be pretty much confined to a dwindling list of System/360 DOS users—but their ranks currently number some 10,000. Other independent sources of self-relocatability are available—both as standalone packages and as combinations of multiple DOS enhancements—and the DOS user will be well advised to investigate a number of alternative vendors carefully before making a decision regarding the future of his operating system. Of particular note in such an investigation are the projected plans of the vendor. Look for the possibility of price reductions and other usage concessions as a gambit for the suppliers of DOS enhancements to gain new business opportunities in your computer installation. This is a particularly good bet now, when it seems likely that comparatively few additional sales of such packages will be made to the general marketplace in the next year, and still fewer after that. Careful shopping for DOS enhancements now could produce significant short-range payoffs.

Anyplace II, although not very well known, stacks up as one of the best buys in the self-relocatability derby, with a price tag about one-half that of some of its more popular competition. □

SYSTEM 2000
MRI Systems Corporation

MANAGEMENT SUMMARY

SYSTEM 2000 is a generalized data base management system that features fast response and access either through its facile imbedded command language or through interfaces to languages such as COBOL or FORTRAN. SYSTEM 2000 can be used on IBM 360 and 370, UNIVAC 1106 and 1108, or CDC 6000 and Cyber 70 systems.

MRI's SYSTEM 2000 is an exceptionally powerful and flexible information update and retrieval system with a wide range of practical applications. SYSTEM 2000 claims the widest exposure to different machine types of any commercially available generalized data base management system; it has been implemented on IBM System/360 and 370, UNIVAC 1106 and 1108, and CDC 6000 Series and Cyber 70 systems. The speed with which it can be used to apply complex selection criteria against large data bases is one of the fastest currently available, and can approach instantaneous response under certain circumstances. For this fast retrieval, users pay an overhead price in data storage space and update time, due to SYSTEM 2000's use of complex file structures.

Commonly, pointers in he form of indirect address references are used in list organizations to physically locate data items. This technique is called "inversion." When combined with some amount of sequential storage, list structures can provide very effective overall data base organizations. SYSTEM 2000 uses this partial inversion approach, and lets the user determine the amount of inversion that will be used. Generally, the more inversions, the greater the potential for quick retrieval. But excessive inversions slow down file creation and updating processes without necessarily further reducing retrieval times. Thus, thoughtful initial definition of the data base (e.g., inverting only the data base items that are normally keyed upon) can greatly reduce both time and storage overheads in SYSTEM 2000 applications by avoiding unnecessary complexity.

In addition to selection of the degree of inversion used in the data base, the creation (and updating) of a SYSTEM 2000 data base requires an intermediate formatting step plus extra processing time for user data to be put into rigidly formatted strings of input values. Alternatively, data can be loaded from a procedural-language program, thus eliminating the string generation step.

The "expansion ratio" for the size of a SYSTEM 2000 data base as compared to that of the raw data is normally about 1.5:1 for 20% keyed (inverted) elements. This ratio is based largely on the complexity of the data base description (e.g., number of repeating groups, etc.) and the percentage of unique values in the data base.

SYSTEM 2000 is ancestrally related to TDMS, one of the archetypal generalized data management systems described in detail in the 1969 CODASYL Committee report on data management systems (available through ACM). Since MRI acquired the Remote File Management System (RFMS) – a version of TDMS – from the University of Texas in 1967, numerous changes have been made to develop the current SYSTEM 2000. Many of SYSTEM 2000's features are specifically designed to meet information retrieval requirements in today's business and scientific communities. Among the most significant of these are the following: ▷

CHARACTERISTICS

SUPPLIER: MRI Systems Corporation, 12575 Research Boulevard, Austin, Texas 78766. Telephone (512) 258-5171.

BASIC FUNCTION: Full-scale generalized data base management system using partially inverted tree files for efficient on-line data retrieval. (The user specifies the extent to which the files are inverted.) Numerous options add interface capabilities for COBOL and other end-user languages to the data base in addition to the imbedded near-natural-language user commands. The system is well suited for fully integrated data bases as well as simple, more specific applications on subsets of data.

OPERATION: SYSTEM 2000 is a terminal-oriented system for interactive use through local or remote locations. A batch version is also available. The user, after analyzing his present and projected future information requirements, sets up a data base definition which is functionally analogous to the record formats used in file layouts of conventional file management systems. Although a certain amount of flexibility is available to the user for modifying this basic format after his data base is created, this capability is rather limited, and the user should take care to make the definition as comprehensive as possible during the initial data base loading phase. The loading process is then performed, in which input strings of data values are fed into the data base. These strings are formatted in a highly structured manner according to the record or data definition set up earlier. Alternatively, a procedural program can be used to create the data base, and thus avoid the string generation.

An optional Immediate Access feature permits direct updating of data base entries through an on-line keyboard, but this process is restricted to the manual speed of the operator, who remains connected in an on-line mode to the central processor during the entire update activity. An interactive process, in which the next question asked by the user is predicated upon the response to his current question, allows a line of inquiry based upon intuition to be pursued. The flexible inquiry language makes it possible to retrieve specific information with a minimum amount of print-out.

A Report Writer feature is also offered to satisfy needs for standard report production. This feature permits the report format to be easily modified, using a simple command to "proof" the format before producing the report.

The basic data storage technique used in the current SYSTEM 2000 version employs combined sequential and inverted files. Inversion facilitates high-speed retrieval, but it carries liabilities in the time used for updating the resulting complex structure and in increased size of the stored data base. Theoretically, a data base can be smaller than the direct size of the raw data when inverted storage techniques are used, but that practically never happens. Variables influencing the size of the final data base include the lengths of individual data values within each record, the number of different data values in the data base compared ➤

- Using the Multiple Thread feature, SYSTEM 2000 is completely re-entrant, allowing multiple simultaneous users to access a given data base under multiprogramming. During such access, requests are processed on a first-come, first-served basis, with suspension of retrieval during an update.
- SYSTEM 2000 has procedural-language interfaces for COBOL, FORTRAN, and (for IBM systems only) PL/1 to permit user application programs to access the SYSTEM 2000 data base. This is particularly useful for environments where the imbedded retrieval language is desirable for fast responses, while a complex user process that manipulates data can modify the SYSTEM 2000 data base during batch operations and can be performed concurrently with or separately from other interactive use.

USER REACTION

Users of SYSTEM 2000 contacted by DATAPRO 70 were unanimous in their praise for the product and its vendor. The users stated that SYSTEM 2000 performs as advertised and yields data base expansion ratios of 1.5:1 or less when the user keeps within the 20% keyed item ratio recommended by MRI. Datapro interviewed users with IBM System/370 and CDC Cyber 70 Systems. One user had particular praise for two features in the latest release he's received: support for 1000 elements (up from 430) and multiple data base support in procedural languages (which allows the user to work with second, third, and fourth data bases while SYSTEM 2000 automatically keeps track of the user's position in each data base). This user also plans to adopt the multiple-user option of SYSTEM 2000. All users said that response times for data base inquiries were very fast. In general, SYSTEM 2000 offers excellent potential for high-speed on-line information retrieval based on complex requests.

Since SYSTEM 2000 is available through several service-center operations, its high price tag need not be an obstacle to users who wish to try the system. MRI quotes studies made by management consulting firms which indicate in-house development costs of $1.5 to 3 million for systems with the capabilities of SYSTEM 2000. As one of the more successful current examples of the increasingly popular generalized data base management systems, SYSTEM 2000 deserves careful consideration by users with management information system requirements. □

to the total number of values, and the amount of inverting used.

The primary concept in the inverted file structure is the "repeating group", which is similar to a repeated field in a variable-length record. Repeating groups can be nested within other repeating groups for a theoretical SYSTEM 2000 limit of 32 levels. The practical limit, however, is about 8 to 10 levels. All of these factors combine to produce a that data base is typically larger than the total raw data by a 1.5:1 ratio.

Other data is stored on tape or disk in sequential files with an index built on disk. Retrievals are accomplished by selecting data from the inverted files on disk plus one pass of the sequential files to pull off any other data needed.

Other features of SYSTEM 2000 are as follows:

- Audit trial facilities preserve a machine-legible file of the updating transactions and are used with an archive copy of the data base for audit or backup purposes.
- Interfaces for high-level procedural languages enable the user to access the data base through SYSTEM 2000 "get" and "put" macros.
- Efficient support exists for data bases ranging in size from a few thousand to hundreds of millions of characters.
- Teleprocessing interfaces to monitors such as TCAM, Multi-Faster, CICS, and others allow scheduling of a SYSTEM 2000 procedural-language program or use of MRI's IMMEDIATE interactive language.
- A Multiple Thread feature allows for very high-volume processing, e.g., where more than 6 SYSTEM 2000 commands per second are to be processed.
- Security provisions utilize passwords to give read and/or write access to selected users for specific data base components.
- Extensive diagnostics help users through the interactive process.
- Data can be loaded either from a terminal or through batch input streams. Intentionally missing components need not be blocked in with dummy data.
- Standard operating system interfaces are used by SYSTEM 2000.

HARDWARE/SOFTWARE REQUIREMENTS: On IBM computers SYSTEM 2000 requires at least a 360/40 or 370/145 with at least 256K bytes under OS or OS/VS. The Decimal and Floating Point features are required, as well as sufficient on-line disk storage for the data base(s) and scratch files. Direct SYSTEM 2000 main storage residence requirements may vary from 140K to 200K bytes.

SYSTEM 2000 also runs under EXEC 8 on a UNIVAC 1106 or 1108, using approximately 28K words of memory, or under SCOPE or KRONOS on a CDC 6000 Series or Cyber 70 computer, using approximately 18K words of memory.

PRICING: Prices below are for the IBM System/360 and 370 version. Contact MRI for prices and feature availability on other computers and for information regarding on-line access to SYSTEM 2000 through a service center.

SYSTEM 2000 is available through three lease plans, two monthly rental plans, and a license agreement. The license agreement permits an organization to provide data center service, paying computer-time royalties to MRI; it is offered for $25,000, which includes installation and training. Installation and one week's training cost $750 the under lease plans and $2,000 under the rental plans. Contact MRI for details about rental plan rates, additional instruction courses, conversions to paid-up leases, and multiple-installation discounts on paid-up leases.

	Paid-Up Lease	1-Year Lease	6-Month Lease
Basic SYSTEM 2000:	$25,000*	$25,000*	$12,500*
Procedural Language Interfaces–			
First:	10,000*	295	355
Subsequent:	5,000*	150	180
Immediate Access:	20,000*	590	710
Report Writer:	15,000*	445	535
Teleprocessing Monitor:	2,500*	75	90
Multiple Thread:	20,000*	590	710

*One-time charges; all others are monthly payments.
Monthly rental plans are also available.

INITIAL DELIVERY: Version 1 (fully inverted form), June 1970; Version 2 (partially inverted), July 1971.

CURRENT USERS: More than 40. ■

Quick-Draw
National Computer Analysts, Inc.

MANAGEMENT SUMMARY

The Quick-Draw packages, which produce flowcharts of COBOL, BAL, FORTRAN, and PL/1 source programs, are well-established entries in the flowcharter field. The NCA packages place emphasis on charting speed, use fewer columns of wider boxes, and provide very explicit back-references. Versions for many computer models are available, especially for COBOL programs.

Quick-Draw COBOL was delivered only 6 months after the EDP industry's first flowcharter. It was soon followed by other Quick-Draw packages that produce sequential flowcharts and diagnostic listings of programs coded in FORTRAN, PL/1, and System/360 BAL. All are useful in reducing programming and program maintenance time by producing better documentation – even for programs written in the English-like COBOL language.

NCA recommends that Quick-Draw be used at three points during the development of a program, plus once for the fully operational final version. The final run, naturally, is for documentation; the earlier runs assist in desk checking to minimize machine test time. The first run should be made after a clean compile, but before test data is processed. The second run should be just prior to operational release, to help locate bugs which are encountered during the initial installation period. The third run should be made before making major changes to an operational program. The control number of this run is compared to that of the control copy, and can be used to verify that the version to be modified agrees with the current master copy.

COBOL

Evaluation of competitive COBOL flowcharters tends to require a detailed analysis of the differences in their features and performance, together with the effects these differences will have in a specific installation. There are, however, a few major differences which can be decisive enough to preclude the need for a long and detailed analysis.

There are noteworthy differences between Quick-Draw COBOL and its primary competitor, ADR's COBOL/ Autoflow (Report 70E-052-03). Quick-Draw places a greater emphasis upon speedy charting, but it does not have the capability to resequence the programmer's coding. ("Two-dimensional" flowcharts, as referenced in NCA documentation, do *not* have the same meaning as in the literature of other companies.)

Some of these functional differences result from a single difference in the design philosophies of Quick-Draw and its prevalent competitor. When the other package analyzes a COBOL source program, it sorts the whole list of boxes to be produced. This allows portions of coding that are written later in the program to be inserted into the flowchart at the point where they are first referenced. This sorting requires considerable time and rules out any overlapping of processing and printing. Thus, when used, the resequencing capability prevents Quick-Draw's competitor ▷

CHARACTERISTICS

SUPPLIER: Quick-Draw is marketed in the United States and Canada exclusively by Informatics, Inc., 21050 Vanowen St., Canoga Park, California 91303; telephone (213) 887-9121. It is marketed in other countries by various agents.

ORIGINATOR AND MAINTAINER: National Computer Analysts, Inc., Highway One, Princeton, New Jersey 08540; telephone (609) 452-2800.

COBOL

BASIC FUNCTION: To produce an easily read flowchart of a COBOL source program, including items referenced only by COPY and INCLUDE statements, with minimum use of computer time; and to back up the flowchart with a number of listings and diagnostics in order to create suitable program documentation.

INPUT: Consists simply of the COBOL program to be charted, together with the standard job-control, header, and ending cards. The header provides details of any reports that are to be suppressed in the output. Source programs can be on cards, or in disk libraries. COPY and INCLUDE statements in the source programs can optionally be recognized to copy coding from the library as input to Quick-Draw.

OUTPUT: The output consists of three listings and the flowchart itself. Two of the listings are produced by the Quick-Draw program, and the third is simply a copy of the source program plus a "control number." This number is based upon a hash sum of all of the characters in the COBOL program, and Quick-Draw will produce a different control number if any changes are made to the source, including punctuation, etc. A control number is also generated for each statement to provide a means of checking on program security. The two created listings – a Cross-Reference by Term and a Diagnostic Flow Summary – are described below after the description of the flowchart.

CHART FORMAT: The main flowchart is arranged in three columns across the paper, in strict order as the programmer wrote it, with all entries from COPY and INCLUDE statements in their logical positions. Each box has up to four identifiers for reference purposes. A Box Number shows the box's position on the page. A Sequence Number shows the number assigned by Quick-Draw itself to each statement. Where a new paragraph is started, the paragraph name is printed above the box.

The shape of the boxes can be rectangular, diamond, or circular. "Slanted" rectangular boxes, for input/output operations, are printed in an unconventional form to maximize their capacity. Boxes for different types of statements are formed from different characters to enhance readability. In general, asterisks are used to outline boxes which involve decisions (such as ALTER and GO TO state- ►

from seriously competing in a speed race. However, in evaluating the two flowcharters it should be noted that the other package can be run without this resequencing of the input, at a higher-than-normal speed.

There are other ways in which the Quick-Draw and its competitors differ. One is in the widths of the various flowchart boxes. With one package, only 17 characters per line can normally be included in a box. By contrast, Quick-Draw boxes can hold up to 30 characters per line. This reduces the number of available columns to three per page but keeps the boxes shorter vertically. Where COBOL programmers keep their data names in short or coded form to reduce writing, the wider boxes may not be too important. But the increasing use of precompilers, which take the abbreviated data names and expand them into easy-to-read forms before the programs are compiled, may make the wider Quick-Draw boxes quite valuable.

Another feature of Quick-Draw COBOL is its ability to include in a program parts that have not been written out in full by the programmer, but simply referenced by him. This capability allows coding referenced by the COPY and INCLUDE verbs to be brought into the program flowchart.

A difference between Quick-Draw COBOL and other flowcharters that earned the mention of one user was Quick-Draw's explicitness in refer-backs. That is, where does a reference point on a page come from? The user, who once had a different flowcharter, says that the package he rejected only indicated with an asterisk the fact that a location could refer back to two or more places, whereas Quick-Draw provides a listing not only of all possible back references, but of the function of each refer-back instruction.

Management may also be concerned about the flexibility of the system, and this is a Quick-Draw strong point. The user can decide the size of the page to be used and choose a printer spacing of either 6, 8, or 10 lines per inch. Moreover, the Quick-Draw program itself is written in COBOL and can be compiled and run on many non-IBM computer systems. It is currently in use on a number of systems other than the System/360 and 370, demonstrating that the capability is not just a theoretical one.

BAL

Quick-Draw BAL was written as a companion to NCA's Quick-Draw COBOL. About one-third of the Quick-Draw COBOL installations also use Quick-Draw BAL.

In addition to flowcharts, Quick-Draw BAL generates six types of listings – which are really some of the tables that have to be formed within the flowcharter anyway – and prints them out for the programmer's attention. In addi-

ments), and "straight-line" characters – dashes and I's – are used to outline non-decision boxes such as MOVE and ADD statements. The boxes vary in size. Rectangular boxes can be as long as required, and there are four sizes of diamond decision boxes, depending upon the number of characters to be displayed.

Four levels of detail are available in the flowchart: expanded, standard, condensed, and "super condensed." All of the Quick-Draw features are used in the standard and expanded forms. A separate box is drawn for every single statement in expanded flowcharts, whereas all like statements in the direct logic flow are put into one box in the standard form. The condensed method lumps all uninterrupted logic flow statements in a box, even if they are different; while "super condensed" charts do not show the actual statements within the box.

Each box shows all the references to it from other parts of the program, listing the page and box number of each statement that can cause a transfer to that box. Exits from the boxes are handled in different ways according to the types of statements involved. All of them use the page and box number form of reference.

An unconditional branch uses a branch symbol with the page and box number printed inside, and also prints the name of the destination outside the box.

A conditional branch attempts to use lines drawn from the three available corners of the diamond decision box to either the next box in sequence (that is, to the next statement on the main line) or to a reference to the destination, by page and box number and by name. The lines may be omitted and only the references printed in some cases, however. In any case, the path to each reference is labeled "YES", "NO", etc. to identify the corresponding condition.

Go-To-Depending exits are listed inside a rectangular, asterisk-lined box which shows the possible values of the variable and the page and box references and names of each of the associated destinations.

LISTINGS: Before the flowchart is produced, the optional listings are created. These include a source program listing, a Cross-Reference by Term Listing, and a Diagnostic Flow Summary.

The Cross-Reference by Term Listing lists all literals, data names, and all paragraph and section names alphabetically, along with the positions in the flowchart or Data Division where they were defined. Alongside each name are statement numbers (assigned by Quick-Draw) of all procedural statements that reference that data name, and indicators showing, by statement, how the name was referenced. These usage indicators show alterations, redefinitions, input, output, PERFORMs, tests, use as a sending or receiving field, etc. This listing also includes PICTURE and VALUE operands, providing easy reference to definitions for checking purposes.

The final listing is the Diagnostic Flow Summary. NCA emphasizes that this contains only supplementary diagnostics and does not replace the compiler diagnostics. There are also references printed on the flowchart which point to the specific problems noted on the Diagnostic Checklist. The list itself consists of the sequence number of each suspect statement and a reference to the type of error suspected.

Quick-Draw currently checks for 30 types of possible errors. These include the lack of a verb in a paragraph, lack of a second quote after a literal, an apparently missing period, or the occurrence of nested conditionals and PERFORMs, as well as cases such as undefined branches.

PERFORMANCE: Quick-Draw processes between 125 and 150 COBOL source statements per minute in typical installations. One user, with a 370/155 under OS with HASP,

Quick-Draw
National Computer Analysts, Inc.

tion, NCA provides a recommended debugging procedure to aid the programmer in using the Quick-Draw-produced charts and listings effectively.

Thus, the role of automated flowcharting is clearly changing. It is important to note that this change is by addition rather than reduction. The original advantages of computer-generated flowcharts – documentation that is complete, economical, standardized, and easy to keep up to date – are undiminished. And purchase can often be justified on these grounds alone. Quick-Draw, in particular, generates charts which use wide boxes, are easy to read, and are produced at comparatively high speeds.

The additional debugging capabilities of Quick-Draw BAL can therefore be considered a bonus, or a valid reason for reconsidering the overall value of the package. They can often be just as valuable as the ability to generate the flowcharts themselves.

FORTRAN

FORTRAN, unlike COBOL, is not a logically complete language. The implied operations used in the control of looping and the setting of switches were quite advanced when FORTRAN was conceived over 15 years ago – but now they look a bit jury-rigged and place a basically unnecessary strain on FORTRAN programmers at debugging time.

Quick-Draw FORTRAN is aimed as much at reducing this debugging strain as at producing the improved program documentation that any automatic flowcharting system provides. Even so, the documentation aspect should not be discounted. Inadequate documentation is an ever-growing problem in many FORTRAN shops, particularly where a large number of programs are being continually modified and remodified, and a good automatic flowcharter can easily justify its price for this purpose alone.

As part of the support for Quick-Draw FORTRAN, NCA supplies a suggested debugging procedure. NCA recommends using Quick-Draw *after* the source program has been cleanly compiled, at a time when the programmer would normally begin testing the program by running it with test data. Quick-Draw can help at this point by bringing out errors of the type that do not produce illegal programs but can lead to incorrect results at execution time.

The number of program errors that can pass through the compilation process without detection will naturally vary from compiler to compiler and from programmer to programmer. And, of course, not all of these undetected errors will be brought to the programmer's attention through the Quick-Draw flowcharting process.

Because the logic flow of the original FORTRAN program is not altered to fit into a more esthetic flowchart, the

reports that the typical run consumes 8 minutes of elapsed time on the system, exclusive of the spooled printing under HASP. Users of various IBM and Burroughs systems report that times vary according to the size and complexity of the program, but that a Quick-Draw COBOL run typically takes about as long as the COBOL compile.

HARDWARE/SOFTWARE REQUIREMENTS: Quick-Draw COBOL is written in COBOL and can be run on a variety of computers, including the IBM System/360 and 370, Honeywell Series 200, Burroughs B 3500 or B 5500, and UNIVAC 9000 Series, 1100 Series, or Series 70. On an IBM System/360 or 370, Quick-Draw occupies about 24K bytes of memory in a 32K system, or about 40K bytes of memory in a larger system. Quick-Draw runs under OS, DOS, or TOS on IBM processors and creates five work files. IBM disk work files can be 2311's, 2314's, 3330's, or equivalent.

There is no limit to the size of a COBOL program which can be run under Quick-Draw. One IBM 2311 Disk Drive is required for about 10,000 COBOL statements, or an IBM 2314 for about 40,000 statements. Specific hardware requirements for non-IBM equipment can be obtained from NCA.

BAL

BASIC FUNCTION: To produce sequential flowcharts and diagnostic listings of source programs written in IBM System/360 BAL (Basic Assembly Language). Control facilities permit printing of optional output and omission of certain parts of the program.

INPUT: Consists simply of the BAL program to be charted, together with the standard job-control cards and a header card. The header specifies the exact charts, listings, etc., to be produced. Source programs can be on cards or in disk libraries. COPY statements in the source program can be optionally reorganized to copy coding from the library as input to Quick-Draw.

SYSLST can be accepted as an optional input to Quick-Draw. This affords the major advantage that all user macros will be cross-referenced and charted exactly as they have been expanded by the compiler, and Quick-Draw sequence numbers will match those of the assembler listing. Quick-Draw can also optionally print SYSLST.

OUTPUT: The following types of output are generated:

1. A source program listing.
2. A Cross-Reference by Term.
3. A Cross-Reference to "Equates."
4. Special Cross-References.
5. Supplementary listings:
   a) Diagnostic checklist
   b) Program linkage operation codes.
   c) Unrecognized operation codes.
   d) Assembler-directing operation codes.
   e) I/O operation codes.
   f) Perform-type instructions.
   g) Path terminations.
6. The flowchart.

The source program listing contains a control number for each statement and for the entire program. These numbers provide a means for checking program security.

CHART FORMAT: The main flowchart is arranged in three wide columns across a 120-character print span. Each box is labeled with three identifiers for referencing purposes: a box number that shows its position on the page, a 5-digit sequence number generated by Quick-Draw, and a programmer-provided tag. These identifiers are printed above the box.

The flowchart boxes come in a variety of shapes: rectangles for processing, diamonds for decision, slanted rectangles for input/output, "circles" for branches and exits, and

Quick-Draw
National Computer Analysts, Inc.

Quick-Draw flowchart may conveniently be used side-by-side with the FORTRAN listing for desk-checking or diagnostic purposes.

Here, then, is a reasonable way to appraise the value of Quick-Draw FORTRAN in your own installation. Run a few typical newly-written programs through the compilation process until they are "clean" (i.e., no more compiler diagnostics). Then flowchart them with Quick-Draw and see how many remaining errors can be quickly spotted by examining the resulting documentation. In this way you can easily judge Quick-Draw's overall value as a debugging aid and "insurance policy" as well as a documentation tool.

PL/1

The general acceptance of PL/1 has not been as wide as originally hoped for when the extremely complex and powerful procedural language was introduced for the IBM System/360 in August 1966. Even though IBM aggressively endorsed PL/1, the difficulty of implementing an efficient compiler has retarded its availability on non-IBM equipment, and for some years the use of PL/1 pretty well locked a particular installation into IBM as a mainframe supplier.

Quick-Draw PL/1 is used primarily as a debugging tool rather than as a program logic design aid, but it does not produce the comprehensive diagnostics required to independently desk-check PL/1 programs. For this reason, NCA recommends that a PL/1 compilation be run before using Quick-Draw for desk-checking. Because the fundamental Quick-Draw philosophy calls for presentation of the flowchart in the same order as the program logic – rather than rearranging the boxes on the flowchart for esthetic purposes – the Quick-Draw flowchart is readily usable side-by-side with the program compilation and diagnostics.

Quick-Draw does, however, produce several lists which assist in debugging. The most helpful of these is the Statement Label Cross-Reference. This list sets forth all of the procedural statement labels and those labels referenced by procedural statements along with the location of each reference. NCA recommends a method of using this list for debugging in the user's manual. In conjunction with that list, the Cross-Reference to Statement Type list and a summary list of diagnostics found by Quick-Draw provide the means to identify nearly all the errors in a PL/1 program with a minimum of actual test runs on the computer.

Capitalizing upon the numerous similarities between FORTRAN and PL/1, NCA has developed Quick-Draw flowcharters for the two languages that are very similar and offer nearly identical capabilities. PL/1 users who wish to develop after-the-fact documentation and benefit

triangles for halts. Boxes for different statement types are formed from different characters to enhance readability. In general, asterisks are used to outline boxes which involve decisions, and "straight-line" characters – dashes and I's – are used to line non-decision boxes such as Move operations.

The rectangular boxes can hold up to 30 characters per line and can vary in depth. There are four sizes of diamond decision boxes, depending upon the number of characters to be displayed.

The entrance to each box lists references to the page and box number of each instruction that can cause a transfer to that box. The entrance is listed even if the referencing box is on the same page as the destination box. All of the entries to a particular box are shown on the flowchart.

Exits from the boxes are handled in different ways according to the type of instruction involved. All of them use the page and box number form of reference. An unconditional branch, for example, uses a branch symbol with the page and box number printed inside, and also prints the name of the destination outside the box. The contents of the boxes take advantage of the 30-character width, and where possible the number of boxes is reduced by combining two or more instructions into a single box. This is particularly true in processing operations, where a single box could contain an ADD followed by two MOVE instructions provided that no entries or exits occurred between them.

Listings of Data Definitions on the charts can be suppressed. The position of a group of data definitions will then be indicated by the word "DATA."

LISTINGS: The Cross-Reference to "Equates" List consists simply of the EQU, DS, and OH statements noted in the program, together with the entries to which they have been equated and all references. The table is provided in alphabetical order for ease of entry.

The Special Cross-References List actually contains references to many instructions that may not have been modified at all. It lists all those instructions which have been referenced by other instructions (other than branches). This includes compare instructions, etc. The list is given in alphabetical order by tag of the referenced instructions, together with the Quick-Draw sequence numbers of the referencing instructions.

The Cross-Reference by Term List is a listing, in alphabetical order, of self-defining terms, literals, and data and procedure tags. Where relative tags have been used in the program – such as COLI + 00001 – each is listed as a separate tag. Any references, aside from branch references, which are made to these instructions in the course of the program are listed. Moreover, any case where a data tag is not referenced at all is also listed. This list uses the sequence number as the reference point for describing where names occur and where they are referred to. Besides the sequence number of the reference, the referencing information includes the op-code used in the reference and the relative position of the operation (e.g., 0065 2 MVC).

The Diagnostic Checklist completes the reference print-outs and is probably one of the most important ones. It includes references to items which for one reason or another are liable to be in error and are therefore called to the programmer's attention. Examples include cases where a branch occurs to an incremented tag (mainly to allow patching to be handled safely), where non-integral increments have been used, where branches are made to self-defining terms, and possible loops with returns.

Two forms of diagnostic warnings appear on the flowchart itself: unentered coding (where some code appears to have no entry point) and unreferenced tags.

PERFORMANCE: Quick-Draw processes between 125 and 150 BAL source statements per minute in typical installations.

▷ from the diagnostic advantages of Quick-Draw can balance the price of Quick-Draw and the cost of its operation in computer time with the potential savings in test run costs and programmer time to produce operational programs.

Pricing of the Quick-Draw systems is competitive, and a management decision based on price will need to carefully consider the required options and available discounts.

USER REACTION

As most observers of the EDP industry know, the popularity of flowcharting packages has been steadily declining. At one time, few users really understood the meaning of documentation, nor were decent standards a part of each shop's procedure. It was the programmer's heyday – he wore a beard, smoked a pipe, and became irritable when "bothered" by management. But now, programmers are being treated like other employees, and documentation standards are widely enforced. What's more, IBM's ANS COBOL compiler has a cross-reference facility, and numerous good and inexpensive language aids, checkout compilers, etc., are available from many sources; just glance at the DATAPRO 70 Index.

But flowcharters still have their place. And, in the case of NCA and Quick-Draw, it can be said that all the users Datapro interviewed have only praise for the company and the services it provides, and that they report the packages perform as advertised.

Among the most meaningful of these users' reports was one from a bank that uses a lot of BAL programs. The bank's operations are audited by the FIC and the State Banking Commission, and complete program documentation, including flowcharts, is a must. The user estimates that he'd be a year behind in this task, just documenting program changes alone, were it not for Quick-Draw BAL. But with the package, he is able to simply run the program through Quick-Draw BAL at audit time to produce the required documents. □

► HARDWARE/SOFTWARE REQUIREMENTS: Quick-Draw BAL is written in COBOL and can be run on a number of computers, including the IBM System/360 or 370, UNIVAC Series 70, and Philco 102. On an IBM System/360 or 370, Quick-Draw occupies about 24K bytes of memory in a 32K system, or about 40K bytes of memory in a larger system. Quick-Draw runs under IBM's OS, DOS, or TOS and creates five work files. There is no limit to the size of a BAL program which can be run under Quick-Draw, provided enough peripherals are available. One IBM 2311 Disk Drive is required for about 6500 BAL statements, or an IBM 2314 for about 26,000 statements. Specific hardware requirements for non-IBM equipment can be obtained from NCA. IBM disk work files can be 2311's, 2314's, 3330's, or equivalent.

FORTRAN

BASIC FUNCTION: To produce an easily read flowchart of a FORTRAN source program, bringing out some of the implied portions of the language; and to do it at speeds high enough to encourage its use during debugging operations.

INPUT: Consists simply of the FORTRAN program deck itself (normally, but not necessarily, on cards), preceded by a header card that specifies some of the available options (whether > and < signs are available for printing, vertical spacing of 6 or 8 lines per inch, etc.).

Because the three work files set up by Quick-Draw FORTRAN are kept in main memory as tables, there is a size limit on the FORTRAN programs which can be flowcharted. On a system with 32K bytes of memory, this limit is about 1000 statements. On larger systems, the limit goes up to about 5000 to 6000 statements.

OUTPUT: The source program is read in and simultaneously listed as the first part of the output. Then, while the various tables are being formed, a statement cross-reference listing, a data name and labels cross-reference listing, and a diagnostic checklist are prepared for printing, followed by the printing of the flowchart itself.

CHART FORMAT: The flowchart is oriented toward the peculiarities of the FORTRAN language. Specific examples are the treatment of the DO, ASSIGN, GO TO, and SUBROUTINE statements.

The DO statement is broken down into a number of simpler statements, some of which are implied rather than explicitly stated in the source program. The DO itself is printed as a comment rather than in a processing or decision box. Next there follows a rectangular processing box which sets up the initial conditions. After this, the contents of the DO loop are charted, finishing up with a processing box incrementing the DO index and a decision box showing the test for completion of the loop.

ASSIGN statements are represented in the flowchart by a special five-sided, asterisk-lined box, pointed to the right. This is a very distinctive box that can easily be spotted on a flowchart. Where an assigned GO TO appears, Quick-Draw uses an arrangement of boxes somewhat similar to those used for DO loops.

SUBROUTINES are a vital part of most FORTRAN programs. The box used for this purpose is a simple rectangle with double vertical lines. The contents of the SUBROUTINE statement, including any arguments involved, are printed inside the box, together with a reference to the flowchart location of the subroutine, if it exists in the program, or the phrase EXTRN if the subroutine is external. The start of a subroutine, whether primary or secondary, is represented by an "oval" box three lines deep. RETURN statements are similarly treated.

Other statements, and the general format of the chart, are handled in much the same manner as in COBOL or other flowcharts. The standard Quick-Draw 30-character-wide boxes are used, and the extra width (compared with competitive flowcharters) definitely increases readability.

PERFORMANCE: Overall flowcharting speed generally ranges from 160 to 350 source-program cards per minute.

HARDWARE/SOFTWARE REQUIREMENTS: Quick-Draw FORTRAN is written in Assembly Language and runs on an IBM System/360 or 370 or a UNIVAC Series 70 with at least 65K bytes of main memory. (A 32K system is practical if DOS is used; the Quick-Draw program requires 22K bytes and can operate with a 10K Supervisor.) Quick-Draw can be used under IBM's OS, DOS, or TOS or UNIVAC's Series 70 TOS or TDOS. Three work files are normally required, along with card or tape input and printer, tape, or disk output. These work files are resident on disk and in main memory.

PL/1

BASIC FUNCTION: To produce an easily read flowchart of a PL/I source program, and to back up the flowchart with a number of listings for diagnostic purposes. ►

Quick-Draw
National Computer Analysts, Inc.

INPUT: Consists simply of the PL/1 program deck itself (normally, but not necessarily, on cards), preceded by a header card that specifies some of the available options (whether > and < signs are available for printing, vertical spacing of 6 or 8 lines per inch, etc.).

Quick-Draw PL/1 uses three work files that are resident in main memory. Because of this, the size of the PL/1 programs that can be charted is limited to about 1000 PL/1 statements for the 32K-byte Quick-Draw, and to about 5000 to 6000 PL/1 statements on full-scale processors (48K bytes or larger).

OUTPUT: The source program is read and directly listed for reference as part of the output. Then, while the various tables are being formed, a Statement Label Cross-Reference Listing, a Statement Type Cross-Reference Listing, and a Diagnostic Checklist are prepared for printing, followed by the printing of the flowchart itself.

CHART FORMAT: Each flowchart page is headed by the name of the procedure shown, and each new procedure is begun on a new page. The page has a 3-column format of symbol boxes, each with a page and box number. Convenient branch points cause a skip to the next page, complete with a note (CONT. NEXT COLUMN, etc.) at the bottom. Interconnecting lines are used on each page to illustrate logic flow. Off-page connectors are shown by a branch symbol with the appropriate entry point identified by page and box number.

The flowchart symbols all have a standard width, and the process symbol, the procedure call symbol, and the I/O symbol have variable heights to accommodate long texts. Procedure CALLS are handled like FORTRAN subroutine calls, with the name of the procedure and any arguments printed within a rectangular box of variable height and double vertical lines.

Other statements, and the general format of the chart, are handled in much the same manner as in COBOL or other Quick-Draw flowcharts. The standard Quick-Draw 30-character-wide boxes are used, and the extra width (compared with competitive flowcharters) definitely increases readability.

PERFORMANCE: Overall flowcharting speed generally ranges from about 160 to 350 source-program cards per minute.

HARDWARE/SOFTWARE REQUIREMENTS: Quick-Draw PL/1 is written in assembly language for IBM System/360 or 370 or UNIVAC Series 70 computers with at least 65K bytes of main storage. (A 32K system is practical if DOS is used; the Quick-Draw program requires 22K bytes and can operate with a 10K Supervisor.) Quick-Draw can run under IBM's OS, DOS, or TOS or UNIVAC's Series 70 TOS or TDOS. Three work files are normally required, along with card or tape input and printer, tape, or disk output. The work files are resident in main memory.

PRICING AND USAGE (all versions)

PRICING: Quick-Draw systems can be leased for 1-year and 3-year terms. Lease prices include maintenance by NCA and installation. There are multiple language discounts, as described below, and multiple installation discounts down to half-price – when those installations are part of the original contract. Renewals on 3-year leases are discounted to one-sixth of the original contract price.

Single Language	1-Year Lease (annual rate)	3-Year Lease (single payment)
COBOL	$2,700	$6,300
BAL	1,800	4,200
FORTRAN	1,800	4,200
PL/1	2,700	6,300

Multiple Languages*	1-Year Lease (annual rate)	3-Year Lease (single payment)
COBOL	1,100	2,500
BAL	700	1,700
FORTRAN	700	1,700
PL/1	1,100	2,500

* COBOL and/or PL/1 cannot be leased as secondary languages to either BAL or FORTRAN.

Mini-COBOL and Mini-BAL systems are available for half of the above single-language prices.

INITIAL DELIVERY: COBOL – January 1968; BAL and FORTRAN – during 1968; PL/1 – May 1969.

CURRENT USERS: There have been over 500 total customers during the lifetime of the Quick-Draw line, and the packages have been used in at least 15 countries. At this time, NCA states the percentages of total users who use each language: 57% of the installations are COBOL, 28% are BAL, 8% are FORTRAN, and the remaining 7% are PL/1. Since almost every one of these users has COBOL, and the total number of "installations" is given as 480, there are at present about 288 Quick-Draw customers. ■

RSVP
NCI, Inc.

MANAGEMENT SUMMARY

This highly human-engineered information retrieval system allows COBOL file searching and report formatting queries to be created from booklets filled in by persons without an EDP background. RSVP runs on IBM System/360 and 370 computers under DOS, OS, and their virtual storage counterparts.

RSVP (Report Service – Very Prompt) is an information retrieval system that allows non-computer people to review existing computer files and produce reports. From the computer management point of view, the importance of this facility lies in the improvement of relationships between the computer department and the other departments of an organization. Management requests for information need no longer disrupt scheduled data processing activities. At the same time, these requests can be answered faster than would be the case if one-shot programs had to be prepared.

As a result of special programs being written outside the computer department, a saving of programmer time will occur and there will be a reduction in the amount of computer time wasted in debugging hurriedly written programs. But these are savings that would be expected from the use of any packaged information retrieval system.

Most retrieval systems, however, will not improve the relationships between the computer department and the outside world. Non-computer people often resist using the aids which are available to them because they feel that they are too complex and too much like programming.

The RSVP documentation successfully combats such resistance. The key item is an RSVP Report Request booklet, which is filled out for each separate request. The booklet is the equivalent of a programmer's coding sheet—except that in effect it includes all the instructions contained in a programming manual as well. It is a well laid out document with a multiple-choice format that needs only one piece of outside support: a list of the fields in each file, identified by a simple reference number. Producing this list is normally the responsibility of the computer department, and need be done only once per file.

The RSVP booklet is one of the best single pieces of work created to date to help non-programmers gain convenient access to the contents of existing files. It guides them through the process of creating a report – from the title, to the columns used, to the totals, etc. Few people will be unable to use it. RSVP does not provide many alternatives or sophisticated usages; these are left to the ingenuity of the programmers, rather than the non-programmers. It is simply an effective, easy-to-use tool. Custom reports are created as a result of running the computer input punched from the booklet.

Of particular interest are the following three RSVP features: ▷

CHARACTERISTICS

SUPPLIER: NCI, Inc., 6075 Roswell Road, N.E., Atlanta, Georgia 30328. Telephone (404) 252-9474. NCI also does business as National Computing Industries. The firm was founded in 1968, and was once a subsidiary of the Continental Telephone Corporation. During 1972, NCI was owned by Honeywell Information Systems, Inc. In Canada, NCI is represented by Comserve Ltd., 1167 Caledonia Rd., Toronto; Telephone (416) 789-7541. In Western Europe, NCI is represented by Gemini International Products Division, Sir Winston Churchilllaan 336, Rijwijk ZH, the Netherlands; telephone (070) 94 93 25; with additional offices in London, Paris, Dusseldorf, and Geneva. NCI software is also in use in Mexico.

BASIC FUNCTION: RSVP uses input created from an organized Report Request booklet to examine a file and create a specialized report. RSVP has the ability to process multiple booklets or reports with a single pass of the data file(s).

OPERATION: The input for RSVP consists of an 8-paragraph booklet. This is broken down into title, report columns, accumulations desired, report sequence and totals to be printed, report type, records required, and arithmetic needed.

The Report Request is prepared by reference to an item list which details, in English terminology familiar to management, the meaning of the data contained in the fields of each file record. An item number, related to each field description, is entered in the booklet.

The selection of a record from the file is defined by means of a series of tests on a field. Five tests are provided to define a specific field. They can be joined within a page for either AND or OR testing. The tests (equal to, less than, greater than, etc.) are applied against a specific value rather than by comparing the value of one field with another field. (To make such a comparison, the creation of a temporary value is required.) Up to 30 pages may be related by AND/OR functions to provide 150 logical selection tests.

Accumulations are defined in terms of what data is to be accumulated from the fields, and when they are to be printed. RSVP accepts item numbers in minor to major sequence; thus, the most significant records are selected last. Control breaks are provided at seven levels, and each one allows the totals of all ten columns to be printed. A new page can be requested at any control break. A request for the number of records in a file can be printed out, if no totals are desired, by checking a box.

Spacing instructions provide for single, double, or triple spacing. Reports can contain either the detail of each ►

RSVP
NCI, Inc.

- ▷ Data can be extracted from multiple files for a single report.
- User-written subroutines can be directly included in RSVP.
- Multiple RSVP reports can be produced with a single pass of the data file(s).

RSVP produces a COBOL source program that includes the specific file description(s) used in a particular run. This COBOL report program must then be compiled for execution. An important feature of RSVP, however, is that these generation and compilation runs are made only once for the same file(s), and are not required for every new request. This means that subsequent report costs are limited to keypunching an average of 10 cards, plus the computer time to scan the file, sort the selected data, and print the report.

RSVP is one of the older information retrieval packages available, and very likely the simplest to use. Although no data base management facilities are provided (the user works with existing files and data bases that can be described in COBOL), this is all the capability many installations need or desire. Many of the retrieval aspects of RSVP have long since been improved upon by competitive systems, but only at a correspondingly higher system cost. RSVP is one of the best information retrieval systems available in its price range.

USER REACTION

RSVP users are very enthusiastic about the package. One DOS user told Datapro that the package paid for itself in 3 months, and that he has seen RSVP produce 9 reports in 10 minutes with one pass of the file. An OS user reports that RSVP cleared up a backlog of about 80 requests for information from personnel/payroll files within its first 30 days of use, and that this backlog represented about 1 man-year of programming efforts.

Users are also unanimous in their opinion that the vendor is competent and responsive, that the package is easy to install thanks to top-notch help in both installation and training, and that user-department personnel can use RSVP with no trouble at all. One user said that it's so easy to use RSVP that security could become a problem. No bugs were reported by any user interviewed, and all those questioned said that the performance is fast enough to please them. □

➤ selected record or simply the counts and totals without the detail. A statistical summary is provided with each report, showing how many records were searched and the number of records that failed each selection test.

At various stages in either the selection or the calculation areas, additional information may be needed. Standard arithmetic operations are available, and up to 10 final results can be obtained, allowing the reporting of analyzed data which may not be in the original file. Computed values may also be used as selection criteria.

For example, assume it is desired to have a report listing employees with salaries that are under $10,000.00 or over $20,000.00 if a 5% increase is applied. The requestor may define item number 85 = 10* 1.05, where item number 10 is employee salary on the user's item list.

Report titles are limited to one line with up to 75 characters. The title and a page number are automatically picked up for each page in the report. Column headings are taken from the field names stored in the RSVP program for each file.

File records for RSVP can be fixed or variable in length, but must be capable of being described in a COBOL Data Division. Each field is defined by an item number, report column heading name, data form, security classification, picture, and an optional edit word. The security classification is used to control who is able to access certain fields in the file. The specification of these factors for each file pair is done once by the data processing department. This data becomes input to the RSVP Generator, which produces COBOL source programs tailored to the files described. Input to the Generator can also include special user COBOL routines inserted at any of 27 entry points provided for this purpose.

After COBOL compilation, the tailored RSVP program is cataloged and ready for inquiry processing. At this point, all inquiries can be processed in "load and go" mode. The cards punched from an RSVP Report Request dynamically modify the cataloged programs for each file at run time to produce the particular report requested.

PERFORMANCE: Request processing times vary widely, depending on computer configuration and size, type and speed of I/O, and size of the file to be searched. Many users report that processing is often at input/output speeds.

HARDWARE/SOFTWARE REQUIREMENTS: RSVP requires a minimum of 22K bytes of main memory on an IBM System/360 or 370 computer under DOS. Under OS/MFT or OS/MVT, a sufficiently large partition or region must be available to contain RSVP plus the COBOL compiler. Under DOS/VS or OS/VS, the foregoing are address space requirements. All IBM disks are supported.

PRICING: RSVP is available under permanent license for prices ranging from $3,950 to $4,950, with installment purchases available. Under a 1-year lease, RSVP costs $2,500. These prices include maintenance for the first year, installation by a systems engineer, two days of on-site customer training, source programs, supporting documentation, and an initial supply of materials and Report Request booklets. Maintenance for subsequent years is optional at an annual charge of $360. Multiply-copy discounts are available.

INITIAL DELIVERY: Spring 1969.

CURRENT USERS: About 80 as of November 1973. Of these, there are 8 in Western Europe, 5 in Canada, and 2 in Mexico. ■

Work Ten
NCI, Inc.

MANAGEMENT SUMMARY

Work Ten is a tool to produce well-documented standard COBOL source code for IBM System/360 or 370 computers. This programming system is presented more as a replacement for COBOL than as an ancillary to its use. Its developers claim that "the scope of Work Ten includes all present-day COBOL programming, plus the simplification of advanced systems now in the planning stage."

Historically, one of the most significant aspects of the usage of COBOL is that as a relatively high-level, self-documenting language, it is easy to learn and use. Soon after the release and early widespread acceptance of COBOL it became obvious that another major advantage accrued with COBOL usage: compatibility. Every major mainframe vendor quickly provided COBOL capabilities for his computer systems. This helped to ease the critical shortage of skilled business programmers—not only because COBOL was an easy-to-learn language, but also because COBOL programmers could readily transfer their largely machine-independent COBOL experience to the systems of other vendors.

On the other hand, one factor that retarded the realization of maximum benefits with COBOL was that the simplicity of the language did not promote a strong discipline, either for development of industry standards for documentation and programming techniques, or for rigid adherence by individuals to such standards as were set up. Thus, the self-documenting capability of COBOL was neither fully nor consistently employed. Furthermore, the mainframe vendors tended to implement COBOL at a number of different levels or subsets of the CODASYL standard, each varying from the levels or subsets offered by other vendors. As a competitive tactic, most COBOL compilers have been enhanced or "extended" with certain unique features in an attempt to lock the user into a particular vendor's system. All of these factors have presented significant user problems at the data processing management level.

The major advantage claimed by NCI for Work Ten is the amount of programmer time saved. The basis for this claim is that Work Ten provides a generalized framework within which the record contents, the report item formats, and the basic business data processing functions of transaction matching and file processing have been built into the Work Ten compiler. As a result, the Work Ten programmer has only to supply the details of the specific work he wants done. Even this area is broken out into a number of different phases in each program, so that the programmer is stimulated to handle each processing step effectively by being asked, in essence, "What do you want to do when such-and-such a thing happens?" ▷

Designed to replace current COBOL programming techniques, Work Ten creates System/360 or 370 COBOL programs and program documentation from standardized input forms filled in by programmers. It relies heavily on the use of a library describing the computer records used by an installation.

CHARACTERISTICS

SUPPLIER: NCI, Inc., 6075 Roswell Road, N.E., Atlanta, Georgia 30328. Telephone (404) 252-9474. See Report 70E-659-01 for additional supplier data and locations of NCI's representatives in Canada and Western Europe.

BASIC FUNCTION: Creation of special COBOL programs, which can be compiled and executed. At the same time, program documentation is produced independently of the documentation provided by the COBOL compiler.

INPUT: Input into Work Ten is written on one or more of three types of forms, using an efficient combination of fixed and free-form entries. One form describes the files and records, one the working storage, and another the procedures.

Individual data fields are referenced by an alphanumeric key. The key has three functions: the first character tells what kind of file the record is in for a given program; the second character is an arbitrary identifier for the current program; and the three-digit numeric part refers to entries in the Work Ten library. Each record is described in the library by its name used for documentation (a maximum of 20 characters, with embedded blanks and special characters permitted), the COBOL level number and USAGE mode, the COBOL PICTURE, and a single standard edit word for output descriptions. Redefinition and OCCURS characteristics are also included in the description.

A working storage definition provides for the temporary field entries needed in programs. All of the standard COBOL storage characteristics are allowed, and are entered for the particular program rather than being taken from the library. PICTURE, VALUE, USAGE, OCCURS, edit words, and COMMENTS can all be used to describe working storage. A single entry in this section can be used to define accumulators for multiple levels of control.

Procedures are broken down into basic and separate phases. These include: (1) the program initialization phase, (2) the treatment of incoming transaction records, (3) the treatment of incoming master records, (4) the treatment of a transaction record after it has been either matched or failed to match to the appropriate master record, (5) the final treatment of updated master records, (6) the actions desired at control breaks, (7) the actions desired when grand totals are taken, (8) the work involved in heading records and beginning new sheets, (9) the work involved at the bottom of a printed sheet, (10) the final end-of-job operations, and (11) procedures executed as closed routines; these can be used throughout a program.

All normal processing is expected to occur at one or another of these times, and the Work Ten user is asked to define the required work separately for each phase. ►

Work Ten
NCI, Inc.

▷ Work Ten offers substantial savings in the normally very expensive area of program maintenance due to its ability to automatically document every program. Missing or obsolete documentation is the main reason for high maintenance costs. With Work Ten, a very readable narrative of the program logic, plus all technical details relating to the files and records used, is produced with every program change, thereby keeping the documentation current and complete.

Work Ten is a major system for use in conjunction with a COBOL programming shop. As such, it competes not only with alternatives to COBOL, such as Informatics' MARK IV (Report 70E-500-01), but also with other COBOL supplements or preprocessors such as ADR's MetaCOBOL (Report 70E-052-07) and PMI's SCORE (Report 70E-694-02).

USER REACTION

Work Ten users, ranging from customers with large multiple-370 installations to those with DOS 360's, all have only praise for the package. They say it saves programmer time (some say up to 50%) while it standardizes programming and provides documentation. Moreover, it is easy to use: one programming manager recognized that after he brought the manuals home overnight and wrote two working programs in an hour the next morning.

At one particularly large installation, nearly 90 programmers are permitted to program only in Work Ten. This use of Work Ten as a company standard by a large, sophisticated System/360 and 370 user constitutes a strong endorsement of the package. It should also lay to rest any arguments that Work Ten's use imposes too many restrictions on programmers. □

➤ **Segmenting the logic of a program in this manner should speed the programming effort. The developers of Work Ten provide a handy worksheet for use by analysts, which allows the specification of program requirements in the same segmented form.**

These procedures can be regarded as being "payload" procedures. That is, the actions which are listed under these 11 sets of situations are obvious, and management can appreciate them without having any particular EDP expertise.

A different group of procedures is oriented to the needs of the data processing specialist. They are also broken down into groups of procedures that are to take place at particular times during a program. The "D" (Declaratives) code, for instance, lists a group of procedures which are to be used each time a non-standard label is met while processing a specifically defined file. It allows the user to move the label to working storage or to process it in place. Other such supplementary codes are provided to allow the Work Ten program to interface with file structures which are not supported in COBOL.

The complete range of arithmetic data manipulation and testing is available as necessary. The addresses used for GO TO's and subroutine execution are the line numbers of the various procedure forms. In addition, procedure forms allow one to specify the printing formats. This is done by a "Stream I/O" shorthand method, providing the identity of fields, strings of Hollerith characters, blanks, etc., to be printed. No forms are needed to describe the desired output. The output format results from the definitions of the printing required, the description of the editing, and the field sizes in the library or working storage, and the output fields are formatted accordingly.

PROCESSING: The Work Ten compiler consists of two main phases separated by a sort. The first phase reads the Work Ten source statements, extracts the necessary records from the library, and writes documentation records, diagnostic records, and COBOL source language to two work files. These two files are then merged together and sorted. The second phase of the compiler reads these records, producing the documentation, the source COBOL, and appropriate job control statements if there were no fatal diagnostics.

OUTPUT: The program documentation produced by Work Ten is very complete and is divided into five sections: (1) the System and Program Notes (Work Ten source code); (2) the Files and Records Used, including a summary of control keys, the logical and physical description of each file, and the detail of each record type within each file, showing where each field was used, altered or printed in the program; (3) a similar expansion of User Storage Definitions; (4) a sample report spacing layout, in X's, Z's and 9's format, to allow verification prior to running test data; and (5) a step-by-step narrative of the program, which clearly describes the logic of the program in segmented form.

PERFORMANCE: The machine time required to generate a program with Work Ten can be considered in two parts: the Work Ten program itself and the COBOL compilation. The COBOL compilation takes the same length of time as any normal COBOL compilation. The Work Ten compilation time is estimated to be about 50% less than the COBOL compilation time in most cases.

HARDWARE/SOFTWARE REQUIREMENTS: The minimum system required for Work Ten is a 65K IBM System/360 or 370 under DOS, OS, or their virtual storage counterparts, with 2 disk drives (any IBM type) or 3 tape drives.

PRICING: Work Ten is available only under permanent license. Prices for the DOS or DOS/VS version range from $9,950 to $12,950, and prices for the OS or OS/VS version range from $11,950 to $15,950. These prices include first-year maintenance, training for a limited number of personnel, on-site installation of a tailored system with supporting JCL by a systems engineer, and an initial supply of manuals, programmer's guides, reference cards, and coding forms. Multiple-copy discounts are available. Maintenance for subsequent years is optional at an annual cost of $1,200.

INITIAL DELIVERY: December 1969.

CURRENT USERS: 102 as of November 1973. Of these, 10 are in Western Europe, 5 in Canada, and 2 in Mexico. ■

SAMSON
On-Line Software, Inc.

MANAGEMENT SUMMARY

On-Line Software, Inc. is one of the independent firms that has developed considerable expertise in the support and enhancement of IBM's CICS, as mentioned in Report 70E-491-02. The company's SAMSON package is a CICS system activity monitor.

In light of IBM's November 1973 announcement (at the GUIDE meeting in Boston) of the CICS On-Line Test/Debug installed user program, the value of SAMSON should be carefully considered in terms of what facilities the user feels are necessary for his CICS installation, with about $500 at stake in the decision.

SAMSON is a monitor. That means it lets the user look at the CICS program in execution, but not to stop it and make a change, step through it an instruction at a time, or stop at a designated address. These things can be done with the IBM program, though. But SAMSON costs only $500 (a 1-time charge), while the IBM CICS On-Line Test/Debug program costs exactly twice as much in monthly rental payments over a year's time. (IBM typically waives rental payments on installed user programs after one year of use.) Also, SAMSON is systems performance and behavior-oriented, rather than being designed for single-program checkout.

SAMSON lets the user obtain a picture of what is going on in the system from any terminal. What's more, it can be caused to do so in an automatic repetitive mode, at specified intervals and for a specified number of times. A display of all active and suspended tasks in the system, plus summary information, is presented by SAMSON.

If you tend to discourage on-line testing and troubleshooting, and instead encourage program developers to work their ideas out very carefully before using machine time, SAMSON can also provide the minimal program testing tools that you may need. That is, the status information it can provide may be sufficient to enable a careful developer to work latent bugs out of a program. SAMSON can also be used to keep tabs on a CICS system and determine how increasing work will affect it in the future. But if you wish to develop and test a CICS application on-line, SAMSON probably doesn't have the power you need, and you should look at the IBM program.

In use as a monitor, SAMSON requires minimal main storage (about 1K bytes). Moreover, since it is memory-resident, it has no noticeable overhead effect on the processing times of applications in the system.

USER REACTION

SAMSON has eight users at this writing. The users interviewed by Datapro all stated that SAMSON functions ▷

SAMSON is a system activity monitor for on-line use in System/360 and 370 installations that use IBM's CICS. Its purchase price amounts to only half of a year's rental for the IBM On-Line Test/Debug program for CICS, but there are some noteworthy differences in the facilities and orientation of the two programs.

CHARACTERISTICS

SUPPLIER: On-Line Software, Inc., 411 Hackensack Ave., Hackensack, New Jersey 07601. Telephone (201) 489-0400.

BASIC FUNCTION: On-line monitoring of status in an IBM CICS environment, either singly or repetitively.

OPERATION: SAMSON is received as a complete object deck, ready to be link-edited into the CICS relocatable program library. It can be rapidly installed in any DOS/CICS-Standard (Version 1.0 or 1.1) or OS/CICS (Version 2.0 or 2.2) configuration. The user simply link-edits the supplied object deck into the CICS relocatable program library, with the module name SAMSON; creates an entry in the CICS/PCT with a transaction identification SAMS, a program name SAMSON, a transaction work area of 50 bytes, and a priority of 254; and creates an entry in the CICS/PPT for the program SAMSON.

Then, using any terminal as input, the user enters his SAMSON monitor request. He does this by entering "SAMSON" from any CRT. This will cause generation of a single display summary of all active and suspended tasks in the system, plus other summary information as described below.

To initate repeating displays, the user enters "SAMSON mmss ccc pp", where mmss is the interval in minutes and seconds at which the user wants the displays to change, ccc is the number of times the user wants displays made (i.e., the number of periodic cycles at the foregoing interval), and pp is the number of seconds that the user wants each page of the display held on the screen for viewing. Default values for these parameters are 1-second cycles and 1 display, held for 1 second. If the number of cycles is set to 000, displays will be presented indefinitely. Nonnumeric input to the mmss, ccc, or pp field causes abnormal termination of SAMSON, but has no other effect upon the operating environment.

The display generated has a heading line that shows: the time of day for the display (hours, minutes, seconds, and tenths of seconds), number of tasks in the system (not counting the terminal control program), the specified display interval, the number of remaining display cycles, the specified paging interval, and the page number.

Subsequent lines in the display have the format "d ppp iii tttt nnnnnnnn term", where:

- d is the dispatchability status for active tasks, and can be D (dispatchable), N (non-dispatchable), or * (currently dispatched); the suspend status indicators are S (suspended), W (waiting), and E (enqueued); ➤

▷ exactly as it is advertised to, without noticable effect on the performance of their computing systems. The vendor, On-Line Software, Inc., has earned universally high user respect for its technical competence. □

➤
- iii is the 3-digit internally assigned CICS task identification;
- ppp is the task priority;
- tttt is the transaction code;
- nnnnnnnn is the program name; and
- term is the terminal identification.

PERFORMANCE: According to users, SAMSON functions as advertised and has a negligible effect on system overhead.

HARDWARE/SOFTWARE REQUIREMENTS: An OS, DOS, OS/VS, or DOS/VS IBM System/360 or 370 computer running DOS/CICS-Standard Version 1.0 or 1.1 or OS/CICS Version 2.0 or 2.2; but not the as yet unreleased VS versions of CICS. The program requires about 1K bytes of main memory, and resides in main storage.

PRICING: The object deck and user's manual are supplied for a 1-time charge of $500.

INITIAL DELIVERY: November 1972.

CURRENT USERS: 8 as of December 1973. ■

KOMAND Data Acquisition System
Pace Applied Technology, Inc.

MANAGEMENT SUMMARY

KOMAND-Data Acquisition System (DAS) is a dual-purpose program product that offers IBM System/360/370 OS and VS installations comprehensive job accounting and resource measurement utilization capabilities. It generates new accounting data, which is collected along with IBM OS SMF (System Management Facility) data, and builds a precision data base of accounting and utilization information. The data base is then processed to generate a broad range of reports and graphs.

The system is one of the most comprehensive available today, offering job accounting and reporting on many detail levels. It is supported for all standard IBM OS control programs, including multiprogramming, multiprocessing, OS-HASP, and all virtual storage configurations. ASP support is planned. Of particular interest are reports and graphs on virtual storage paging statistics.

KOMAND evolved from a contract between Pace Applied Technology and the State of Illinois in 1970 to develop a custom job accounting system to meet Illinois needs. Pace initially marketed "PACES" (Program Accounting, Costing, and Evaluation System), which represented substantial modifications and improvements to the Illinois System. "PACES" was first installed in early 1971. Support of virtual storage and the addition of modifications and extensions to "PACES" in 1973 resulted in the announcement of KOMAND-DAS, a new program product.

KOMAND-DAS provides management, users, and other EDP personnel with workload processing, equipment utilization, and job charging information with which to evaluate their computer operations. Its reports clearly present resource utilization and charging information at levels ranging from individual job run information to a summary of multiple computer system utilization and charges. Detail utilization information is also available at the step level.

KOMAND-DAS interfaces its own on-line routines with the user exit capabilities provided in the IBM SMF. These routines generate utilization records which are collected along with SMF utilization records to form a comprehensive accounting data base.

KOMAND-DAS is functionally organized into two subsystems: Data Base Generator (DBG) and Data Reporter (DTR). The DBF insures the accuracy and comprehensiveness of the SMF data base, and the DTR provides the reporting, charging, and analytical capability. Of particular interest to those concerned with the performance of virtual storage control programs are the ▷

KOMAND-DAS, a job accounting resource utilization measurement system for IBM System/360 and 370 computers under OS and OS/VS, generates accounting data which is collected with SMF data to form a data base, and also generates a wide range of management reports. Several optional components expand the system's capabilities.

CHARACTERISTICS

SUPPLIER: Pace Applied Technology, Inc., Suite 1100, 1117 North 19th Street, Arlington, Virginia 22209. Telephone (703) 527-4810.

BASIC FUNCTION: Job accounting and resource measurement on IBM System/360 and 370 OS and OS/VS configurations. The technique employed is to generate new accounting data which is collected along with OS System Management Facility (SMF) data via standard user exits, and to build a utilization/dollar charge data base that can be processed by report generation modules provided within the product.

Additional optional facilities are available, including: a Resource Billing System (RBS) to distribute user charges, including a Year-to-Date option; a batch-oriented Data Inquiry System (DIS) to answer queries regarding utilization and dollar charges; and a Direct Access Measurement System (DAMS) to provide utilization and charge data for direct-access storage units to the track level.

OPERATION: During normal computer operations, the KOMAND-DAS on-line routines, which interface at the OS SMF user exits, generate unique job/step utilization data. These routines are coded in Assembly language, and occupy only about 2K to 3K bytes of storage. Major features of the data base generation subsystem are:

- Comprehensive editing for data base integrity.
- Tape/disk mount accounting
- HASP/SMF compatibility via a special support routines.
- User account code validation
- System downtime accounting.
- In-line job and job-step accounting reports.
- Processing of the DAS data concurrently with the main job stream.
- Recovery of "lost" accounting data due to accidental resetting of the accounting file.
- "Interval" accounting of long-running or teleprocessing jobs.
- Detection of accounting data range errors.
- Recovery of SMF buffer data after operating system failures.
- True accounting of JCL errors.

The second part of the KOMAND-DAS system is the data reporting subsystem, which consists of COBOL-coded ➤

VS paging statistics report and VS paging volume and rate graphic plots.

The basic KOMAND-DAS product is complemented by three optional facilities that support more advanced customer requirements:

The KOMAND Resource Billing System (RBS) distributes total EDP charges to users in the form of "invoices", using the KOMAND-DAS master accounting data base as input. The invoices include charges for support services such as tape/disk volume rental, keypunching, training, systems programming, etc. A wide range of analytical and statistical reports is included. A special option provides year-to-date reporting.

The KOMAND Data Inquiry System (DIS) accepts special queries from users regarding utilization and charge data not directly presented in DAS or RBS reports. DIS processes the KOMAND-DAS data base and generates special reports in response to the queries.

The KOMAND Direct Access Measurement System (DAMS) extends the KOMAND-DAS master acocunting data base by generating new data on direct-access space utilization and charges not provided by OS/SMF. Precision utilization measurements of user volumes, on-line data sets, and dynamic (temporary) data sets to the track level thus become available. Like DAS, DAMS generates a broad range of statistical, analytical, and audit/control reports.

The accounting philosophy used by KOMAND-DAS is based upon the principles of consistency, equity, simplicity, completeness, and flexibility (and who could ask for anything more?). Consistency means that utilization measurements (and charges) are essentially the same for multiple runs of the same job, regardless of the multiprogramming job mix. Equity means that the user is charged fairly, in that he is only charged for the resources he has actually used. A prime aspect of flexibility is the computing center's ability to modify its charge rates without reprogramming.

KOMAND-DAS separates computer resources into three categories: dynamic, dedicated and support. Dynamic resources are charged out by a dollar value per actual utilization hour (e.g., CPU time and channel time). Dedicated resources are charged out by a dollar value per "occupancy" hour. (Occupancy time is the time that the job would require if it were running alone in the system; the use of occupancy time eliminates measurement errors due to involuntary wait time.) Dedicated resources include tape and disk drives (user-invoked), main memory, and teleprocessing devices. Support resources are charged out by a dollar value per unit of activity; support resources include spooling and tape/disk mounting functions.

report generation routines. These routines operate in the normal job stream and require only 90K bytes of main storage (except for the cost calculation routine, which requires up to 130K bytes); they can be run at the user's convenience. Unit charges can be easily modified by computer center management by updating the system tables.

Some features of the KOMAND-DAS reports are:

- An audit listing of all jobs submitted.
- Interval accounting distribution data.
- Accounting data error diagnostics.
- Computer center revenue analysis.
- Computer center summary utilization data.
- Utilization data for each device address.
- Resource utilization data by job and job-step.
- VS program paging activity.
- Record totals for control purposes.
- Revenue simulation capability.
- Abnormal termination reports.
- Performance evaluation information.
- Multiple mainframe accounting.
- Graphic plots of real core utilization, I/O device utilization, TSO region utilization, virtual core utilization, VS program paging volume, and VS program paging rates.

The optional Resource Billing System (RBS) integrates DAS-derived computer processing charges with charges for support functions, including personnel services, training, special forms, tape/disk rental, etc., to present an invoice with all charges. RBS also features summary invoice generation, crediting capability, past balance and credit carry-over, excess expenditure warning, and multiple-level discounts. It also produces reports in the general categories of summary accounting, total charge analysis, and credit/debit analysis. It includes a special Year-to-Date feature that accumulates information for fiscal year accounting purposes.

The optional Data Inquiry System (DIS) is of particular interest to large data center users. It accepts queries on cards in a batch mode to answer operational questions regarding utilization and charge data not directly presented in RBS and DAS reports.

The optional Direct Access Measurement System (DAMS) extends the KOMAND-DAS master accounting data base by generating new data on direct-access space utilization and charges not provided by OS/SMF. It yields utilization measurements of user volumes, on-line data sets, and dynamic (temporary) data sets to the track level. Like DAS, DAMS generates a broad range of statistical, analytical, and audit/control reports.

PERFORMANCE: According to its users, the KOMAND-DAS system and its optional features perform as advertised, are well debugged, and do not cause users to complain of any excess overhead or report processing times.

HARDWARE/SOFTWARE REQUIREMENTS: KOMAND-DAS requires an IBM System/360 or 370 operating under any standard version of OS or OS/VS. HASP is currently supported, and ASP support is planned. The KOMAND-DAS interface to the operating system is resident in main storage and requires about 2K to 3K bytes, while report

KOMAND Data Acquisition System
Page Applied Technology, Inc.

KOMAND-DAS can meet the growth needs of its customers, since it can operate without modification in the OS/MFT/MVT/HASP or OS/VS1/VS2 environments. For further flexibility, the customer can set parameters to establish his own methodology for accumulating utilization and charge data, within reasonable limits. For example, in a VS environment, the customer can elect to have paging times reported separately or included as part of the measured job/step utilization times.

USER REACTION

After having been alerted by our subscribers, who graded KOMAND-DAS very highly in two responses to the first annual Datapro survey of proprietary software users, we interviewed three additional users of the package. The users we spoke with had only praise for the product and its vendor.

One user has had the package since July 1971. His organization provides EDP services for all organizations in a large county on its OS/MFT System/370 Model 155. The customer uses all of the available optional modules and is now looking at DAMS. The package forms the installation's basis for billing its services, and the user states that it is "great from an operational standpoint to optimize partition sizes using KOMAND historical data". This user runs Pansophic's Easytrieve (Report 70E-727-01), against KOMAND files on a daily basis to look at EXECP's (I/O requests) on disk drives in order to reorganize data sets in such a way as to eliminate disk arm contention, first on a per-spindle basis, later on a per-pack basis. The user also says that KOMAND answers most questions about resource utilization, except for sophisticated analysis of CPU and channel wait times (but we'll see what another user has done about that), and provides the basic tools for system monitoring. This user concluded by saying that the package performed as advertised without modifications.

Another user, this one a large life insuror in Canada, has used the system in a 360/50 and 360/65 OS/MFT installation, is currently on an OS/MVT-HASP 370/155, and is going to a 370/168 under OS/VS2, Release 1.6. This user has just ordered the RBS module, and he claims the following four benefits from KOMAND use: (1) one report, Device Utilization per Week, made him aware of a wasted tape drive, and enabled him to justify KOMAND's cost in a few months; (2) the utilization summary consistently points up utilization problems; (3) SMF data is reformatted so that it can easily be handled by COBOL post-processors for exception reporting, etc., on the data base; and (4) he uses KOMAND in conjunction with Boole & Babbage's PPE, CUE, and DSO (Reports 70E-098-01 and 70E-098-02) in program configuration and other system tuning.

A third Datapro interview was with a large U.S. insurance company that currently uses KOMAND on twin 370/165's under OS/MVT and is moving to 370/168's under OS/VS2. This user cites the major benefit of KOMAND as its provision of data for internal performance analysis and external billing – a job that would be "dirty" using just SMF. This installation also uses the package for workload/performance reports for system tuning and planning. The user also has the DIS module, which he points out can make information available without having to write a program to get it.

In summary, KOMAND checks out as a thoroughly debugged tool for advanced system accounting and resource measurement. It appears from user testimonials that it does the former job excellently, and the latter so well that only the most detailed performance studies require additional programming – and, significantly, leading packages can easily process against its data base. Its overhead is quite small: just 2 to 3 thousand bytes for the DAS on-line modules that interface with SMF and are resident in main storage. The remainder of the DAS programs are executed in the normal job stream and use less than 90K bytes, except for the cost calculator which requires up to 130K. Large OS and OS/VS users interested in the functions provided by KOMAND-DAS should consider it seriously. □

➤ **programs operate in the main stream and require up to 130K.**

PRICING: KOMAND-DAS and its features are available only on a license basis, which involves a one-time fee. First-year maintenance is included in the payment as part of the warranty. Multiple-system purchase discounts are available. Installation services are included in the fee, except personnel travel. Supporting consulting, training, and product customization are available under separate service agreements. The following prices are for paid-up licenses and for annual maintenance charges after the initial year.

	License	Maintenance
Data Acquisition System (includes Data Base Generator and Data Reporter)	$10,800	$1,080
Resource Billing System (RBS)	4,850	485
RBS-Year-To-Date Module (YTD)	1,150	155
Data Inquiry System (DIS)	5,600	560
Direct Access Measurement System (DAMS)	9,950	995

INITIAL DELIVERY: January 1971. The optional facilities were delivered shortly thereafter, except for DAMS, which will be available for general use in October 1973.

CURRENT USERS: KOMAND-DAS – about 85 users with 90 installations. RBS – about 25; DIS – about 7; DAMS – 1; YTD – about half of the RBS users. ■

Panvalet
Pansophic Systems, Inc.

MANAGEMENT SUMMARY

Panvalet has been significantly expanded since its original introduction under a different name (Psi-Valet) in late 1969. It is currently in Version 7 as a result of the vendor's continuing enhancement program. Panvalet, with 964 users, may be the leading library routine on the market today. Among DATAPRO 70 subscribers, at least, it appears to be the most widely used and one of the most highly regarded of all proprietary software packages.

There are typically two major thrusts used to market library routines: convenience and economy in handling source records that are frequently used, and protection against disaster. Pansophic probably dwells on the importance of protection more than other vendors, noting the vulnerability to violence exhibited by many commercial computer installations late in the 1960's.

In evaluating Panvalet, several stages of reaction are involved in bringing it into proper perspective. The initial reaction is one of wondering whether all this is really required just to maintain a library of source records. As the various functions begin to sort themselves out, there is a stage when you wonder about the complexities of the many outputs and wonder whether the results are worth the effort of managing the bookkeeping on many different files of source records. Then you begin to see that the responsibilities of the personnel involved are pretty much the same as without a library system, and things begin to fall into place, resulting in the impression of a pretty neat package.

Basically, Panvalet consists of several independent programs that divide the functions of adding/modifying data sets, deleting data sets, and restoring to the library previously deleted data sets into three separately initiated steps. This provides the capability to exercise as much control as is desired over management of the library file. If security is not of prime concern, then the opportunity to batch like functions provides a neatness of operation satisfying to those whose responsibility it is to maintain the integrity of the library.

The primary strengths of Panvalet are its capabilities for maintaining control, flexibility in outputs, and simplicity of use (once you understand it).

Any data that can be recorded in 80-character records can be stored in a Panvalet library.

Special formats for commonly used programming languages, such as BAL, COBOL, FORTRAN, and PL/I, are provided to aid in compressing data for storage in addition to blank compression. Typically, sequence number fields ▷

Panvalet provides capabilities for establishing and maintaining a library of 80-column data on direct-access devices for an IBM System/360 or 370 or UNIVAC Series 70 computer. The package, consisting of several independent programs, features very flexible control procedures for maintaining the integrity of the library.

CHARACTERISTICS

SUPPLIER: Pansophic Systems, Incorporated, Suite 720, 1211 West 22nd Street, Oak Brook, Illinois 60521. Phone: (312) 325-9600.

BASIC FUNCTION: To build and maintain a library of source programs, data, or, in effect, any 80-column data set on an IBM System/360 or 370 direct-access device. The system features flexibility of output for building job streams, transferring data sets to other libraries, etc., and an unusual degree of concern for the safety and integrity of data.

OPERATION: There are six programs in both the DOS and OS versions of Panvalet. Additionally, facilities for maintainance and security of a central library of source and object programs, card-image data, and JCL files are provided by the PAN Command Processor in a TSO environment and by the PAN #1 program in an OS execution mode. PAN #1 is the basic program for adding and modifying data sets; in addition, selected data sets can be output to create a job stream file, and data sets can be flagged for deletion.

In the Pansophic philosophy of how a data processing installation ought to manage a data library, PAN #1 is used by the programmers to create, modify, and test programs. The other programs, and hence functions, are restricted to use by management or supervisory personnel. The basis for this philosophy stems from the desire to limit as much as possible the effect of deliberate or inadvertent destruction of parts or all of the library. The emphasis is on protection from willful destruction or natural calamity, but there are several safeguards to ensure that users are working with the latest version of the library contents and that proper command names have been entered.

PAN #2, Management's Program, allows the whole library to be dumped for backup protection; it is also used to rewrite previously deleted data sets back into the library. PAN #4 (there is no longer any PAN #3) is the Library Initialization Program. It is used to initialize, format, and label disk packs or data cells intended for holding a Panvalet library.

PAN #6 (there never was a PAN #5) is the Analysis Program; it is used to produce a summarized presentation of a Panvalet library's composition. PAN #7, the Cross-Reference Program, lists the location of all Panvalet INCLUDE statements and produces a detailed cross-reference list. Finally, PAN #8, the Scan Program, produces a listing of statements within the library which contain a user-specified character string. ➤

are eliminated, and new sequence numbers are generated each time a data set is output.

The library file can be maintained either by an IBM System/360 or 370 operating under DOS or OS or by a UNIVAC Series 70 operating under DOS or TDOS. Any direct-access storage device supported by the computer's operating system can be employed. Pansophic has installed Panvalet on all delivered equipment combinations.

Pansophic is a consulting firm oriented toward providing services to corporate and data processing management. The firm once marketed Panmaster, a separately developed auxiliary storage management system. In mid-1973, Pansophic became the exclusive marketing and support agent for Easytrieve (Report 70E-727-01).

USER REACTION

In addition to contacting six Panvalet users whose names were provided by Pansophic, Datapro analyzed the responses to its May 1973 users' survey of proprietary software. Interestingly, Panvalet turned to be the leading package in terms of frequency of use, and was also one of the most universally and highly praised.

Among the 191 proprietary software users reporting, Panvalet had 29 users. The responses to the survey questions showed Panvalet earning ratings that were consistently in the good or excellent range. Importantly, all users agreed that Panvalet performs as advertised immediately, and only 3 of the 29 users found any need to modify the package.

The aspect of the program most mentioned by users was security, with backup a close second. Several users liked the elimination of card handling. Users were divided on the controls in the PANVALET system; some praised them while others found them inadequate. The division on this was about 33% to 33%, with the other one-third of the users abstaining.

Users interviewed by Datapro had System/360 Models 30, 40, and 50 and System/370 Models 135 and 145. Most agreed that the PANVALET programs used at most 44K bytes. Overhead in a test environment was rated at an easily affordable 3 percent.

In response to a user's question as to where the names PANSOPHIC and PAN #1, etc came from, we looked into our dictionary and found: PAN–involving all of a specified group; whole, or general; and SOPHIC (per se)–the root of such words as sophisticate. What you get is an all-inclusive, sophisticated valet to handle your program wardrobe. □

PANVALET LIBRARY: A Panvalet library can be stored on any IBM System/360 or 370 or Univac Series 70 direct-access storage device (e.g., 2311, 2314, or 3330 Disk, 2321 Data Cell), and can be changed from one to another of these devices if reconfiguration or other reasons make that action desirable.

Information is stored in records of up to 80 characters. For some formats, such as COBOL and FORTRAN source statements, certain fields (primarily sequence and identification number) are removed. In addition, blanks are compressed. Usually, records are well under 80 characters long. Records are blocked onto tracks, at any specified blocking factor. A maximum of 65,535 blocks constitute one Panvalet library, regardless of the type of direct-access device used. One data set can be an entire library. Typical for an entire library are about 250,000 source cards (2311), 1 million cards (2314), or 4 million cards (3330). About 1 million cards can be contained on a 2321 Data Cell. These figures are based on a survey of Panvalet users, who reported an average figure of about 30 bytes per source card.

The attributes of a data set include name, level, type, user code, status, and activity. All except user code have some effect on library maintenance. Data sets are stored in random sequence at the block (track) level and are accessed by name (up to 10 alphanumeric characters). Only one level, or version, of a data set is permitted in the active library file. (Duplicating a data set with a slight name change is the standard and convenient way to work with multiple versions of the same data set.) Several types of data sets can be identified, such as COBOL, AUTOCODER, or FORTRAN source statements; such identification allows greater compaction in the library file. Alternatively, a "no-format" data set type can be specified to preserve all 80 columns.

Status and activity are interrelated status indications. There are three complementary pairs of status and activity indications. A data set can be identified as PRODUCTION/TEST, ENABLED/DISABLED, and ACTIVE/INACTIVE. The principal significances of these indications are that PRODUCTION and DISABLED data sets cannot be modified and that a user can distinguish between DISABLED and INACTIVE data sets for group deletion.

DATA SET ADDITION AND MODIFICATION: PAN #1 provides a set of commands that can be broken down into three basic groups: add a data set to library (1 command), modify a data set (6 commands), and effect output of a data set, including slight format control (7 commands).

When a data set is added to the library, it is automatically identified as level 1, with a status of TEST, ENABLED, and ACTIVE. As long as the TEST status is maintained, the program can be modified. However, as soon as the PRODUCTION status is invoked, no changes can be made under any circumstances by any of the Panvalet programs. The only thing that can be done is to delete the data set.

A total of 11 different data set types, or formats, can be identified when a data set is added to the library, including AUTOCODER, BAL (or ALC), COBOL, COBOL-72 (non-standard COBOL), FORTRAN, PL/1 (or PL/I), RPG, OBJECT, JCL, DATA, and OTHER. These labels are used in the Directory listing of the library and to control internal formatting of records.

Four of the commands in PAN #1 for modifying data sets are used to rename a data set, change the modification level number, change the status, and change the user code, a four-digit number printed in the Directory Listing for user convenience.

Panvalet
Pansophic Systems, Inc.

Two commands are used to modify the source records in a data set. One identifies that an update is being made; it is followed by a series of commands specifying the actual change to be made. Each change command is followed by the new statements (records), if any, to be added.

The change technique is simple, straightforward, and very effective. All that is required is the Panvalet sequence number of the statement where an insertion is to be made; all new records in the data stream before the next command will be inserted before the next old statement. If deletions are required, two sequence numbers are specified and all statements between the two, inclusive of the specified numbers, will be deleted. Both operations can be combined in the same command to give a particularly straightforward way of writing update commands. The heading command for the update operation can include a specification that allows the complete data set to be replaced without the need to write additional change commands. Each set of update commands automatically increases the modification level number by 1.

Input to PAN #1, the file of commands and user data, can be on punched cards, magnetic tape, or disk, but only one job stream can be accommodated.

Data sets can enter the library in two other ways besides a PAN #1 add operation. A copy function within PAN #1 allows a data set to be duplicated with a different name. Such a data set is automatically assigned a status of TEST, even if the data set being copied carries a PRODUCTION status. Essentially, this is the only way a PRODUCTION data set can be accessed for modification.

Data sets that have been deleted from the library (PAN #2) and written onto a protection file can be re-entered into the library through a PAN #2 REPLACE or RESTORE operation. A RESTORE function implies that there is not a data set with the same name as the one coming in from the protection file already in the library; this operation is oriented toward handling seldom-used data sets that are kept in a protection file rather than occupying space in the library. In a REPLACE operation, the data set coming in from the protection file replaces a data set with the same name, if any, already in the library; a TEST data set cannot replace a PRODUCTION data set. This operation is convenient for returning to a previous production method.

DATA SET OUTPUT: The possible outputs from the Panvalet system include work files and data set listings (PAN #1), library dump (PAN #2), protection files (PAN #2), and transfer files (PAN #2).

Four PAN #1 commands used together form a powerful facility for creating a work file. The basic WRITE command initiates output of data sets by name to the disk, magnetic tape, punched card, or printer file. The punched card and printer files can be the unit record devices themselves or magnetic tape or disk for later transcription. The printer file is the only means for obtaining a listing of the source records.

Source records from the input stream to PAN #1 can be transferred to the work file without being added to the library. In addition, selective portions of a data set can be output. A further facility allows embedding a call for a data set name within another data set; when a data set is output, any embedded calls are honored. Any number of INCLUDE calls can be placed in a data set, but if they are nested, all beyond the first level are output as a source record and do not result in a call to the named data set.

When outputting a printer file, page overflow can be initiated at any time by a special command. In addition, an ID comment record can be output as the first page. Standard ODS or OS job stream terminators (/* and/or $/) can be added to the end of a work file.

DATA SET DELETION: Data sets can be flagged for deletion through a PAN #1 status change command. Such data sets can be identified as DISABLED or INACTIVE. The actual removal from the file can only be accomplished through PAN #2. Deletion operations can be identified as all data sets flagged DISABLED, all data sets flagged INACTIVE, or by individual data set name (any status). Flagging in PAN #1 accomplishes nothing other than identification.

The reason behind the two-level deletion process is security; i.e., throwing away only what you want to throw away. However, unless the initiator of PAN #2 (a manager, supervisor, or systems analyst) has carefully checked a directory listing (also output by PAN #2) just before he initiates a deletion run, the possibility exists that a particular data set many have been incorrectly flagged for deletion. This is an inconvenience rather than a disaster because all data sets deleted are written to a protection file; such a file must be on-line or PAN #2 will not execute. Identification of data sets removed is provided by a printed report.

LIBRARY OUTPUT: To safeguard against a computer-room disaster, a complete dump, in compressed format, of the entire library file can be made at any time through PAN #2. If selected data sets are desired, say for shipping or transmitting to another Panvalet library, PAN #2 can be called in; this operation also results in a compressed format output.

LIBRARY SECURITY: Other than by losing a tape reel or disk pack, the only ways in which source records can be irrevocably lost are through normal modifications to a TEST data set or through one of the delete/replace operations. Deleted data sets are written to a protection file. If a data set with the same name is already on the protection file, it is expunged. If, for some reason multiple copies of a version of a program are desired, VERSION = 0 allows the user to keep several copies of the same named data set; bookkeeping can then become a problem. When a previously deleted data set is moved from a protection file back to the library, or when a data set is replaced from a DUMP or TRANSFER file, the copy already in the library, if any, is lost; again, careful bookkeeping is required.

Protection against normal programming errors is accomplished through several facilities. PRODUCTION data sets cannot be modified and cannot be replaced by a TEST data set. When modifying a program, all commands must be successful before the update is considered successful; i.e., partial updates are flagged. In addition, the execution of any command or group of commands can easily be made conditional on the successful execution of a preceding command.

All deletions and creation of TRANSFER files are performed through PAN #2. If desired, the execution of this program can be made dependent on supplying a code in the input stream. The code can be a simple five-digit number or can be internally calculated from two factors supplied in the input stream. The need for such a control code can also be eliminated. As installation can make it easy or difficult to obtain unauthorized access to the deletion functions of PAN #2, depending on its own need for security.

Panvalet
Pansophic Systems, Inc.

➤ DIRECTORY LISTING: A complete listing of the contents of the library can be made only through PAN #2. In addition to basic information, such as name, modification level, user code, status, date of last maintenance, and type of last action taken for each data set in the library, the printed listing can include the number of statements, blocks, average bytes per record, and per cost of assigned space actually utilized.

RECENT ENHANCEMENTS: Panvalet is now officially in "Generation II, Version VII," and some enhancement features have been added since Datapro first reported on the program. They are summarized in the following paragraphs.

A new SUPPRESS command in PAN #4 provides additional security by allowing DP management to suppress certain commands and parameters in PAN #1; a CONTROL command and the proper control code allows knowledgable PAN #1 users to temporarily use the suppressed PAN #1 commands and parameters.

Two optional parameters were added to the PAN #4 CLEAR command to expand user control over the process of initializing the Panvalet libraries; this is why a data set can now contain 65,536 direct-access storage device blocks and why block size now ranges from 500 bytes to full track capacity on the storage device.

User-written routines can now be link-edited to PAN #1 for special processing of input stream and/or output file records, because two program exits are now provided in PAN #1.

Storage and retrieval of small files and data sets (such as JCL) have been simplified by defining new types of data sets as subsets and supersets. A superset stores data in the form of subsets. Subsets can be attached or deleted from supersets, and can be selectively retrieved or included by using a qualified name consisting of the superset name and the subset name. They can also be retrieved collectively by referencing just the superset name.

A new PAN #1 COMMENT command allows placing of up to 50 characters of additional information on a library data set. It can be reported when requested by the user.

The PAN #1 command UPDATA now has a TEMP (temporary) parameter for creating intermediate-duration data set modifications to a data set.

Data sets can now have optionally assigned version numbers to provide a means of creating a historical protection file of deleted data sets.

In addition to the foregoing, dumps are now said to be faster, and certain functions of the dumps are automatic for DOS users and semiautomatic for OS users. Finally, enhancements to already existing Panvalet features provide such functions as reports on data set statistics after an update or addition, statistics at end-of-job, library status and activity report, date of last access, partial additions, error codes and documented messages, and more extensive file checking. If any portion (disk extent) of the library is missing or in improper order, processing is discontinued and an error message is displayed.

TIME-SHARING OPTION: The PAN Command Processor under TSO allows a programmer to request Panvalet functions in a conversational mode. A special technique for updating and handling internal format options in TSO Panvalet allows TSO-environment programmers to forget any incompatibilities between batch and TSO systems.

The TSO user can reference members of the Panvalet library directly from the TSO terminal by invoking the name PAN. The Command Processor supports special sequence-number processing for the ASM, COBOL, FORT, PL1, and CNTL TSO source language formats, and the sequence numbers are retained and/or generated to provide complete compatibility with the TSO EDIT feature.

The Panvalet functions available to the TSO terminal user are: ADD, COPY, CHANGE (alters LEVEL, STATUS, or USER CODE), RENAME, RETRIEVE (WRITE), AND STORE (UPDATE . . . ALL).

Provisions have been made for the TSO user who desires overnight batch processing. These features, in summary, are: (1) the program is resequenced by 10's as it is added to the library; (2) a complete comment record is produced as a data set is retrieved; (3) for COBOL, the ID portion of the statement will be blank after the first 3 cards; (4) for TSO EDIT, the INCLUDE commend is not expanded, so as to allow batch or immediate compiles following the edit and storage back into the library with the INCLUDEs expanded; (5) the command processor can store a module on the library; and (6) a batch-mode update of a program can be made using PAN #1.

HARDWARE REQUIREMENTS: Panvalet is available for just about any IBM System/360 or 370 configuration running under DOS or OS that also includes Decimal Arithmetic. The DOS version requires a partition size of from 22K to 34K bytes. The variations depend on whether a 2314 is used (8K) and whether the protection file is match-merged during a PAN #2 deletion run (4K). The OS version requires a partition or region size of 26K bytes (2321), 28K bytes (2311), or 36K bytes (2314). It can also be used with the nearly equivalent UNIVAC Series 70 (former RCA) systems. For the TSO option, region or partition sizes for PAN can range from 42K to 60K bytes depending on the block sizes chosen for the Panvalet Library.

SOFTWARE REQUIREMENTS: Panvalet will run in a background or batched foreground partition under Release 18 or later of DOS. The OS version runs under PCP, MFT I/II, or MVT. Panvalet will operate in conjunction with such support programs as POWER under DOS and HASP under OS. The TSO version requires OS with the Time-Sharing Option. Panvalet also runs under DOS/VS, OS/VS1, and VM/370. It is presently running at test sites under .OS/VS2.

PRICING: Panvalet is available under perpetual license for $4,980 per site plus $600 per year maintenance after the first year. The Time-Sharing Option's perpetual license fee is $2,400 per site, and the source code can be obtained for $17,500; each of these options also bears a maintenance fee of $600 per year after the initial year. Training (½ day) is provided free for the basic package and the TSO option, and additional training is available at $350 per day, with a 1-day minimum. An unlimited license is available for a flat fee of $27,500.

Rental terms for Panvalet on a monthly basis include maintenance. The basic package rents for $210 per month, and the TSO option also rents for that amount.

Both the perpetual license fees and monthly rentals are subject to discounts of 10% on the 2nd through 5th installation and 25% on the 6th and subsequent installations.

INITIAL DELIVERY: December 1969.

CURRENT USERS: About 964 as of June 1973. ■

PHI Generalized Payroll System
PHI Computer Services, Inc.

MANAGEMENT SUMMARY

The PHI Generalized Payroll System is one of the early "best sellers" among computer programs. Originally sold only to banks, it is now available to individual firms, both for their own use and for resale.

The PHI Generalized Payroll System is one of the most successful systems currently available. Many of the approximately 115 current users are banks which service upwards of 100 different payrolls. Thus, there are actually many hundreds of corporations using the PHI system for their payrolls.

Accordingly, you may want to consider the use of the PHI system in one, two, or all three of the following ways. First, your company could obtain the system strictly for its own use. Second, you could obtain it for your own use and also market payroll processing services to other companies. Third, you could have your company's payroll processed by a bank or service bureau that uses the system.

Behind each of these alternatives is the question of the efficiency and flexibility of the PHI Generalized Payroll System. Its history of sales and use, with little modification found necessary, suggests very strongly that the service is efficient–at least in a banking atmosphere. Perhaps this is a necessary part of its success. The computer programs–not the user–dictate the form design, check sizes, and, in a general way, the reporting structure. Whether used through a service bureau or a bank, or operated in-house, the PHI System is capable of delivering very satisfactory throughputs.

In the Spring of 1971, Cullinane Corporation and PHI announced a special version of Cullinane's CULPRIT report generator (see 70E-272-03) specially tailored for PHI's Generalized Payroll. The new version is called Payout. It allows PHI Generalized Payroll users access to all files including Labor Cost, Payment History, and Sorted Transaction information. Extended formatting and computational facilities are included in Payout, and a total of up to 99 reports per pay cycle. In addition to the benefits of extended reporting, Payout executes faster than the built-in report generation system. However, it may not be as easy for non-programmers to learn due to the highly compacted statement form. Payout goes for $5000 and can be acquired through either Cullinane or PHI

In May 1971, PHI introduced a more powerful payroll system, Payroll II, for use on System/360 or 370 computers with a minimum of 90K bytes of core storage. Payroll II is similar to the Generalized Payroll System in functional capabilities, but features more versatility and fewer restrictions. It is fully described in Report 70E-684-03. ▷

CHARACTERISTICS

SUPPLIER: PHI Computer Services, Inc., 800 Massachusetts Avenue, Arlington, Mass. 02174. Telephone (617) 648-8550.

BASIC FUNCTION: The PHI Generalized Payroll System is a software package that provides for the maintenance of a number of different master payroll files. Each file is handled as an independent unit capable of providing for specialized reports and management decisions; yet all the files can be combined and processed as a single unit so far as standard functions, reports, and operating procedures are concerned.

The program is written in COBOL, and is designed to serve the needs of a centralized or service bureau payroll operation.

INPUT: All inputs and outputs are standardized. The format of the input cards is the same for all companies, regardless of how each company transmits the data into the computer center.

The major inputs for each cycle consist of job cards and/or time cards. The system accepts both types of cards, and as many separate cards as are necessary can be entered for each employee.

The prime difference between the job cards and time cards is the "override" feature, included only on job cards. This allows the rates for a particular job to be independently set on a one-time basis instead of being obtained from the master file. This feature is restricted to only the first four categories out of the ten possible, so that the job card size can be held to 80 columns. By contrast, the time cards can contain entries for any or all of ten categories.

The use of these terms such as time cards, however, can be misleading. While the cards are normally used for time card purposes, they can also be used for other functions. A time card, for instance, need not represent time. It might be punched with the number of bonus dollars, pieces completed, shoes made, dogs clipped, etc., depending upon how the employee makes his living.

The payroll master file is sequentially arranged and contains two types of records: "company headers" and employee records.

Company headers include company name and address, the company names given to various earnings and deductions, the type limit codes which are used to control calculations, report formats, and specifications, and certain constants which are common to all employees. Each company header effectively controls the processing of that company's payroll within the PHI Generalized Payroll System. ➤

PHI Generalized Payroll System
PHI Computer Services, Inc.

Technically, there can be no doubt that the PHI Generalized Payroll System is one of the major achievements of the computer services field to date. It has automated the production of disparate payrolls very effectively. It provides a company with an easy way to move into more intricate job costing or deduction handling–and to do it securely. Its place in the service bureau industry is firm: and if the data processing manager recommends that it be used, he will be standing on solid technical ground. □

Each employee record is 1360 bytes long. Over and above the necessary payroll information, it includes an area for personal data such as telephone extension, college degrees, wife's name, etc.

PROCESSING: The payroll is handled in a two-phase, batched processing manner. All companies' payrolls are normally run at once, although if the number of payrolls with other than weekly periods becomes large, they may be kept on separate tapes. No multiple tape mounting is required for companies processed consecutively.

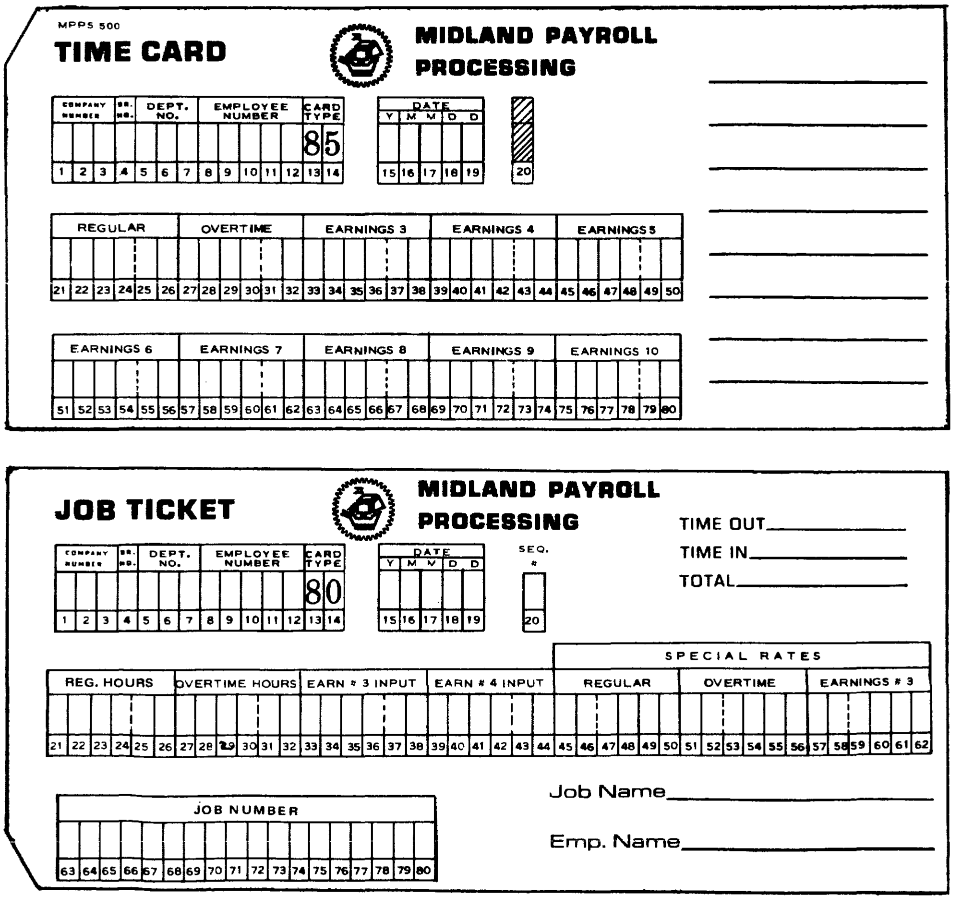
The first phase of the operation takes place as soon as all the input is ready, without sorting or computing control totals. The first phase is more than an editing run; it does all the updating and the calculations, but prints only what is needed to verify the input. Phase I operations include the production of a consolidated Proof Journal showing the hours and extended amount for each time card, old and new contents of each field that has been changed during the update operations, notifications of changes or transfers between departments, error messages for non-matching input cards, or apparently erroneous calculation results.

Computer processing stops after Phase I in order for input control totals to be balanced. The information provided in the Proof Journal is normally sufficient to eliminate the need for input card listings if totals do not agree with predetermined results. It is normally necessary to re-run Phase I after the errors have been reconciled.

Phase II is run after Phase I has been balanced. The updated master file produced as a result of the first phase is used to print all major reports such as checks, registers etc.

CALCULATIONS: The calculations are broken into two main areas: earnings and deductions. Earnings are usually only computed where the company is active and the employee is active. Up to 10 different types of earnings are allowed; some, such as salary, are computed automatically, while other require specific card input. Earnings No. 1 is always regular earnings, and No. 2 is always overtime. Numbers 3 through 10 can be defined and labeled by the customer. The labels identify the various amounts printed on the statement and registers. Earnings position No. 4 is usually used to hold the number of pieces produced, so that comparisons can be made with standard costs.

Most customers use both time cards and job cards, and mixtures of both are permitted. Job tickets provide entry data for the first four types of earnings only, including the



MPPS 500

TIME CARD

MIDLAND PAYROLL PROCESSING

COMPANY NUMBER	BR. NO.	DEPT. NO.	EMPLOYEE NUMBER	CARD TYPE	DATE Y M M D D	
				85		
1 2 3	4	5 6 7	8 9 10 11 12	13 14	15 16 17 18 19	20

REGULAR	OVERTIME	EARNINGS 3	EARNINGS 4	EARNINGS 5
21 22 23 24 25 26	27 28 29 30 31 32	33 34 35 36 37 38	39 40 41 42 43 44	45 46 47 48 49 50

EARNINGS 6	EARNINGS 7	EARNINGS 8	EARNINGS 9	EARNINGS 10
51 52 53 54 55 56	57 58 59 60 61 62	63 64 65 66 67 68	69 70 71 72 73 74	75 76 77 78 79 80

JOB TICKET

MIDLAND PAYROLL PROCESSING

TIME OUT__________

TIME IN__________

TOTAL__________

COMPANY NUMBER	BR. NO.	DEPT. NO.	EMPLOYEE NUMBER	CARD TYPE	DATE Y M M D D	SEQ. #
				80		
1 2 3	4	5 6 7	8 9 10 11 12	13 14	15 16 17 18 19	20

				SPECIAL RATES		
REG. HOURS	OVERTIME HOURS	EARN # 3 INPUT	EARN # 4 INPUT	REGULAR	OVERTIME	EARNINGS # 3
21 22 23 24 25 26	27 28 29 30 31 32	33 34 35 36 37 38	39 40 41 42 43 44	45 46 47 48 49 50	51 52 53 54 55 56	57 58 59 60 61 62

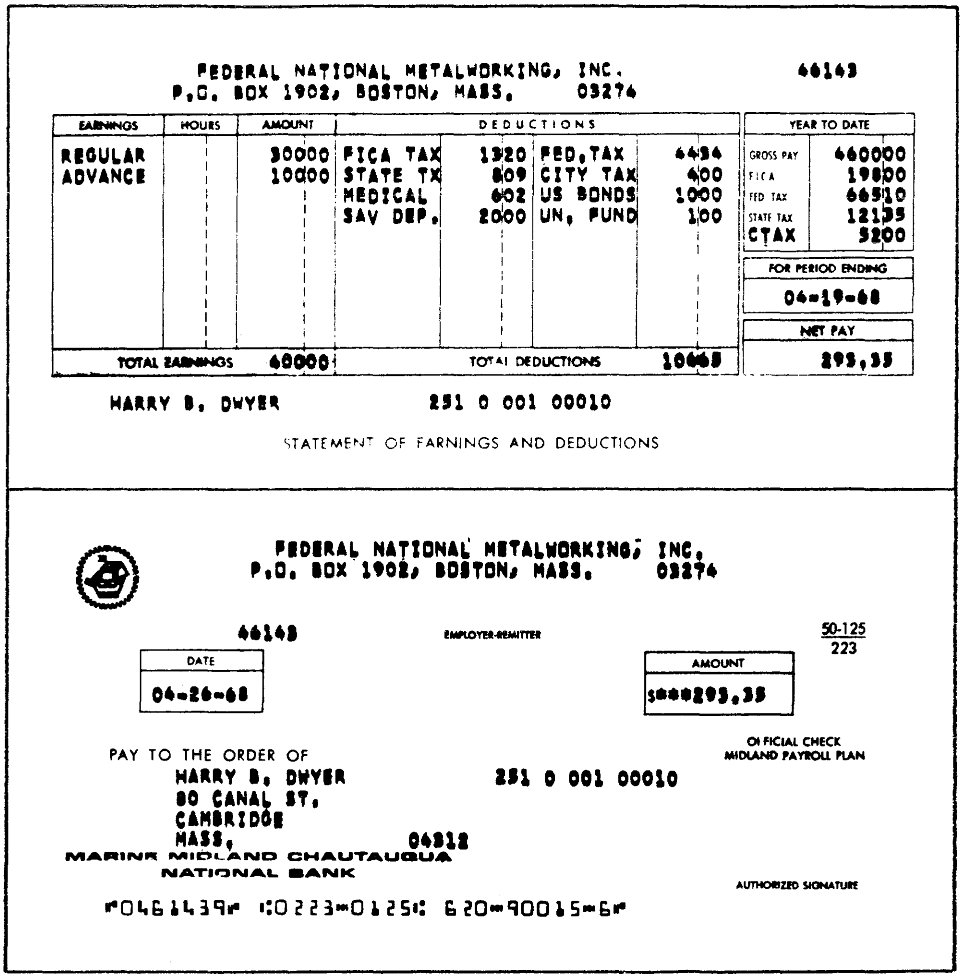
JOB NUMBER
63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80

Job Name__________

Emp. Name__________

The time card (top) and job ticket forms shown here clearly illustrate the differences between the two standard methods of reporting earnings to the PHI system.

PHI Generalized Payroll System
PHI Computer Services, Inc.



FEDERAL NATIONAL METALWORKING, INC. 46143
P.O. BOX 1902, BOSTON, MASS. 03274

EARNINGS	HOURS	AMOUNT	DEDUCTIONS				YEAR TO DATE	
REGULAR		30000	FICA TAX	1320	FED.TAX	4434	GROSS PAY	460000
ADVANCE		10000	STATE TX	809	CITY TAX	400	FICA	19800
			MEDICAL	602	US BONDS	1000	FED TAX	66510
			SAV DEP.	2000	UN. FUND	100	STATE TAX	12135
							CTAX	5200
TOTAL EARNINGS		40000	TOTAL DEDUCTIONS			10665		

FOR PERIOD ENDING
04-19-68

NET PAY
293.35

HARRY B. DWYER 251 0 001 00010

STATEMENT OF FARNINGS AND DEDUCTIONS

FEDERAL NATIONAL METALWORKING, INC.
P.O. BOX 1902, BOSTON, MASS. 03274

46143 EMPLOYER-REMITTER 50-125/223

DATE	AMOUNT
04-26-68	$***293.35

OI FICIAL CHECK
MIDLAND PAYROLL PLAN

PAY TO THE ORDER OF
HARRY B. DWYER 251 0 001 00010
80 CANAL ST.
CAMBRIDGE
MASS. 04812

MARINE MIDLAND CHAUTAUQUA
NATIONAL BANK

AUTHORIZED SIGNATURE

⑈0461439⑈ ⑆0223⑉0125⑆ 620⑉90015⑉6⑈

Principal output documents from the PHI system are the deduction slips (top) and paychecks (bottom), which are printed in side-by-side fashion for each employee. The standard formats shown here must be adhered to in all cases.

➤ regular and overtime groups. In addition to regular and overtime pay rates, special override rates may be entered on job cards. These rates only affect the current calculations, and not the master-file data.

Tax liabilities are indicated separately for eight different categories. These differ from each other and from gross earnings, depending upon the types of earnings and who is collecting the tax. An example might be a restaurant worker who must pay F.I.C.A. on his tips, but whose employer does not have to pay matching F.I.C.A. on the tips and may or may not (depending upon the state concerned) have to pay unemployment taxes on these tips either.

Deductions are handled in 15 separate categories for any particular company, with 3 used for F.I.C.A., Federal, and state taxes.

OUTPUT: The system produces a number of different outputs, including the following ones.

A standard check is normally used for all payrolls, with the name of the company printed by the computer. Standardized checks printed with the company name are available, but are more expensive. Where the amount due to the employee is deposited into a bank account, a "Notification of Deposit" in the same format as the checks is used. Pay Receipt and Memorandum statements are also prepared for employees paid in record adjustments.

A Payroll Register, which is a combined report of earnings and deductions, or separate Earnings and Deduction Registers are supplied to each company. The Payroll Register provides, for each individual, 12 standard fields and 15 additional company option fields; but it cannot handle all the different types of earnings and deductions allowed for in the payroll system itself. The developers estimate that 80% of the companies using the system will accept the Payroll Register, although alternatively a detailed report consisting of separate Earnings and Deductions Registers can be produced.

The sequence of the Payroll Register is employee number within department and branch. Totals are shown by department and by branch.

The separate Earnings Register spreads the ten earnings categories across the page, in 10 columns starting with Regular Earnings. Current and year-to-date figures are also provided. Partially carbonized earnings logs can be obtained and are used as data collection reports by the client companies to report the next week's work. (This method cannot be used when job costing is required.)

The separate Deductions Register is similar to its counterpart, the Earnings Register, and shows details of all the 15 possible deductions. ➤

PHI Generalized Payroll System
PHI Computer Services, Inc.

The Totals Recap Report is a one-page summary that completes the payroll cycle by branch and company. A "super company" recap can be provided for a group of companies. The Recap Report is used to balance the system, and to provide summary entries for the user to post into his books.

The Transaction Proof Report, which is produced as a part of Phase I, shows in detail how each incoming transaction card was processed. This report notes the errors that were detected and assists in the balancing of the Earnings Log, etc. Any errors detected by the computer programs are printed on the Proof Report, together with an error count.

A Deductions Not Taken Report, showing where the available pay is insufficient to meet particular deductions, is produced. Where deductions, such as insurance premiums, are normally recovered during the next pay period, the system produces a special Receovery Card to be added to the next cycle's input to affect this.

Custom Reports, specifically designed for each customer, can be produced. These include four variable fields in addition to the standard individual data for each man. Some of this data may come from the 200-byte individual area included with each master-file employee record. This report can be sorted on one of four fields and can be controlled by the contents of a particular field so that not all the personnel have to be included on it.

REPORT GENERATION: A special generator system is included as a part of the PHI system. This generator is set up to create reports from the contents of the master file, including data from the 200-byte section that is reserved for individual use by each company.

The report generator provides arithmetic and logical selection capabilities for extracting and manipulating the data in the file. Output is in the form of unblocked card or printer images sutiable for re-entry into the computer or for a variety of other purposes.

Up to 36 separate reports can be produced in a single pass of the master file. The report requirements are defined, using a special report-generation language, for each type of output required. For instance, where there is a requirement for both a card-output file and a printed record of its contents to be created, they are run as two separate reports, rather than as two output formats of a single report.

An important use for this facility is to allow maintenance of the master file itself under unusual circumstances. The report generator can be set up to produce change cards ready for re-input into the normal runs after special tests and computations are made for each employee. This type of maintenance could be used advantageously, for example, in the case of a union settlement that involves job rates and is based on a series of variables such as length of time in the firm, as well as on standard factors such as job category.

A different type of use of the reporting facility is made by some users in conjunction with the 200-character private area in each employee's record. This is to provide a personnel information service for general management use. The additional area is used to hold a variety of items including education, experience, etc. (often in coded form). The report writer is used to print out and label these items in answer to typical personnel queries. This capability can be used quite effectively non-programmers as well as by programmers.

PERFORMANCE: In-house runs of the PHI Generalized Payroll System have produced the following computer timings, based on one complete cycle of 2800 pays from a tape file containing 4000 records involving several companies:

	IBM 360 Model 40	IBM 360 Model 50	IBM 360 Model 65
Phase 1 edit run	27.3 min.	7.8 min.	6.5 min.
Phase 1 actual run	27.3	7.8	6.5
Phase 2	21.0	4.2	2.4
Phase 3	22.0	8.4	4.8
Total	97.6 min.	28.2 min.	20.2 min

Model 40 times include on-line printing. On Models 50 and 65 the printing time is omitted, as this is handled through an output writer in a multi-task environment.

HARDWARE/SOFTWARE REQUIREMENTS: A System/360 or 370 under DOS or OS with at least 65K bytes of core storage and capacity for 5 high-speed files. In general it is preferable to have the master file on tape and the intermediate files on disk.

Other systems, with equivalent core storage and peripherals, that can run the package include the Burroughs B 3500, CDC 3300, UNIVAC (ex-RCA) Series 70.

PREINSTALLATION REQUIREMENTS: A certain amount of basic work must be undertaken before the PHI system is brought into use. The system itself is never modified, so that all the work involved is in preparing the user's files and procedures to dovetail with the system rather than in modifying the system to meet the user's needs. The amount of work therefore varies greatly from company to company and normally takes from a minimum of two weeks to a matter of months.

Typically, the program is brought onto line gradually, with perhaps two thousand employees being paid in the first operational period, four thousand the next, etc., until the previous system can be discontinued.

PRICING STRUCTURE: A license for an IBM System/360 or 370 (OS or DOS) costs $20,000 and provides maintenance to a single installation for 12 months. Thereafter, maintenance costs $2,400 per year for each site maintained. The cost for other systems may vary slightly. Service bureaus are charged $20,000 for the first geographical site involved, $8,000 for the second one, and $1,000 for subsequent bureaus.

The price above includes installation and training; three man-weeks of effort, not including travel expenses, are provided to the customer to be used in whatever distribution of effort the customer desires.

INITIAL DELIVERY: The system was originally delivered in 1966 and was organized for use by banks, in processing the payrolls of their customers.

CURRENT USERS: As of September 1972, there were about 115 users. ■

Payroll II
PHI Computer Services,Inc.

MANAGEMENT SUMMARY

Faced with larger users' interest in more flexibility and with some formidable competitors on the national scene, PHI rewrote its Generalized Payroll System (Report 70E-684-01) in 1971, adding just the right items to keep it competitive for large-scale payroll installations.

Payroll II is a completely new system that uses many of the same computational techniques as the PHI Generalized Payroll package. The most visible differences between the two systems are in the methods of implementation and in the resulting capabilities.

The Generalized Payroll is delivered in the same source form to all customers. The COBOL source code for Payroll II is generated fresh for each user, incorporating the specific options desired for each installation.

Some of the most important capability changes include a large number of types of earnings and deductions (a combined total of 99 should keep anyone happy), a variable master-file key, and an expansion of the non-payroll information areas in the master file. Coupled with the report generator in the package, the increase in the employee information that can be stored presents greatly enhanced opportunities for generating management reports. Other changes from the Generalized Payroll package are summarized in the accompanying table.

Intended for large-scale users (128K IBM System/360 or larger), Payroll II features custom generation of a payroll program set tailored to fit the customer's needs. It is similar in functional capabilities to PHI's Generalized Payroll System, but with significant extensions.

CHARACTERISTICS

SUPPLIER: PHI Computer Services, Inc., 800 Massachusetts Avenue, Arlington, Massachusetts 02174. Telephone (617) 648-8550.

BASIC FUNCTION: Intended for large users, to provide more flexibility than the PHI Generalized Payroll System in the areas of the number of different types of deductions and earnings, the sequencing of checks, and the amount of non-payroll information that can be stored in the master file.

OPERATION: With the exception of several added decisions required when installing Payroll II and the added flexibility of the extensions, the functional capabilities of Payroll II are exactly the same as those of the PHI Generalized Payroll System (Report 70E-684-01).

The extensions are summarized in the accompanying table.

DIFFERENCES BETWEEN THE TWO PHI PAYROLL SYSTEMS

Feature	Generalized Payroll System	Payroll II
Earnings and Deductions	10 earnings; 15 deductions	Combined total of up to 99
Checks –		
Y-T-D totals on stub	4 standard; 1 optional	4 standard; 6 optional
Sequence	Master file sequence only	Any sequence based on a separate report key
Master File Key	4 levels; 12 characters, fixed	6 levels; up to 20 characters
Personnel Information	200 bytes per employee	Up to 8000 bytes per employee
Reports	Access to Employee Master File only with basic package; reports in Master File sequence only. With Payout option, access to all files is possible with extensive sequencing, formatting, and calculation capabilities.	Access to Employee Master File only; resequencing of standard and custom reports is possible.
System	65K IBM System/360 or 370 & others	128K IBM System/360 or 370
Price	$20,000 (plus $5,000 for Payout option)	$25,000

Payroll II
PHI Computer Services, Inc.

Payroll II represents an attractive package for current large-scale users of the PHI Generalized Payroll. For a reasonable additional investment ($13,000), these users can upgrade their payroll processing with minimum conversion problems. Current users won't be able to beat this price anywhere, and a conversion program is also included at no additional charge.

For non-users of the previous PHI payroll package, which is still available and actively marketed, the decision to use Payroll II must be examined more carefully.

PHI cites the adaptability of the system to growing needs by regeneration of the programs as a significant benefit of the system. Another point to consider is how other accounting functions are handled. While Payroll II is not part of a comprehensive system including a general ledger package, etc., a PHITAX system to compute locally imposed taxes and a fixed amount billing system are also provided by PHI. On the other hand, it is not really that difficult to strip out Payroll II information for input to another accounting system. (And Payroll II does provide extensive facilities for writing custom files on cards, tape, or disk with reformatting and resequencing of data records and fields.) However, this approach can lead to maintaining duplicate information in several files, a situation that has been decried—but widely practiced—by computer professionals for years. □

➤ Unlike the Generalized Payroll System, Payroll II is generated for each installation. Decisions required by the user include:

- Number and name of the user-specified earnings and deduction categories
- Sort keys for checks and reports.
- Number and name of year-to-date totals to be printed on check stubs.
- Character length of each level break for the master file.
- Amount and organization of information stored for each employee.

Once these decisions are made and the system implemented, changes require regeneration of the system.

PERFORMANCE: The performance of Payroll II is essentially the same as that of the Generalized Payroll package. However, most users will probably take advantage of options that will increase processing time. Using the check sequencing option, for example, involves a sort. The expanded personnel information option will mean additional reports and thus additional processing time. Additional deductions and earnings increase the size of the master file records and thus also add to running times.

HARDWARE/SOFTWARE REQUIREMENTS: Payroll II runs on a large configuration, requiring 90K bytes on an IBM System/360 or 370. PHI is willing to negotiate conversion to other systems. Master file record lengths can be more than 10K bytes, based upon user options requested.

PRICING: A license costs $25,000 and provides maintenance for 12 months, including tax changes. Thereafter, maintenance costs $2,400 per year for each site maintained. Regeneration of the Payroll II system for changes in the specifications costs $5,000 provided the same computer operating system is involved. A liberal discount is provided to service bureaus installing the package in several different geographical locations.

INITIAL DELIVERY: May 1971

CURRENT USERS: 18 operational systems as of September 1972. ■

GRS (General Retrieval System)
Program Products Incorporated

MANAGEMENT SUMMARY

GRS was first marketed in November 1967, making it one of the earliest proprietary programs. Originally intended for use on a variety of systems, GRS was written in COBOL for transferability between different computers as well as for ease of maintenance and direct modification by users. The original GRS system had a number of inefficiencies, due largely to the use of the high-level COBOL language for data manipulation and file handling; but subsequent versions have cleared many of these up, especially for System/360 and 370 computers.

Consisting of three basic phases, GRS requires one version of phase I per installation and unique versions of phases II and III for each master file and output report to be kept on a library file by the user.

GRS-B was developed toward the end of 1970 as a major revision of GRS and completely recoded in IBM System/360 assembly language with some COBOL statements included in strategic locations to simplify the user interface. GRS-B runs on IBM System/360 or 370 computers only, at 5 to 15 times the speed of the earlier GRS while selling at the same price. Among the limiting features of GRS which have been removed with the release of GRS-B are the following:

- A unique compilation and library retention of GRS phases II and III for each file is no longer required, and
- The record qualification and data selection process for GRS-B is much more efficient, with only the required fields pulled onto an intermediate file for sorting and report production instead of whole records, as is the case with GRS.

However, certain other limitations of GRS remain in GRS-B. With the introduction of a new PPI information retrieval system called The Data Analyzer (also for IBM users only) in October 1971, the following features were added:

- The Data Analyzer (described in Report 70E-691-03) can produce any number of calculation result fields, while computations in GRS or GRS-B produce a maximum of only 5 result fields.
- The Data Analyzer presents data in a variety of easily selectable visual output formats including bargraphs, mailing label formats, etc., while the same presentations in GRS or GRS-B would require specific modification of the basic system. ▷

General Retrieval System (GRS) consists of a generator program that produces special search and report-writing programs for individual files. First marketed in 1967, it is usable on an unusually wide range of computer systems.

CHARACTERISTICS

SUPPLIER: Program Products Incorporated, 20 Old Turnpike Road, Nanuet, New York 10954. Telephone (914) 623-6868.

BASIC FUNCTION: GRS is an information retrieval system that can produce programs to generate management reports. Written in COBOL, GRS contains skeletal source-language coding, with specific parameters needed to specify data qualification criteria, etc., entered by the user. Parameters specifying up to 100 reports can be batched for entry at one time, but seldom are more than 15 or 20 reports called for during a given run.

OPERATION: GRS is divided into three phases plus a standard utility sort. These phases are GRS I, which reads and analyzes control statements and prevents "dirty" input from reaching the execution phase; GRS II, which sets up and runs user-specified tests and selects records from the master file that qualify under the user-specified search criteria; and GRS III, which produces the management report(s) with up to 25 fields per line and 5 subtotals and a grand total per report.

GRS I is a general program, only one version of which is required for the GRS user no matter how many input files or output reports are specified for use by this system. GRS II (select) and GRS III (print) are the results of unique compilations for each different file to be run, and multiple versions of these two programs can be cataloged in the user's system library. (In GRS-B, no unique compilation is required, and only one version of phases II and III must be maintained for each installation).

Phase II puts complete qualifying records onto an intermediate file. This file is sorted by a standard utility into a sequence specified by the user in control statements. A GRS "Generator" is also available to produce unique GRS II and III programs tailored to different masterfiles. This feature permits easy creation of GRS II and III programs for additional master files.

GRS can have up to 55 analysis statements associated with each report. AND and OR relationships can be tested, and standard arithmetic operators +, -, *, and /, as well as relational operators =, ≠, >, and <, can be used. A "range" statement is also available that specifies upper and lower boundaries which delimit values a field may have in order to qualify for selection. Fields can be compared to constants entered with the search statement, or to other fields within the same or different records. Up to 5 result fields per report can be produced to hold the answers to calculations.

The following options are available:

- Summary reports without detail lines can be printed. ►

GRS (General Retrieval System)
Program Products Incorporated

▷
- The external function library in The Data Analyzer makes the newer system easily changeable and expandable to accommodate parameters for additional macros or subroutines. Thus, the capabilities of the system can be extended with ease, while the same enhancements to GRS or GRS-B can be made only by altering code.

- Use of FORTRAN as the report program language in The Data Analyzer permits much more detailed manipulation of the master-file data if desired by sophisticated users.

Thus, for IBM or non-IBM users, GRS in the COBOL version remains available, while GRS-B (written in System/360 assembly language) and more recently The Data Analyzer (producing FORTRAN report programs) have been developed specifically for IBM users.

With most efforts by the independent software firms naturally devoted to software for which the biggest market exists – namely IBM systems – GRS stands as a time-tested product available for non-IBM computer users. About 30 percent of the GRS installations are on non-IBM equipment.

GRS was developed at Information Science Incorporated (ISci), today the major stockholder in PPI. In 1971, a group at ISci realized that software package proprietorship was inappropriate for ISci, primarily a consulting and service organization specializing in the development of human resource systems. This group bought the rights to market GRS, obtained additional backing, and formed PPI. It was at ISci that preliminary development, about a year's worth, went into PPI's Data Analyzer.

USER REACTION

One GRS user that Datapro contacted has used the package since 1969, initially on a Burroughs B 3500 (using about 60K bytes of main memory), and now on an IBM 370/145 under OS/VS1. The user reported no problems at all in transferring the package; all he had to do was change the device names. This user is a data center manager in a bank that provides service to about 50 fuel oil dealers. He reports that GRS has provided all required services and been a real money-maker for the data center because of the savings it yields in programmer time. Another saving is less direct: the data center updates master files in a separate run of a utility program, using cards produced as an option under GRS as detail file input.

A second user who has since moved up to The Data Analyzer reported that GRS consistently enabled his DP department to respond to management requests within 24 hours or less. No user interviewed had any bad words about GRS performance or anything except praise for the PPI support.

Although The Data Analyzer has largely supplanted GRS for System/360 and 370 users, there are those who could use it well and save money in the bargain. For non-IBM users, GRS represents a data retrieval and reporting system that bears looking into. □

➤
- **Sample detail lines can be printed at fixed user-specified intervals.**
- **The results of two reports can be merged into one.**
- **The input or output records can be limited by the user to a specified number.**
- **Literal data can be specified for direct insertion in the output report.**
- **A moderate amount of format control over the output is available, including changes in column spacing, etc.**

HARDWARE REQUIREMENTS: For IBM equipment, a System/360 or 370 with 65K bytes of memory under DOS, OS, or their virtual storage counterparts, with a printer, card reader/punch and either two disc drives or one disc drive and four tape units. Main memory resident requirement is about 40K to 50K bytes. Such a system allows about ten searches at a time, with additional searches requiring about 2100 bytes of memory each. GRS can also be used on the following non-IBM computers: Burroughs B 3500 and B 5500, Control Data 3000 and 6000 Series, Digital Equipment PDP-10, Honeywell Series 400, 600, and 6000, and UNIVAC Series 70 and 1100. Contact PPI for specific configuration requirements on non-IBM equipment. A conversion to the Burroughs B 6700 is currently being made.

PRICING: The first GRS for an installation sells for $10,000, with additional copies available for $5,000 each. Standard improvements are provided for three years following purchase at no additional charge including centralized maintenance support and documentation. (Nine enhanced releases of GRS have appeared to date.)

INITIAL INSTALLATION: November 1967.

NUMBER OF USERS: About 180 to date. ■

The Data Analyzer
Program Products Incorporated

MANAGEMENT SUMMARY

The Data Analyzer is an information retrieval and reporting system that makes generation of management reports a simple task, yet it ranks as one of the most flexible and powerful retrieval and reporting systems on the market today. Its flexibility and modes of reporting make it suitable for a wide range of applications. The system can be used on IBM System/360 and 370 computers under DOS, DOS/VS, OS, or OS/VS. Optional features include the capability to accept input from multiple files and interfaces to the most widely used System/360 and 370 data base systems, IBM's IMS (Report 70E-491-01) and Cincom's TOTAL (Report 70E-132-01).

A very simple specification sheet for use in defining reports, plus similarly straightforward Master File Definition Forms, make The Data Analyzer readily usable for management reports. The system is well suited for use both by experienced programmers with intimate knowledge of the master files and by nontechnical personnel. Extensive default options allow quick report generation following generally accepted rules of presentation, such as automatic column headings, line spacing, and line width. What's more, simple override measures allow the user to tailor the default parameters to his individual needs.

Many information presentation functions and data manipulation requirements that could be regarded as unusual or sophisticated can be fulfilled by The Data Analyzer's standard external functions. These include bar graph and table presentation, label printing, table analysis, counts, averages, standard deviations, table look-ups, range breaks, date retrieval, statistics, analysis of repeated fields, and preparation of preprinted forms (e.g., checks, W-2's, and audit confirmations). The user can also add his own routines to this library of functions. Application modules are also provided for auditors (e.g., aging, stratification, random selection, etc.). To ensure file security, users of The Data Analyzer cannot write to, update, or otherwise modify master files.

PPI officially announced The Data Analyzer in October 1971 after having installed the system at a number of key accounts among prestigious and sophisticated data processing users. The system also benefits from the experience PPI gained through the development and marketing of GRS (General Retrieval System), an earlier information retrieval package.

USER REACTION

Datapro interviewed seven users of The Data Analyzer. They rated the system as follows:

This flexible information retrieval and reporting system for DOS, OS, and VS System/360 and 370 computers provides management reports based on a variety of free-form languages. Its features include extended reporting functions, an optional Multi-File Capability, and IMS and TOTAL data base interfaces.

CHARACTERISTICS

SUPPLIER: Program Products Incorporated, 95 Chestnut Ridge Road, Montvale, New Jersey 07645. Telephone (201) 391-9800.

BASIC FUNCTION: The Data Analyzer is a high-level information retrieval and reporting system designed to produce reports based on information in data bases or files. The basic command language is completely free-form, with a minimum of syntax rules. User-defined reports can be requested that involve selection, sorting, computations, and printing. There are minimal limits on selection criteria, the number and field length of sort criteria, the number and complexity of computations, and the number of data elements printed.

OPERATION: The Data Analyzer consists of several subsystems written in IBM Assembler language. These subsystems retrieve, analyze, and present the data. In addition, there are support programs for building the File and Data Definition (FDD) Library and the External Function Library.

The first of the three processors is the *Extract Generator.* It performs the functions of reading and editing the search request, generating a core image program for data selection and extraction, and building the parameter records used in the report generation phase. This processor uses the standard logical operators (AND/OR), plus six relational operators (equal, not equal, greater than, less than, equal or greater than, and equal or less than) and a range operator specifying lower and upper boundaries for the value of a field, as well as parentheses for logical grouping.

The second, a *Macro Processor,* retrieves external functions from the Function Library and modifies them as necessary to comply with the user's request. The output of this phase is a work file of instructions, which is placed in the standard report program. Appropriate sort keys are appended to records in the work file to allow sorting by a standard utility sort program, which is automatically executed by The Data Analyzer. The Macro Processor permits an expansion of the repertoire of allowable facilities in the basic system. CALL statements can be included by the user in the computational section to specify parameters to be used with certain subroutines or macros. These macros are called "external functions," and a wide variety of them is available to users of The Data Analyzer—or the user can write any additional ones he wishes for storage in the library.

A third processor, the *Report Generator,* builds the report program. As the program is built, the output code from the Macro Processor is merged. The result is a source-language program which is automatically compiled and executed to generate the desired output. This program processes the sorted work file produced by the Extract Generator. Computations can be performed at logic points such as before or after processing of each individual detail record, at the beginning or end of the run, or at control break levels including start of page, subtotal or total lines, etc. Up to 10 levels of totals can be defined in a single report, with a

The Data Analyzer
Program Products Incorporated

	Excellent	Good	Fair	Poor	WA*
Overall satisfaction	4	3	0	0	3.6
Throughput/efficiency	3	4	0	0	3.4
Ease of installation	4	3	0	0	3.6
Documentation	2	3	1	1	2.9
Vendor support	4	3	0	0	3.6
Training	1	5	1	0	3.0

*Weighted Average on a scale of 4.0 for Excellent.

The users represented the following computer configurations: IBM 360/65–1, 370/145–3, 370/158–2, and 370/168–1. Their operating systems were: OS–3, OS/VS–3, and DOS/VS–1.

The system performed as advertised immediately for two of the users, and eventually for four. The seventh user was not in charge of the system at the time of the installation. "Eventually," for most of the users, meant within one week. During that time, they made minor modifications to the system. The exceptions to this statement were two early users who actively participated in the system's development.

One of these early users summarized his feeling about the system this way: "We have had the system for four years and just ordered DATATRAN; we are completely satisfied." (DATATRAN, with the Report Phase Procedural Statements, expands the reporting language by combining the statements into routines to be called with simplified language, e.g., "CALL, MACRO NAME" or "PROC, PROC NAME.")

The Data Analyzer, the users reported, is primarily used by nonprogrammers to prepare reports in such varied application areas as medical research, housing statistics, marketing, union negotiations (collecting history on seniority, rates, etc.), and equal opportunity reporting.

On the basis of its flexible capabilities and its satisfied users, The Data Analyzer deserves serious consideration from System/360 and 370 users with a requirement for management reports based upon information derived from one or more computerized files or a data base. □

maximum of 40 numeric fields summarized at each total level. Standard arithmetic operators are recognized, as are parentheses for logical grouping of arithmetic operators. More than 25 standard mathematical functions are provided, including square root, absolute value, and certain trigonometric and logarithmic functions. In addition, the user can insert any valid FORTRAN statement directly into the computation specifications. Default options for management reports have been programmed into The Data Analyzer, but these can easily be overridden with explicit Option statements.

The Data Analyzer operates as a batch program, with job control statements entered only at run time to specify such things as the media upon which the input, output, and work files are to be read/written, and whether or not the source or object versions of the report program(s) are to be retained for future use.

File and Data Definition statements are used so that The Data Analyzer can intelligently read the files. The statements can either be cataloged in The Data Analyzer library or submitted at run time with The Data Analyzer specifications. Files composed of certain kinds of variable-length records, as well as other nonstandard file organizations, can be read as "External Input" files. For handling these nonstandard files, custom support is provided at no additional charge to tailor the Extract Generator.

At run time, the normal mode of operation is to sequence completely through the report specifications by the Extract, Macro, and Report Program Generator phases, followed by a utility sort and a compilation of the resulting object program, which is then executed using the work file for production of the output report(s).

The Optional Multi-File Capability allows The Data Analyzer to select data from more than one input file, for reports based on information integrated from more than one source of data. Depending upon present logic tests that may be satisfied during processing, The Data Analyzer will attach or substitute other input files.

Other user options include NOTE and EJECT statements. The NOTE statement lets the user add comments and documentation notes to his Report Request Language, which are printed out with the Language Analysis. EJECT signals The Data Analyzer to change pages between Report Request Language Analyses when there is more than one report.

Further system enhancements include extended diagnostics and new messages. The system's diagnostics cover most known limits and errors, and where possible will note and correct user errors.

PPI has also developed comprehensive interfaces to both IBM's IMS and Cincom's TOTAL. The Data Analyzer provides an easy-to-use IMS interface which requires minimal knowledge of the actual data base structure. The interface automatically accepts standard Data Analyzer free-form statements from the user, interprets these statements, searches the IMS data base, and presents one "logical record" at a time back to The Data Analyzer. The logical record is made up of only those segments containing data fields necessary for the set of reports being run. In the TOTAL version there is a Data Base Interface Language (DBIL) which allows the user to define any chain of events in retrieving data allowed by the TOTAL system. This also allows the user to qualify the data at any level before proceeding to read additional data sets.

HARDWARE/SOFTWARE REQUIREMENTS: The Data Analyzer will run under DOS, DOS/VS, OS, or OS/VS on an IBM System/360 or 370 computer with a card reader/punch, a printer, and either two disk drives or one disk drive and two tape drives. Floating-point hardware is also required. Under DOS, about 40K bytes of main memory are required, and about 85K bytes are required for the comparable version under OS.

PRICING: The Data Analyzer can be purchased for $16,000 in its DOS (or DOS/VS) version or $18,000 in its OS (or OS/VS) version. The price includes documentation, installation, and five one-day basic training sessions. Out-of-pocket travel and living expenses must be paid by the user. Multiple-copy discounts and lease/purchase arrangements are available; the discounts range from 40 to 60 percent. Maintenance and standard enhancements are free during the first year and cost $750 per year thereafter, plus five percent of the purchase price of any optional features. The price of the Multi-File Capability option is $3,000. The TOTAL interface feature costs $4,000, and the IMS interface costs $5,000.

INITIAL DELIVERY: The basic package was delivered in October 1971. Multi-File Capability was delivered in mid-1972. The TOTAL interface was delivered in mid-1973, and the full IMS interface during the last quarter of 1973. An extended IMS interface, which provides much more flexibility in data base accessing and searching, is now available.

CURRENT USERS: There are currently 152 users of the system, about two-thirds of whom are using OS and one-third DOS. Approximately 40 of these users are operating in a VS environment. ■

Corporate Shareholder System
Programming Methods, Incorporated

MANAGEMENT SUMMARY

The PMI Corporate Shareholder System is a heavily used package that was previously marketed by Management Science of American (MSA) as the Corporate Trust System/Stockholder Accounting System. The basic system as marketed by PMI is essentially the same as when marketed by MSA. The change in marketing representatives occurred when MSA ran into financial difficulties and the originator of package, The Trust Company of Georgia, withdrew marketing and support rights from MSA. PMI acquired the marketing and service responsibilities for the System in July 1971 and has installed 10 systems as of March 1972.

The wide usage of this system is not truly reflected by the number of installations (62 to date) because many of the users are banks operating as transfer agents for numerous client corporations. Shareholder information maintenance and stock accounting is being performed for roughly 1200 companies. Approximately 30 million dividend checks are prepared by this system annually. Clearly, this software package has been well exercised.

The shareholder accounting application involves many functions and is a very sensitive area. The PMI Corporate Shareholder System is a straightforward and comprehensive implementation of this application. Few, if any, conceptual difficulties should be encountered in converting from existing operations. As implied, this system represents no breakthrough in the art of handling information. But, before derating it for this reason (in this enlightened era of on-line programming and sophisticated file structures), think for a moment about how many of your bread-and-butter applications programs are also "merely straightforward" implementations of existing, well-rehearsed business procedures.

The primary concern, aside from the legal implications of shareholder rights, that makes a shareholder accounting system difficult to automate is the number of different outputs desired. Everything from new stockholder mailings to dividend checks to tabulations of proxy voting is included. The actual computational processing is trivial in comparison with just plain keeping track of what has happened. The PMI system provides the capability for handling virtually all of the activities normally associated with stockholder accounting, including maintenance of billing information for service organizations performing this function for multiple clients.

PMI recommends the system for handling between 5,000 and 250,000 stockholders, which is a clever way of saying it can be used by almost any publicly held company. The ▷

The PMI Shareholder System provides a comprehensive set of facilities for performing shareholder and stock certificate accounting and generating analysis reports from this information. Highly modular, the system can be run on small to medium IBM System/360/370 configurations with 65K bytes or more of main memory.

CHARACTERISTICS

SUPPLIER: Programming Methods, Inc. (a subsidiary of GTE Information Services Inc.), 1301 Avenue of the Americas, New York, New York 10019. Telephone (212) 489-7200.

BASIC FUNCTION: To provide maintenance of shareholder account information, stock certificate accounting, and various analysis reports for management. The Shareholder System runs on a 65K IBM System/360 or System/370 under DOS or OS. The system is set up to handle multiple companies or multiple stock issues by a single company.

OPERATION: The Corporate Shareholder System consists of 35 programs arranged in 8 modules. Three subroutines are written in assembly language; otherwise, all coding is in COBOL. To support the system, four master files are maintained. All master files are on disk storage and are organized in indexed-sequential form. Each module is run independently, although output from one module serves as input to another module in some cases. The system is operated in a batch mode, invoking each module as the requirements for file updating or maintenance and report generation dictate.

Separation of the system into modules is along the lines of application functions as well as processing convenience. The eight modules are: Process (transaction input validation and master file updating), Report Request and Special Process (report generation and stripping of the master files for dividend and proxy information), Cash Dividend, Proxies, Stock Dividend/Split, Yearly Tax Reports, Accounts/Certificate Delete, and File Backup.

MASTER FILES: The four master files maintained are the Company/Issue Control, Account, Certificate, and Cross-Reference.

The Company` Control file is sequenced by company number. A different number is required for each class of stock included in the system. A total of 999 companies/stock classes can be accommodated. Maintained in this file are various totals such as authorized number of shares, shares outstanding, number of open balance accounts, number of zero balance accounts, total accounts, dividends paid by period, and year-to-data taxes withheld. Also included are various items such as name and address, corporate tax code, date company entered system, demand deposit account number, and dates of record for dividends and one other application. For service organizations, extensive billing data pertaining to the number of functions performed (dividend checks printed, new accounts entered, ►

largest current installation is a bank operating as transfer agent for multiple companies; a total of 400,000 stockholder accounts are contained in the master files for this installation.

Approximately 50,000 stockholder records and the related certificate records can be stored on a single IBM 2314 disk pack. Extensive use of multiple name and address entries for stockholders would decrease this capacity.

Two aspects of the PMI Shareholder System need some comment. One is its high modularity; the second is the division of master file information into multiple files.

High modularity permits the Shareholder System to be run with minimum main memory requirements (53K bytes under DOS and 80K bytes under OS). PMI feels this is justified because so much of the system is exercised infrequently; lumping it all together in one big package would be needless luxury. Probably the only portion of the package that would benefit from increased core would be the report generation module, and even there the benefit may be more theoretical than actual. The fact of the matter is that the package spends most of the execution time handling input/output operations rather than in computation.

Division of the master file information, stockholder account and certificate, into two files is based on a subtle difference in the processing requirements for this information. Certificates are highly volatile in comparison to stockholder accounts. The certificate records are much shorter than the stockholder records. By separating the information into two files a double benefit is realized; the size of the most active file is reduced, and the blocking for records used in many reports is effectively increased. Both of these points enhance processing efficiency.

Many auxiliary files are created in the system. Many are used as created, but in some cases, they tend to be saved for batch input with other entries. This demands operational discipline to see that the correct copies of the files are used at the right time.

It is difficult to separate the features of the PMI Corporate Shareholder System from the environment of the stockholder/certificate accounting application. In criticizing some aspects, you run the risk of criticizing the application itself. The approach of the package has been toward batch operations. For most companies, this approach is sufficient. PMI estimates the total processing time on an IBM System/360 with 2314 Disk Drives at only about two hours per week to handle 50,000 shareholders. The primary concern is the effectiveness of the accounting and handling of the multiplicity of functions required rather than processing efficiency. However, some

labels printed, etc.) can be included in this file. Billing rates along with monthly and quarterly accumulations of functional items can be maintained.

The Account file contains up to about 600 bytes of information on each stockholder represented in the system; typical record size is 200 bytes. Alphabetic data is compressed by eliminating blanks to conserve disk space. The file is sequenced by account number within company number. All of the expected information is in this file, including shares held as of two record dates, year-to-date dividends and taxes withheld, and date of last activity. A Transfer Journal listing all stock transfers is output from each transaction input cycle. The pages of this journal are numbered sequentially throughout an entire year. A field within the Account file record contains the page number showing the last activity for the shareholder. On the Transfer Journal listing, the previous last activity date is shown, thus forming an audit trail for all transactions involving each stockholder. Provision is made in the Account file records for storing two name and address sets: the legal one and the one to which dividends are mailed. A zip code is stored in a separate field to facilitate the sequencing of any output in this order.

The Certificate file provides records for stock certificate accounting. One record is maintained for each certificate issued. The file is sequenced by certificate number within account number within company number to fac-litate processing of master file updates. Each record is only 30 bytes long and contains number of shares (to three decimals) date of issue, date of surrender, reason for issue, co-agent codes for issue and surrender, and a stop order code.

To facilitate handling cases where the input information includes a certificate number but not an account number, a Cross-Reference file is provided. This is a file of short records, sequenced by certificate number within company number, which contain the corresponding account number, date of issue, and a stop order code.

A master file update operation requires at least two disk accesses and may frequently require three—not counting an index reference if you don't hold the cylinder index in core. With IBM's ISAM coding being what it is, the processing efficiency leaves a lot to be desired.

PROGRAM MODULES: Three modules—Process, Report, and Backup—form the nub of master file maintenance and analysis report generation. These modules will be the ones most frequently used because they are central in on-going operations. The other five modules—Delete, Cash Dividend, Stock Dividend, Tax Reports, and Proxies—will be used as needed on a less frequent basis.

The Process module processes all changes to the master file. Transaction input can be on punched cards or magnetic tape (from a key-to-tape recorder). In either case transaction records are transcribed to magnetic tape, referencing the Cross-Reference file where needed to pick up an Account number (typically for certificate surrender transactions). Transactions are sorted, balanced, edited/validated, and processed against the master files. If a batch is out of balance, the whole batch is rejected. Assuming the batch is in balance, all transactions that pass the editing steps are processed, and only the transactions in which errors have been found need to be resubmitted. If desired, the processing can be stopped after the sort and held for weekly batching. A merge of transaction batches and system outputs (dividends, stock

Corporate Shareholder System
Programming Methods, Inc.

▷ users are modifying the system for on-line data entry and inquiry.

When DATAPRO 70 interviewed users of the system, they seemed happy enough. One user had trouble maintaining sufficient discipline over input and backup procedures in the beginning. Another user with a particularly large processing system was busy "demodularizing" the package and commented that the COBOL procedures used were a little odd. Another user had no problems at all, but he was only using a small portion of the full facilities.

Overall, the system looks better the longer you look at it. The shareholder accounting problem is sufficiently different from the average file-oriented data processing activity to make the solutions used in PMI's system look strange at first.

In contemplating a conversion to PMI's system, keep in mind that the cost of the package itself may be small in relation to the other expenses incurred in the conversion. □

➤ dividends, company billing data, etc.) can be performed and input to the validation step.

The basic outputs from the Process module include edit results, a Transfer Journal (formal listing of stock certificate transactions), new stockholder listing, new stockholder mailing labels, stockholder cross-reference cards, and a listing of duplicate/missing certificate numbers. The editing listings and duplicate/missing certificate number report are always output; the others are optional. The duplicate/missing certificate number report provides a check that numbers have not been skipped or incorrectly assigned to newly issued certificates, which is accomplished in the transfer operation. Another optional report lists the transactions in the current input batch involving stockholders identified with the Special Activity Code in the Account Master file. This report allows convenient monitoring of the activities of any one particular group of stockholders.

The Report Request and Special Process module includes a specialized report generator that allows structured access to the data contained in the Account and Certificate master files. Essentially all of the information is available, but only in the report formats provided. Because the system is coded in COBOL, additional reports could be added without major effort. Alternatively, a commercial report generator could be used, though the separation of the information into multiple files might complicate this approach a bit.

The output reports provide a fairly broad range of analysis, such as listing stockholders by classification code, account number, residence code (state), or share range. The classification code is a three-digit code carried in the Account master file. Each of these reports can be limited to those shareholders holding over a certain amount of stock. Separate summary reports by classification code, share range, or geographical code (state) can be printed. Mailing labels or inserts (for window envelopes) can be produced with various sort patterns. In addition, files can be produced on magnetic tape for input into the Cash Dividend module or the Proxy module. Calculation of dividends is included. These outputs can be qualified by one of the two record dates contained in the Account master file.

The Cash Dividend module uses the file output from the Report module and simply prints dividend checks and a register. The check format can be selected from several standard options. The checks and register can be printed in account number or zip code order. Two files are output from this module: one is used to update the master files with dividends-paid information; the other can be used in an external check reconciliation system. Facilities are included to restart, renumber, or reissue checks, making corresponding changes to the check register and reconciliation file.

The Stock Dividend/Split module handles the disbursement of stock certificates arising due to stock dividends or stock splits. These two items are handled in an identical manner, differing only in the titles of the listing and notices and in the code assigned to records written to the Transaction input tape. Input is a series of parameters on cards identifying the action taken, the split/dividend ratio, and the market value as of the record date, which must be previously entered via a file change to the Company Control file. The Account file is accessed and the dividend/split is calculated. Results are separated into whole and fractional shares. The system can be set up to issue fractional shares, to produce buy/sell option notices, or to produce cash-in-lieu-of checks for affected shareholders. Stock certificates are produced directly by this module. Option returns are matched against a file produced by the module when first run and validated; whole share certificates are then issued for buy options and checks for sell options.

Each time the Stock Dividend/Split module is run, whether to figure dividends/splits or to resolve options, a tape is produced which serves as transaction input to the Process module for master file updating. The market value of the stock issued against a dividend is added to the dividends paid. For those electing whole-share-only approaches, the share corresponding to a fractional dividend/split remainder is identified as a dividend/split creating a slight imbalance in the total stock shares issued as stock dividends/splits; i.e., the fractional share purchased by the shareholder to make up a whole share is carried as a stock dividend/split as far as the total number of shares goes, but not as part of the dollar amount of dividends paid. Thus, to get the numbers for entry to the company's books, the details of each dividend/split would have to be summed separately to distinguish between capital surplus and additional paid-in capital. For a large company with many thousands of stockholders, the result could be sizable.

The Proxy module uses the file output from a Report module run along with descriptions of the various options input directly to the Proxy module to create a mailable file of printed proxies. The proxies are normally printed on prepunched cards or prenumbered MICR documents. A "proxies issued" file is maintained, and several such files can be merged together. When the proxy statements are returned, any no or abstain votes are punched or encoded; yes votes require no manual data entry. Returned proxies are accumulated on a file. At any time, this file can be run against the "proxies issued" file for a tabulation of the voting. The tabulation is presented in the form of a detailed listing by stockholder, showing the voting for each option. Up to 25 questions can be printed on the proxy and tabulated. A check is made to insure that shareholders do not overvote. A shareholder can split ➤

Corporate Shareholder System
Programming Methods, Inc.

➤ his vote and vote blocks of his holdings independently. Revotes are permitted by the system.

The Yearly Tax Reports module allows the preparation of 1099, 1042, and state tax reporting forms for individual stockholders, as well as a tape for satisfying the company's reporting requirements.

The Deletion module provides for the preparation of a detailed listing of each shareholder's account, the deletion of zero-balance accounts, surrendered certificates, and a stop order listing. When accounts or certificates are deleted, appropriate listings can be produced.

The Backup-Restart Audit module transfers the master files from disk to tape (for backup) or from tape to disk (for ISAM reorganization). In addition, an account-by-account audit of account share balance to certificate detail is performed. The system also computes the overall control figures, which include shares outstanding, year-to-date dividends, year-to-date taxes, and "hash" totals on certificate and account numbers.

OPTIONS: Several extra-cost options are available for the Corporate Shareholder System.

One option prints dividend checks two-up for the first three quarters and prints a check with the corresponding 1099 form for the fourth quarter. Obviously, this can reduce the processing time considerably for preparing dividends.

A tracer option permits balancing to be performed on smaller transaction groups than full batches. If you expect a lot of bad input, this feature allows tracing errors more easily.

For those companies that want to tie stock dividend options to a variable stock price, a module is available to accommodate them.

A sequential module is also available for outputting all added records on tape in proper order. When restoring the master file, these tapes are merged with the current dump file. This option is useful for companies using stock dividends or splits.

The Dividend Reinvestment Module provides an automated capability of allowing stockholders to reinvest their cash dividends into a participation plan.

PERFORMANCE: The performance of this package is difficult to judge. Dependence on the Indexed Sequential Access Method, the use of four disk-based master files, the use of many sorts, and the use of many auxiliary files means that the majority of time involved in running the package will be devoted to I/O overhead. PMI estimates that an installation with 50,000 stockholder account records could be handled on an IBM System/360 Model 40 with 2314 Disk Drives with about two hours of machine time per month.

HARDWARE REQUIREMENTS: The Corporate Shareholder System will run on an IBM System/360/370. A DOS configuration of at least 65K bytes is required; the system will run on any OS configuration with sufficient peripherals. Normally, at least three disk drives (2311, 2314, or equivalent) and two magnetic tape units (or equivalent) are required in addition to the operating system requirements. Because the system is written almost entirely in COBOL, it can be adapted to non-IBM systems. Installations have been made on Burroughs B 3500, RCA Spectra 70, and Honeywell 635 systems.

SOFTWARE REQUIREMENTS: The system is delivered in source-language form and requires compilation on the customer's system; this minimizes, or at least more easily identifies, difficulties due to operating system options and hardware configurations. The system is available in COBOL D for DOS, in COBOL E for OS, or in ANS COBOL. For the DOS 65K-byte minimum configuration, either a 12K or 14K Supervisor can be used, with disk cylinder indices brought into main memory to cut down on the chaining of CCW's,

PRICING: The system is offered on what amounts to a perpetual lease for a one-time charge of $25,000. (To be precise, additional charges are $1 per year after 15 years.) This price includes complete documentation (which has been favorably commented on by users), 10 man-days of on-site training and support, the source programs, program demonstration files, and a one-year warranty on all elements.

The price of the two-up dividend module and the tracer module is $2000 each. The variable-price stock dividend and dividend reinvestment modules cost $2500 each. The sequential module costs $500.

After the first year, a maintenance agreement is available for $1,000 per year.

INITIAL DELIVERY: The Corporate Shareholder System was first delivered to a commercial customer in 1969 by Management Sciences of America (MSA), then the marketing and service agent for the system's originator. PMI delivered the system to its first customer in mid-1971.

CURRENT USERS: There were 62 users of the Corporate Shareholder System as of March 1972. Ten of these were installed by PMI; the others were delivered before PMI took over the package. ■

Easytrieve
Ribek Corporation

MANAGEMENT SUMMARY

Easytrieve is an easy-to-use system that permits straightforward information retrieval and reporting operations from existing BSAM and ISAM files. The primary design objective of Easytrieve is to handle the most frequently encountered jobs in the easiest possible manner.

Indeed, many of the fancy "bells and whistles" found in higher-priced systems such as Mark IV, CULPRIT, ASI-ST, etc., have been sacrificed for the sake of simplicity and economy. Certainly the $6,000 one-time payment for Easytrieve is a rather modest charge considering that a variety of tasks, from mailing label production to standard report generation or basic file updating, can be done at costs and speeds that compare quite favorably to those obtained in-house by individually coded procedural-language programs in COBOL or PL/1.

Easytrieve has been referred to as a "super utility" by its users. Inasmuch as Easytrieve makes light work of routine (and some not-so-routine) tasks–especially those "busywork" retrieval and/or file manipulation jobs that nevertheless require so much coding effort–the above description is an apt one. Users contacted by the Datapro staff report that Easytrieve is used for numerous file-stripping tasks and has effectively replaced RPG coding for "quick-and-dirty" special reports–and even for some recurring routine applications. The free-form queries used for Easytrieve allow programmers to formulate requests at the keypunch, punch the simple Easytrieve commands, and run the system to produce output immediately.

In summary, the information retrieval requirements of many users of both small and large-scale computers could well be handled by this highly simplified and rather basic system. Although many of the features in Easytrieve are a bit primitive, this condition is adequately reflected where it counts–in the price tag. One user reported to Datapro that his installation saved more than the full cost of Easytrieve during the first month of usage, based directly upon reduced labor costs. □

Easytrieve is a highly simplified and rather basic information retrieval system with few frills and a modest price tag. Uses range from mailing-label production to general-purpose retrieval and basic file maintenance in small to medium-scale IBM System/360 or 370 installations.

CHARACTERISTICS

SUPPLIER: Ribek Corporation, 10425 Burnt Ember Drive, Silver Spring, Maryland 20903. Telephone (301) 445-2255. Easytrieve is also available through a number of distributors, including International Systems, Inc., 150 Allendale Road, King of Prussia, Pennsylvania 19406. Telephone (215) 265-1550.

BASIC FUNCTION: Easytrieve is an information retrieval and report generation system written in assembly language for IBM System/360 or 370 computers under DOS or OS, or for UNIVAC Series 70 systems under TDOS or DOS, that can be used to produce reports from up to two input files on any of a variety of output media, as well as new or updated master files.

OPERATION: Easytrieve accepts a series of card input queries and a "library" describing the card, tape, or disk file(s) to be processed. As the queries are read, an executable program is compiled that immediately begins processing. Following the "mounting" of the master file or files (a master file can be a fixed or variable length magnetic tape file with blocked or unblocked records, a BSAM or ISAM disk file, or an input stream of punched cards or card-image records), Easytrieve validates the user requests, matches the file(s) mounted against the Library descriptions, and then processes the files to produce the output report(s) or the new or updated master file(s).

Easytrieve operates as a batch-oriented system in either of two environments specified by the user: Single-File Input or Multi-File Input (2 files). Within these basic environments, Easytrieve can operate in either Single Output mode (for the Single-or Multi-File Input environment); or Multiple Output Mode A (for the Single-File Input environment only).

For Multi-File Input, the user can specify three basic edit criteria:

- Match, in which all primary file records and any specified matching secondary records are passed for detailed processing.
- Match Only, in which every matching pair of primary and secondary file records only are passed for further processing.
- Merge, in which all primary and all secondary records are passed for detailed processing.

With Single-File Input, all records are considered for detailed processing subject to user-specified selection criteria. The Single Output Mode produces a single report or file that covers a maximum of up to 50 user requests or "condition sets." Each condition set consists of one or more query statements and associated data manipulation or calculation instructions compiled by Easytrieve. Thus, the user can combine rather extensive report production sequences for processing as one job during a single pass of the master file(s).

►

Easytrieve
Ribek Corporation

Three submodes are available within the basic Single Job Mode, each of which also produce a single printed report or a single output file:

- Edit submode—all condition sets are examined and then EQUATE statements are executed before the record is actually written out to the printer(s) or to one or more tape/disk files. (This submode is assumed as a default if no mode card specification is provided by the user.) An EQUATE statement alters an input masterfile record according to the query statement logic and values either imbedded in the statement or found in the input transaction stream.
- Update submode—similar to the Edit submode except that all input records are passed to the output file, regardless of whether or not any condition sets were satisfied.
- Edit Multiple submode—writes to the output file(s) those records that satisfy one of a number of condition sets. In this submode, output records are produced immediately after a condition set is satisfied rather than after processing the record against all the condition sets (as in the Edit submode). Because of this processing flow, multiple output records can be created based upon a single input record. Frequent use of this submode is made for mailing-label production and test file generation.

The Multiple Job Mode (Mode A) allows more than one print command to be present in a string of up to six condition sets, and provides for up to six separate print jobs and/or output files to be generated one after another without operator intervention. SORT commands can be associated with each condition set. Following a single pass of the input file, one or more intermediate files will be created on tape containing subsets of the masterfile for those records meeting the selection criteria. These intermediate files can either go directly to the printer or be sorted prior to printing.

Easytrieve has a basic set of six relational operators (EQ, NQ, GR, LS, GQ, and LQ) that can be combined with IF, AND, and OR connectors to form logical Boolean expressions. A named master-file field can be matched against other named fields in the master file, compared to more than one literal value contained in the condition set, tested against an inclusive range of literal values expressed in the condition set, or examined by combinations of the above using the Boolean operators.

The query statements that make up Easytrieve's condition sets must be in card-image format and cannot be split between two cards, although statements on contiguous cards can be related with AND and/or OR. Multiple statements on the same card must be connected either by all OR's or all AND's. Fields with up to 256 positions can be searched tor blanks, with Easytrieve handling search values of up to 256 bytes for alphabetic and non-quantitative zoned numeric data, 15-byte quantitative zoned numeric fields, and 8-byte (15-digit) packed numeric fields.

For Equate statements, Easytrieve permits the use of the four basic arithmetic operators (+, -, x, and /). Any number of Equate statements can be used in succession following a Query statement, with the results of each Equate process available to the next Equate process. Values and work fields used arithmetically in Equate statements cannot exceed 15 characters in length, and all operations are performed in strict sequence from left to right. Parentheses are not provided to alter this processing flow, thus requiring the use of intermediate equates in some cases. Using this technique, complex computations with sophisticated hierarchies of processing can be accommodated, but generally with some awkwardness.

A number of other features are present in Easytrieve, including:

- Condition Set (Query) Modification commands, including (1) PERFORM, (2) SUCCESS (causing termination of testing with immediate output of the record if one of a sequence of tests is met), (3) FLUNK (used to screen input and reject those records meeting the "flunk" criteria), (4) STOP (to end the job immediately following first satisfaction of the selection criteria without printing or writing the qualifying record), and (5) STOP AFTER (same as STOP except that the qualifying record is processed).
- Imperative commands that specify operations to be performed on the data (including resequencing), and output type and form specification.
- SUPPRESS, to suppress the printing of the Easytrieve statements on the output so that the output consists of data only.
- SORT, for in-line utility sorting on up to five fields. Only one SORT command is allowed per job stream in Single Job Mode, while each condition set may include a SORT command in Multiple Job Mode.
- PRESORT (for Single Job Mode only), to sort the input file(s) prior to testing the condition sets.
- BREAK command, for basic subtotals and/or page ejection.
- CONTROLS statement, for more flexible subtotals, allowing up to four levels of subtotals plus final totals. A maximum of 13 fields can be subtotalled in a single report.
- COMPUTE command(s) following a CONTROLS statement, to calculate percentages and other arithmetic results at subtotal and final total time.
- WRITE command, to cause qualifying output records to be written on a tape or disk file in the same format in which they were input.
- LIST command, to produce a printed report containing user-specified data from selected records. Up to 50 fields can be specified for an Easytrieve print line. More than one line can be printed per record.

A special LABELS command is provided for from one-up to four-up label formats in Cheshire format. Easytrieve output can also be punched on cards with the special PUNCH Command option.

HARDWARE/SOFTWARE REQUIREMENTS: Easytrieve runs on an IBM System/360 or 370 under DOS or OS, using the standard IBM Sort utility if required, or on a UNIVAC Series 70 under TDOS or DOS. Under IBM's DOS, Easytrieve's resident memory requirement is about 28K bytes; under OS the requirement is about 40K to 45K bytes.

PRICING: Easytrieve is available only on a perpetual license basis for a one-time payment. The full Easytrieve system costs $6,000, and additional installations (at different geographic sites) cost $4,800 each. A subset of Easytrieve, capable of handling only 1 input master file, is available for a single-copy price of $4,800. Maintenance and updates are included with any Easytrieve system for 1 year at no additional charge. Maintenance in subsequent years costs $240 per year.

FIRST INSTALLATION: Preliminary version, 1967. Current version, February 1970.

CURRENT USERS: About 65 as of August 1972. ■

Inquiry and Reporting System (IRS) Sigma Data Computing Corporation

MANAGEMENT SUMMARY

IRS is an information retrieval and report writing system that can be helpful both for harried, overworked, computer staffs and for servicing information requests from non-systems personnel. Systems staff people use it in place of high-level languages (like COBOL) to produce same-day output. Users and programming departments use it for recurring and ad hoc report preparation, and also frequently incorporate it into major systems.

Using IRS, one can respond quickly to fast turnaround demands and unscheduled or one-time requests for information. Recurring report requests can also be handled. This can be done by systems personnel without sacrificing normal operating efficiency. Non-programming users can also use IRS provided they have some knowledge of the data and can grasp basic data processing techniques.

IRS was first installed in October 1969 at the Marriott Corporation in Washington, D.C. Sigma Data has installed IRS on IBM System/360 and 370 and UNIVAC Series 70 (Spectra) computers. Batch versions are available for DOS (32K practical partition size) and OS or TDOS (60K practical partition or region size). By changing the job control language (JCL), inquiries to IRS can be made under any of the three operating systems, as well as under VS.

In addition, an on-line, interactive, free-format version of IRS has been implemented on the System/360 and 370 under IBM's TSO and other data communications monitors. The terminal user writes parameters at the terminal, where they are edited and corrected interactively. The query can then be executed in remote batch fashion or saved for subsequent submission.

IRS provides users with the capability to perform these important file processing functions:

- Read multiple physical or indexed sequential input files; files can be read from several devices simultaneously, and records can be fixed- or variable-length.
- Retrieve user-specified records based upon comparisons that can be logically connected.
- Access character, packed decimal, or binary data by either start position or field name.
- Produce multiple outputs with one pass of the input file.
- Sort records, in ascending or descending sequence.
- Chain to data in auxiliary files, in either the sorting, selection, or output phase of operation.
- Extract data items from selected records.
- Calculate and/or modify data for use in the selection, sorting, or output operation.
- Generate tailored, user-specified reports, report images, and summaries on printer, cards, tape, or disk.
- Create and/or maintain files on cards, tape, or disk. ▷

IRS enables users, including nonprogrammers, to extract information from computer files and produce card, printer, tape, or disk output. It operates on IBM System/360 and 370 DOS and OS systems and on Univac Series 70 TDOS systems. An on-line, free-format, interactive version runs on a System/360 or 370 with TSO or Intercomm as the message processing facility.

CHARACTERISTICS

SUPPLIER: Sigma Data Computing Corporation, Suite 506, 4720 Montgomery Lane, Bethesda, Maryland 20014. Telephone (301) 657-4455.

BASIC FUNCTION: IRS allows programming of common output-oriented file processing problems in a batch mode on IBM System/360 and 370 systems (under DOS or OS) and on UNIVAC Series 70 systems (under TDOS), or in a remote batch mode with interactive query preparation on System/360 or 370 systems under TSO, Intercomm, or other communications monitors.

OPERATION: First, the problem is evaluated for output requirements with regard to the files to be accessed, characteristics of the records to be processed, and the format of the final output. Second, the user codes the IRS forms; there is a maximum of three of these: Record Retrieval and Sort, Output Specifications, and Computation and Data Transfer. Third, data on the IRS code forms is punched on cards to be combined with run control cards. Finally, the IRS program is processed.

Execution of IRS programs is in two phases, corresponding to the two IRS modules, select/sort and report generator. Also, the program provides diagnostic messages, error statistics, and housekeeping data. An IRS card that has an error will not halt the processing of a batch, but only of the related query. In the interactive IRS version, queries are edited on-line and then either saved for later use or submitted as remote batch tasks. In the batch version, the two main modules are made up of sub-modules which are called automatically by the operating system.

The select/sort module sequentially reads records from up to 36 primary files per inquiry. Each primary file can be chained to up to 10 auxiliary files. The chaining conventions are that the primary and auxiliary files must contain a common data element. If ISAM, the auxiliary files must be organized on the basis of the common element. If PSAM, the primary and auxiliary files must also be in the same sequence.

Records are read and processed in one of three modes: select (records selected according to user-supplied testing criteria and output in file sequence), sort (records sorted), or select/sort (records selected and all those selected sorted). Record selection can be on criteria supplied in a literal or literal string, or against fields in other records.

Tests can be for equal, less than, greater than, less than or equal to, greater than or equal to, or not equal. If more than one condition or one of several conditions must be satisfied before a record is selected, conditions can be related by logical AND, OR, or ELSE connectors.

Records selected are prefixed with an additional nine (for fixed-length) or five (for variable-length) bytes for query identification, plus a variable number of bytes for sort key data, and then passed to the sort program. Files can be physical or ISAM, and records can be fixed- or variable-length, blocked or unblocked. ➤

- Perform single subscripting (indexing) for repetitive compares.

Printed reports can be on any type of form (e.g., labels, checks, invoices, ledger cards, etc.). Files created can be combined or collated from selected data on tape, cards, or disk. Files can be maintained by adding complete new records, adding to existing records, altering existing records, or deleting records. In the interactive version, the program displays final edited queries.

The system is installed with copious help from Sigma Data. Designated personnel are trained, documentation is provided, and the system is installed. The vendor provides post-installation support and a performance warranty. Users' guides are provided at a three-day seminar conducted by Sigma Data, during which formal instruction and applied, "hands-on" training are given to the customer. Sigma Data was founded in October 1968, and also provides custom contract programming services.

USER REACTION

Datapro conducted telephone interviews with five users of IRS. They rated the system as follows:

	Excellent	Good	Fair	Poor	WA*
Overall satisfaction	4	0	1	0	3.6
Throughput/efficiency	3	1	1	0	3.4
Ease of installation	3	1	0	0	3.8
Documentation	0	3	1	1	2.4
Vendor support	4	0	1	0	3.6
Training	3	1	1	0	3.4

*Weighted Average on a scale of 4.0 for Excellent.

The users represented the following computer configurations: IBM 360/40–1, 360/65–2, and 370/158–2. Their operating systems were: DOS–1, and OS–4.

The system performed as advertised immediately for four of the users and eventually for the remaining one.

The users interviewed found the system to be flexible and capable of a wide variety of tasks—from simple reports to large tabulations. One user found IRS as efficient as a well-written Assembler-language program, but cautioned against using IRS as a file maintenance system, as it is "basically a retrieval and reporting system." Another user would like to see the program logic in the report generator rather than in the select phase of the system. All the users found the system easy for nonprogrammers to use.□

The sort program puts records in the query number order and, within each query, within the user-specified sort order. Up to 36 levels of sort totalling 252 characters of data can be used, with a mixture of ascending and descending sequences. Data to be sorted can be supplied from PSAM or ISAM files (as auxiliary files), from the primary file (base record), or from temporary data fields created through computation during the selection process. Final output of the select/sort module is a sequential output file to serve as input to the report generator.

The report generator module accepts the aforementioned output of select/sort or any other physical or ISAM fixed-length tape or disk file. Input is read sequentially. A new reformatted tape or disk file can be created from the input file. Although only one primary input file is read by the report generator module, it can chain to, and obtain additional data from, auxiliary files via sequential or random-access matching. The data thus gathered can be included in the output. A common data element must exist in both the primary and auxiliary file. If ISAM, the auxiliary file must be arranged on the basis of the common data element. If PSAM, both primary and auxiliary files must be ascending sequence by the common data element.

Arithmetic (add, subtract, multiply, and divide) and data transfer (move) operations can take place during record reading by the report generator. Also, calculations involving summary data generated by the system can be performed prior to the production of sums for final output.

REPORT OUTPUT: System options governing line spacing, headings and/or footings, line overprinting, page and/or line numbering, and splitting of records between pages allow use of nearly any standard or preprinted form that the printer can accommodate. Alternatively, the output can be punched.

Up to nine levels of control breaks can be used. Operations at control breaks include: data summarization (total, record count, cumulative totals, maximum value, minimum value, and average value in one or more fields), skip (one, two, or three lines, or to a new page), group indication (print or punch only the first record in each group), and tabulation (print, punch, or create a summary record only for each group).

Up to 99 print lines of 132 characters and/or 80-column punched cards of data can be produced for each record read. Data edits can be specified using edit masks or "easy edit" options. Default options are provided for output format.

ON-LINE IRS: IRS FREE FORM is the interactive on-line version of IRS. It was developed under contract to Manufacturer's Hanover Trust Company. It has been implemented to run on a System/360 or 370 under Intercomm or TSO, but can be modified to run under any TP monitor that uses standard linkage conventions in a matter of 30 days. The program in this version converts conversational input into IRS parameter cards at run time for use in a local or remote batch mode by IRS.

As the system was originally designed, the user needed a CRT terminal for FREE FORM, but that is no longer required. During query entering and editing, query statements are stored on a temporary disk data set. When a query is finalized, the user identifies it as a "save data set" for subsequent IRS use.

HARDWARE/SOFTWARE REQUIREMENTS: For batch IRS, an IBM System/360 or 370 running under DOS or OS, or a UNIVAC Series 70 running under TDOS is required. Minimum core requirement under DOS is 22K plus supervisor, but 32K is more realistic. Under OS or TDOS, a minimum 44K partition or region is required, but practical minimum core is about 60K. Also required are 3 cylinders of 2314-type disk storage plus disk space for work areas. For IRS FREE FORM, an appropriate terminal and a System/360 or 370 running under a communications monitor employing standard linkage conventions (e.g., TSO or Intercomm) is required.

PRICING: IRS can be purchased outright for $20,000, or on a one-year payout basis of $3,500 down plus $1,500 per month for 12 months (total of $21,500). Rental terms are $2,500 down plus $650 per month. The cost of the FREE FORM capability is an additional $10,000 for the first site and $4,500 for each additional customer's site. There is no charge for maintenance.

SUPPORT: Support includes a three-day seminar conducted by Sigma Data at the user's location, at which formal and applied training are supplied. Sigma Data presents users' guides to the customer at the seminar. The vendor fully warrants IRS and maintains a permanent headquarters staff for continuous technical assistance. Full documentation is provided, and installation and post-installation support are given.

INITIAL DELIVERY: October 1969.

CURRENT USERS: 54 as of February 1975.■

ADABAS
Software AG

MANAGEMENT SUMMARY

ADABAS (Adaptable DAta BAse System) is an interesting and powerful data base management system on the same order as IBM's IMS (Report 70E-491-01), Cincom's TOTAL (Report 70E-132-01), and MRI's System 2000 (Report 70E-652-01). Pronounced "aid-a-base", the German-built system recently made its U. S. debut following a hugely successful European marketing effort begun in 1970, where more than $1.5 million in ADABAS sales have been recorded to date. Of course, the high price tag ($120,000) has helped ADABAS achieve that impressive figure with relatively few sales; but even so, an installed customer base of 5 or 6 systems for such a complex package would be more than an adequate measure of product maturity – and ADABAS currently has several times that number.

In fact, while ADABAS ranks as a top-notch state-of-the-art data base management system, few of its features offer unique capability (although one or two certainly *are* unique). Rather, the system has been put together with such thoroughness and skill that its most impressive features are not directly evident to the end user and appear in the form of reduced system overheads. For example, ADABAS uses a data compression algorithm to load data into the data base. This simple technique is widely available on a stand-alone basis, but its incorporation into ADABAS as an integral function of the system typically results in about a 30 percent reduction in the total volume of the structured ADABAS data base as compared with the original size of the raw, unformatted data before being loaded. Typically, finished data bases in most such systems are larger than the input data used to create them, and the "expansion ratios" for such bases range from about 2:1 to as much as 8 or 10:1, depending upon the inherent complexity of the data relationships and the overhead imposed by the data base system itself.

One of the most exciting potential benefits of ADABAS is related directly to the efficiency of the system and its logic design flexibility. While most of the parameters that define the system's capability are generally in the same ball park as those of other contemporary data base management systems, ADABAS has been designed to accommodate truly huge application environments (up to 4.2 billion-record data bases or larger). For example, ADABAS is nominally able to handle 255 files (or data bases), but can be used for up to 65,000 files with a minor internal modification not requiring a design change. Each file can have a maximum of more than 16 million records, and each record description can have 500 types of data fields, of which 200 can be key or "descriptor" fields. In addition to ADABAS' logical ▷

This efficient West German data base management system boasts a high level of craftsmanship in its implementation. Functionally on a par with IMS, TOTAL, and System 2000, the tightly designed ADABAS system includes all of the most desirable and proven state-of-the-art data base features at a high but well-justified price.

CHARACTERISTICS

SUPPLIER: Software AG, 12124 Basset Lane, Reston Virginia 22091; telephone (703) 620-9498. European (home) office; 61 Darmstadt, West Germany, Hilpertstrasse 20; telephone 61-51-82747.

BASIC FUNCTION: ADABAS consists of a data base manager and a number of utility programs used under DOS or OS with BDAM for generating and accessing a data base with automatic cross-referencing among data records. ADABAS provides a generalized file coupling network capability using a variety of high-efficiency data management techniques. A data communications interface to TSO is presently available. ADABAS operates mainly as a "host" language system, although a rather cryptic "self-contained" language capability is also provided.

OPERATION: ADABAS operation requires that the user perform the following steps (presented in the typical development sequence, although considerable overlapping normally takes place):

- The applications must be defined in terms of functional requirements and types of data to be processed. Individual applications programs must then be written in COBOL, PL/1, FORTRAN, or Assembler language to perform these applications, with calls inserted at appropriate points for the ADABAS access commands. Existing sequential processing programs using the data must be recompiled after some modification (consisting mostly of changing the Reads and Writes to ADABAS calls).

- All of the individual data requirements for each application should be coordinated into an overall data base requirement. This task is most appropriately handled by an individual serving as the Data Base Administrator. Note that this function is much more nearly administrative in nature for ADABAS than for systems such as IMS, where the "administrator" duties include rather sophisticated data base design responsibilities as well.

- Communications and other on-line retrieval requirements must be evaluated, both in terms of functional (application) needs and in terms of the network design criteria (terminal types and quantities) that may be present. If communications or interactive query requirements are present, a choice must be made of data communications monitors; an interface has been developed for TSO, and plans have been announced to interface ADABAS with CICS (Report 70D-491-01), Intercomm (Report 70E-694-01), and Taskmaster (Report 70E-866-01). ➤

ADABAS
Software AG

"network" capability, data bases can be geographically distributed under control of up to 255 ADABAS systems. This capability is a few years ahead of practical application, but many of the present ADABAS users and prospects are studying this feature for possible future use in their distributed data base networks.

On the other hand, ADABAS has two functional restrictions at the present time that partially offset the system's otherwise outstanding capabilities: lack of a native data communications monitor for remote access, and lack of an easy-to-use interactive query capability. In the first case, ADABAS, as a "host" system, has a generalized "call" that provides the ability to establish a remote access network, even though not inherently part of the ADABAS system. At the present time, a TSO interface is available, and plans have been announced for interfaces to Intercomm (Report 70E-694-01), CICS (Report 70E-491-02), and Taskmaster (Report 70E-694-01). In the second case, ADABAS has an interactive facility, but one that employs a very cryptic command structure requiring the end user to learn a rigid set of language rules. ADABAS, however, was designed primarily as a host-language system, and the interactive facility was intended for use by systems programmers, for whom the brevity of the ADABAS commands would be quite convenient. What clearly is needed for most commercial users is a basic front-end interpreter for that interactive facility to accommodate a more nearly natural-language inquiry capability. At the present time, although plans have been announced to develop this facility before the end of 1973, ADABAS is rendered vastly more usable through imbedded (host-language) application program calls in a communications-oriented or batch mode of operation than for interactive use.

The basic file structure used by ADABAS is a partially inverted file that lets the user identify the key elements for retrieval, but is developed in a way that imposes the minimum amount of storage and processing overhead possible in any given application environment. Records in each file can be logically coupled to any number of records in up to 80 other files. These coupling relationships can be defined *after* initial loading of the file, and the father-son relationship can be defined *at the time* of inquiry, thus resulting in a previously unattainable after-the-fact "network" capability.

In its internal structure, ADABAS embodies many of the best state-of-the art data management concepts and is an outstanding contemporary example of what a comprehensive data base management system should be. Aside from a few special statements, ADABAS has five basic commands to load, modify, read, find, and delete records. A reasonably complete set of about 100 diagnostic messages is included in the basic system to facilitate use of ADABAS and aid in debugging.

- At this point the actual data base can be put together, using special-purpose off-line utility programs. The existing data base or raw data generally must be reformatted into an input stream acceptable to ADABAS.

The ADABAS data base is stored on disk using IBM's standard Basic Direct Access Method (BDAM) and consists of three main components: (1) the actual data stored in disk locations with a unique Internal Sequence Number (ISN) assigned to each record, (2) an "Associator" consisting mostly of pointers or tables of indices defining the logical relationships between data items in the data base, and (3) an Address Converter work file that couples the data base relationship contained in the Associator to the physical location of the records.

The data storage portion of ADABAS permits up to 4.2 billion records in as many as 255 logical files to be treated as an entity. Each file can contain a maximum of 16.8 million records, and each file can hold 500 distinct field names, 200 of which can be designated as descriptors. The maximum individual value length is 255 bytes. Each record can have fixed-length or variable-length fields containing alphanumeric, binary, fixed point, or decimal packed/unpacked data.

The heart of this data base structure is the Associator, which holds the logical and inverted file relationship of the data base and, through the Address Converter, provides an address conversion capability to assign/locate specific records to/from physical storage. These logical relationships are stored in tables of sorted ISN's that contain user-specified key fields or index elements known as "descriptors", which are used in retrieval requests to identify frequently keyed-upon fields. These descriptors are also used to provide the information for coupling files together. The Associator information is updated automatically, and this process is approximately linear. Each ISN is 3 bytes (24 bits) long and is used to reference up to 16,777,214 records in each file. As the data base is built (or modified), a directory or "histogram" is kept that records the range and frequency of occurrence of unique values assumed by each field name in the data base. This is created and updated as a normal part of the Associator. In effect, this is a powerful and convenient shorthand table of contents to the full data base, and an easy-to-use read instruction is available to access this information.

When a retrieval request is processed, the user-specified search criteria are matched against the appropriate inverted fields, and the most economical search strategy is chosen. The shortest list of qualifying ISN's for a given parameter is retrieved first, thus minimizing the number of disk accesses required to satisfy the request. Fewer than three accesses are ordinarily required by ADABAS to fetch a particular record, with typical business applications averaging about two accesses. There is a version of the FIND command that allows the output to be sorted on up to three keys; the ISN "hits" are sorted internally before accessing.

ADABAS is supported by about a dozen disk-resident ancillary subsystems. These include:

- **A fast Loader System to build the data base using complex input data edit criteria and a data compression technique. The data compression routines eliminate the storage of leading zeros for numeric fields, trailing blanks for alphanumeric fields, and imbedded**

ADABAS
Software AG

North American ADABAS prospects (who have examined ADABAS in-house for a $2500 demonstration fee) verify that the system fully lives up to the claims made for it by Software AG. A major strength of ADABAS appears in a multiple-simultaneous-access environment, where application programs in up to 20 partitions or regions obtain simultaneous use of ADABAS's fetched results. For such uses, a 1K-byte Cross Partition Module provided by Software AG is attached to each application program. Another major strength reported by persons experienced in ADABAS operations is that considerably less user expertise is required in the design of the ADABAS data base than would be required for other systems such as IBM's IMS. The reason for this is that the basic record and file structures in the data base can be altered after initial file definition without a full-scale data base reorganization such as is required by IMS for similar changes. This is a considerable benefit, as any experienced data base user knows from usually bitter and always costly experience.

What becomes clear as the ADABAS system is studied is that a truly remarkable implementation of many worthwhile data base concepts has been made in ADABAS. In fact, it seems that there are no poorly implemented aspects of ADABAS's basic design. The fundamental weakness of classic hierarchical, semi-hierarchical, or network file structures (such as that of ADABAS) stems from the fact that a trade-off must be made between efficiency of data organization for retrieval purposes and the complexity of creating and updating the data base. Through clever programming techniques, ADABAS has been able to minimize the data structure overhead and consequently reduce the complexity of modifying the data base, allowing efficient retrieval without undue update overhead.

In all, Software AG has put more than 40 man-years of development effort into ADABAS to date, and a considerable number of European users are finding to their satisfaction that the resulting system has considerable merit. U.S. acceptance so far has been slow, not only because of the high price tag (on a par, however, with the European charges), but mainly because of the not-invented-here syndrome and the rather long umbilical cord of support required from West Germany. While understandable, much of this concern can be eased by the assurances of the vendor that U.S. personnel are being trained in ADABAS internals to give full local support, and by the truly remarkable performance of this system.

If a medium-to-large-scale data base management system is one of your needs, then an in-house demonstration of ADABAS on your own data is well worth having. □

null fields. All of these capabilities exist in the core-resident non-overlaid system for on-line updating, but they are not as efficient as the off-line batch utilities. In ordinary business environments, ADABAS' data compression results in about a 50% reduction in the volume of stored data as compared with the total length of raw input data. (Indices and related information add an overhead that can restore the final data base size to about the full size of the unedited input data; typical final size, however, is about 70% that of the raw input data).

- A Coupler (Koppel) to logically connect each file to up to 80 other files. Individual records in one file may be coupled to any number of records in another file without a parent-child relationship having to be defined in advance.
- A Checkpoint/Restart (RESTARTM) program to resume ADABAS operations following an interruption, including resetting the data base and Associator to the most recent checkpoint for rerun. Up to 199 checkpoints can be written onto tape during a single ADABAS run. ADABAS can *automatically* (1) regenerate lost data (e.g., bad disk track), (2) rerun an application program from a previously defined checkpoint, and (3) recover from machine failures, virtually guaranteeing the integrity of the data base.
- A Diagnostic system (VERST) used to print out the communications area containing user-developed interface coding to ADABAS for debugging.
- Different on-line modules to permit interactive access to ADABAS with a simple command language (i.e., in a "self-contained" sense).

Two unique aspects of ADABAS include (1) a phoneticized retrieval capability (although phoneticization currently is based upon German pronunciation only), and (2) a data encryption or cypher program for ensuring data base security. Both of these functions are noteworthy, but, in their current form, they will appeal only to a subset of general-purpose data base users. On the other hand, ADABAS provides up to 15 levels of security for the program, user terminal, etc., to give a good degree of confidence in the integrity of data stored in ADABAS.

PERFORMANCE: Two separate aspects of ADABAS performance must be considered: (1) data base creation or modification, and (2) ADABAS operation for processing retrievals.

For data base creation, ADABAS has been timed to compress, invert, couple, and load from 100,000 to 350,000 records per CPU hour on a System/370 Model 155, and is claimed to be capable of loading up to 1 million records per CPU hour on a 370/165. In a specific instance, ADABAS's batch loading rate has been clocked at 140,000 records per hour on a 370/165 for fairly dense 1400-byte records containing 250 fields, of which 20 were designated as descriptors.

For retrieval, many parameters are involved; but as an example, a 360/50 processing a 1 million-record data base with 200 fields per record against a 5-element set of search criteria took 4 seconds to determine the number of qualifying records, retrieve their ISN's, and read the first record itself. The second and subsequent records were returned within the average access time of the secondary

ADABAS
Software AG

➤ storage device, as all of the qualifying ISN's reside in main memory after the initial ADABAS analysis.

HARDWARE/SOFTWARE REQUIREMENTS: ADABAS consists of a non-overlaid nucleus of some 20,000 instructions (about 75KB) plus about 80,000 additional instructions (approximately 350KB) for the not-normally-used utilities. The nucleus plus a nominal 35KB buffer area must be resident in main memory at all times, while the remaining ADABAS modules can be stored on disk. A re-entrant 8K front-end module in ADABAS is used to queue access requests, while access itself is handled in a single-thread or non-reentrant mode. ADABAS can be run in a 110KB partition or region under DOS or OS (or their VS counterparts) on an IBM System/360 or 370, or on an IBM-like Siemens 4004 or Univac Series 70 system under the Disk Operating System (PBS for Siemens). IBM 2311, 2314, 2305, or 3330-type disc devices must be used for the data, Associator, and work files, and one magnetic tape unit is required for data protection.

The size of the data base itself depends upon the amount of raw data and the degree to which it is compressable (i.e., number of null fields, etc.). Typically, the data can be compressed to about 50% of its raw input size for commercial applications. In addition to this, an Associator is built containing the data base structure relationships. Typically the Associator requires 3 to 4 bytes per descriptor occurrence. On the whole, the final data base size tends to vary from about 50% to 100% of the size of the raw input data.

PRICING: Software AG (North America) provides ADABAS on a permanent-license basis for $120,000 for the first CPU, including all normal training. The second CPU license for ADABAS in the same physical location costs $40,000. A minimum 6-month lease plan is available that provides for a one-time installation fee of $3,000 plus a training charge, followed by monthly payments of $4,500. Training charges are $240/day for on-site courses, and a total of 10 days is generally recommended over a period of two months. Sixty percent of the monthly charges on a leased system can be applied toward purchase during the first six months of installation. For either "purchase" or lease, maintenance is provided at no additional charge. Future releases (expected to be about one per year) which focus upon efficiency will be available at no additional charge.

A user-paid demonstration is also available for $2,500. Software AG will convert part of the user's data base to ADABAS, run one or two of the user's existing sequential processing programs against it at the user's own site, and demonstrate the use of ADABAS commands for retrievals from that file.

INITIAL INSTALLATION: Version 1 (Basic ADABAS) –March 1971; Version 2 (factor of two improvement over Version 1 due largely to improved buffer management) –September 1972; Version 3 (very fast update procedures and optimized retrieval strategy for individual records vs. sets of records)–announced for June 1973.

CURRENT USERS: Over two dozen; one is a U.S. service bureau, while the others are located in Europe. ■

GRASP
Software Design, Incorporated

MANAGEMENT SUMMARY

As one of the most successful supplements to DOS, GRASP is widely respected as a powerful spooling system that greatly extends the power of DOS and helps to increase system throughput. SDI maintains that an overall throughput improvement of 15 to 25 percent is typical, with many users able to achieve up to a 35 percent increase. This enhanced DOS operation may result in reduced expenditures for additional equipment or overtime charges, and may permit postponement of a transition from DOS to OS on the grounds that a more powerful operating system is required to handle a heavier workload. A total of six releases of two versions of GRASP since 1968, combined with the infused experience gained from more than 300 installations, has resulted in a well-running program product with many features designed for the convenience of the operator.

The concept of spooling (Simultaneous Peripheral Operations On-Line) is not new. Widely used on second-generation large-scale computers, spooling allows input streams of jobs and/or data to be transcribed from card readers or other low-speed input devices to higher-speed peripherals such as magnetic tapes, disks, etc. A similar process places output intended for low-speed devices onto high-speed magnetic peripherals, for later conversion to the final output medium. This input/output processing is done simultaneously with other computing tasks.

As originally released, the IBM System/360 Disk Operating System did not have a spooling capability. This oversight was taken care of by IBM when the field-developed POWER routine was belatedly added to DOS as a no-charge supplement. IBM has subsequently gone a level higher with the release of POWER II. The system throughput improvements provided by either version of POWER are substantial, but leave a wide margin for further improvement by independent packages such as GRASP, which SDI claims can increase system throughput by 10 to 15 percent more than the IBM Spooler can.

An unusually frank and informative users manual openly lists some dozen errors found and corrected in earlier versions of GRASP, with a form and procedure to follow in reporting any further problems found in the current software release. This approach tends to create confidence in SDI's ability and desire to maintain GRASP. The thoughtful, clear documentation also contains many helpful hints to users of GRASP, as well as tips on how the efficiency of DOS can be improved through effective system loading, blocking, etc.

Computer resource utilization statistics accumulated by GRASP can be used in a performance analysis by the system manager to help determine the efficiency of his installation's workload mix.

GRASP, the most widely used non-IBM spooling supplement for System/360 DOS, increases system throughput by making it possible to perform multiple media conversion operations simultaneously with up to three independent user programs.

CHARACTERISTICS

SUPPLIER: Software Design, Incorporated, 999 N. Sepulveda Boulevard, El Segundo, California 90245.

FUNCTION: As a supplement to IBM's Disk Operating System, GRASP runs by itself in one partition or transparently shares the Foreground 1 partition (GRASP II) to provide spooling support for most standard IBM peripherals, as well as for "virtual" or phantom devices which need not be physically attached to the system. Input job streams are processed by GRASP and may be placed in intermediate disk buffers prior to loading into main memory for execution. Output files begin direct media conversion to cards or paper, using intermediate disk and main memory buffering for that portion of the output which exceeds the on-line capacity of the card punch or line printer. Flexible job accounting routines, automatic scheduling capability, remote terminal communication capability, and retrieval of cataloged procedures from the system's source program library are also available in GRASP and GRASP II.

OPERATION: GRASP is initiated as a user job in the input stream through job control statements, and is available either in basic GRASP or the more advanced GRASP II version. Both versions are for DOS only. GRASP occupies one of the three DOS memory partitions (F1 or F2), leaving 2 active partitions for other user programs.

GRASP II shares Foreground 1 with an attached user program, and runs without interference to the operation of that program in what SDI refers to as "Foreground 0" (F0). The use of the F0 or "dummy partition" technique permits three user programs in addition to GRASP II to reside in memory at the same time in a multiprogramming environment. Other specific enhancements to GRASP II include the following:

- Dynamic Partition Balancing to reassign run priorities to the I/O-bound and CPU-bound tasks, giving highest CPU priority to the most I/O bound tasks.
- User Program Relocatability, permitting any program to be executed in any partition large enough to hold it.
- Automatic Partition-Selection (APS) for automatic scheduling based on program priority and availability of partitions and peripherals.
- Remote Terminal Support (RTS) of the IBM 2770, 2780, 360/20, or System/3 in EBCDIC or 6-bit Transcode.
- Simultaneous Tape/Printer Output, which permits writing of formatted, blocked printer output lines on magnetic tape concurrently with, or instead of, spooling to the disk buffers.
- Device address independence, allowing logical file assignment to device type rather than to a specific peripheral address.

GRASP
Software Design, Incorporated

GRASP is also available in a restricted subset version called Mini-GRASP to run on very small processors. The GRASP product line, consisting of mini-GRASP, GRASP, and GRASP II, plainly offers a potential increase in throughput for IBM DOS systems and deserves careful consideration by such installations. □

- Ability to insert cataloged procedures into the jobstream through the use of "PROC" cards. Procedures can be nested up to 5 deep, thus greatly extending the power and flexibility of GRASP II.

All of these GRASP II features are intended to allow large DOS installations to increase processor throughput in ways not generally required on smaller DOS Systems where basic GRASP may be used.

Primary features of GRASP, present in both versions, include the following:

- Early printer start, enabling output to be printed while the job creating the output is still running.
- Two-level output buffering to both disk and main memory, enabling the printer or card punch to be driven from main memory while additional output is buffered to a disk.
- No changes to DOS or previously set up JCL for input job streams.
- Multiple print file support, with more than one printer driven at the same time.
- Support of IBM 1400 emulation.
- Automatic restart.
- Ability to run attached tasks, including concurrent teleprocessing, in the GRASP partition (does not apply to GRASP II).
- Job accounting routines to itemize CPU, device, and channel usage under multiprogramming by job step within user job stream.
- Support of UCS for printer output.
- Multiple output copies.
- Ability to backspace the printer file for recovery.

The different versions of GRASP are tailored for each user's system, and are supplied in object form based on macro specifications prepared by the user. The "phantom" macro defines the support required for each device including device type; the "accnt" macro describes the type and level of detailed job accounting data to be accumulated, and the "grasp" macro sets up the general environment, including the memory requirement for buffer areas, console message detail level, etc.

A number of support programs are available for use with GRASP or GRASP II for lineup of forms, communications requirements, and a number of other housekeeping or utility-type functions associated with the spooling process. Further information about these support programs may be obtained from SDI sales representatives.

The operation of the GRASP systems allows the operator to enter all of the available job streams onto intermediate disk storage, from which they will be loaded into main memory for execution. Automatic or manual procedures may be used to schedule and control the execution of the jobs, with minimal operator interface required. Normal operator intervention is required as usual for each user program except for input/output, which is handled by the GRASP system. Because of the two-level output buffering and the efficient blocking used, SDI has found that GRASP disk spool files typically require only about 1/4 of the space needed to spool the entire job to disk.

PRICING: Prices are shown below. Maintenance is included for all systems at no additional charge, including documentation. A one-time installation charge of $100 covers operator training, setting the package up on the customer system, etc. Configuration alterations are performed at no charge for the first 90 days, after which a $50 charge is made. More than 25 technical support personnel based in 9 major metropolitan areas provide maintenance for GRASP.

	Monthly Rental		
	Month-to-Month*	2-Yr. Lease	Purchase
Mini-GRASP	$216	$180	$5,400
Basic GRASP	300	250	7,500
Procedure libraries	0	0	0
Each additional non-standard device	30	25	750
Simultaneous printer/tape output	20	17	500
Basic GRASP II	360	300	9,000
Job accounting	0	0	0
Procedure libraries	0	0	0
Each additional non-standard device	30	25	750
Simultaneous printer/tape output	20	17	500
Self-relocatability	60	50	1,500
Foreground 0 (F0)	30	25	750
Remote Terminal System (RTS) (per terminal)	100	83	2,500
Partition balancer	20	17	500
Attached sub-tasks	30	25	750

*Lease may be cancelled on 30 days' notice.

HARDWARE REQUIREMENTS: IBM System/360, Model 22 through 65, or System/370, Model 135 through 155, with Storage Protection, line printer, card reader/punch, operator console, and a minimum of 20 cylinders on a 2311 or 2319-type disk device. Card punch files typically require 2 cylinders each; card reader files 5 cylinders each; and a printer file 11 cylinders.

SOFTWARE REQUIREMENTS: A DOS multiprogramming supervisor with command chain retry. GRASP requires a minimum of 4K bytes of main storage (2.6K bytes for GRASP plus 1.4K bytes for buffering), with additional main memory buffer areas reserved in 2K increments. When running GRASP II in F0, the F1 partition must be set up to include the GRASP II memory requirement in addition to the user program requirement.

FIRST USE: GRASP was introduced in Europe in 1968. GRASP is currently in its third release (December 1970), and GRASP II is also in its third release (December 1971).

NUMBER OF USERS: More than 300 to date. ■

GRASP-Spooling
Software Design, Inc.

MANAGEMENT SUMMARY

GRASP, one of the best-selling software packages and a member of Datapro's 1973 Software Honor Roll, has been unbundled. Its famous and sometimes unique features are now available collectively or individually to IBM System/360 or 370 DOS users. This report covering spooling is followed by reports on the other GRASP facilities.

Software Design's GRASP is among the most widely sold software packages in the world and is probably the most frequently used. Introduced in 1968, this spooling enhancement to DOS is now believed to have at least 1000 users in North America and probably 400 more overseas.

In September 1973 SDI unbundled GRASP, so that it is now possible to install many of the DOS enhancement features that GRASP had acquired in its continuing development on a standalone basis. Any package in the GRASP line can be installed with or without the others on any IBM System/360 or 370 system that includes a multiprogramming supervisor and the storage protection feature, provided the GRASP II Basic Module is installed.

This report is about GRASP-Spooling and the features attendant with it. Spooling (Simultaneous Peripheral Operations On-Line) allows input streams of jobs and/or data to be transcribed from card readers or other low-speed devices to high-speed peripherals such as disks or magnetic tapes for subsequent input to the CPU, and conversely permits output from the CPU to be placed on a high-speed magnetic device for subsequent conversion to the final low-speed output medium such as printed forms or punched cards, all automatically.

That's a simplified look at spooling in a nutshell, and nearly everyone knows that it's a desirable addition to a computing system. As a matter of fact, IBM, belatedly and under heavy pressure from users, began to offer a DOS spooling supplement free to its users; this is POWER, the IBM DOS spooler.

So why pay for GRASP? Because use of POWER entails hardware costs and use charges that usually exceed the cost of GRASP, and because GRASP can provide functional features unavailable in POWER that can more than justify GRASP's cost all over again.

One user put it this way: "Twice a year my company's controller tells me that POWER is free, and twice a year I show him that POWER's core requirements alone would cost more than GRASP." Then the user goes happily back to his system, as he has for the last three years, not even bothering to explain to his controller that GRASP is also letting him get more work out of the system in less time than would ever be possible under POWER. The simple facts are that GRASP requires less than half as much main storage as POWER, can use as little as one-quarter as much disk space as POWER, and can increase system throughput by 10 to 15 percent more than POWER can.

▷

CHARACTERISTICS

SUPPLIER: Software Design, Inc., 880 Mitten Rd., Burlingame, California 94010. Telephone (415) 697-3660.

In the United States, SDI has offices in the vicinity of Atlanta, Boston, Chicago, Cleveland, Dallas, Detroit, Los Angeles, Miami, Minneapolis, New York, Philadelphia, Pittsburgh, San Francisco, and Washington, D.C.

SDI's European headquarters is Software Design, Ltd., 24A Chemin Ed-Sarasin, 1218 Grand Saconnex, Geneva, Switzerland; telephone 022 98 40 22. International offices are also located in London, Melbourne, and Milan.

BASIC FUNCTION: GRASP is a collection of enhancements to IBM's DOS operating system for the System/360 and 370 computers. GRASP-Spooling is an alternative to IBM's POWER that also offers functional extensions similar to those found in OS/HASP. Cataloged Procedures, a free GRASP-Spooling option, makes OS-type PROCLIBS available to users spooling readers. PCI Fetch, another free GRASP-Spooling optional feature, enables 360/50 and faster CPU's to find more time to do useful work by letting the channels work harder.

Symbolic Device Equates, a separately priced GRASP-Spooling optional feature, assigns symbolic I/O units that let jobs run in any partition without the user having to keep duplicate or triplicate JCL sets. FØ is another option; it is a fourth partition created exclusively for GRASP that lets the user enjoy spooling with three complete and storage-protected batch partitions. Tape Spooling, another optional feature, enables spooling to tape either along with or separately from disk spooling; it is useful as printer backup. Remote Terminal Support, a final GRASP-Spooling optional feature, extends the advantages of spooling to remote users, and also provides extended facilities that are either unavailable or available only through the acquisition of additional IBM hardware and software.

OPERATION: GRASP-Spooling features extreme ease of operation and flexibility. It is transparent to DOS; that is, DOS "thinks" it is working with unit-record I/O. There are no changes to operations or to JCL, except for additions to support extended features the user may desire. The DOS operator just pushes the INTERRUPT button to communicate with the system in the normal manner, and the clearer GRASP messages appear on the console instead. (If you wish, as an advertising gimmick, SDI will replace the INTERRUPT button with a GRASP button; it's free.) Commands to GRASP-Spooling can be English-like or abbreviated.

➤

GRASP-Spooling
Software Design, Inc.

Saving GRASP's whizbangs and technical doodads for the Characteristics section, let's continue to discuss the subject you're reading this to learn about: cost-effectiveness. We're presently left with two key questions: "What about the new GRASP pricing structure?" and "How does DOS/VS affect GRASP?"

Discussing the price of GRASP is a ticklish subject. GRASP has traditionally been the highest-priced of any alternative to POWER, but this factor can be easily offset by the extensive testing of the software to ensure bug-free performance that is undertaken routinely by SDI, and by the strength and quality of its technical support. SDI even has a diagnostic and computing center in its Burlingame, California, headquarters to which users can be connected. And SDI has a reputation for not only having solved the few GRASP problems that pop up, but also for having solved problems in such IBM software as DOS itself and BOMP (Bill of Materials Processor).

The unbundling of GRASP meant more than just availability of some GRASP features on a standalone basis; the price structure was also altered. The meaning of the new pricing is clear: small users will generally pay less for GRASP, and large users will generally pay more. One can offer only conjecture at the rationale behind the price adjustments, but here's a try: The higher prices to large users cause them to pay for a larger share of the quality SDI support, which is only fair since these are the users who are generally using SDI technical support most often—especially as they get into areas where SDI solves problems with IBM or others' software, and where the problems stem from sheer size itself.

Meanwhile, price reductions at the lower end relieve small users, who seldom if ever need technical support, from paying for it at the same percentage rate. Not only that, but it makes GRASP more attractive to new small-scale users who might otherwise have been lured by lower-priced GRASP competitors. Also, the reduction at the low end can enable GRASP to remain attractive to potential users of POWER under DOS/VS.

Datapro has learned some of SDI's plans for GRASP under DOS/VS. The version for DOS/VS is currently in the final stages of beta-test, and will soon enter production tests at selected user sites. SDI's reputation for bug-fee software derives from the extensive testing of its packages. Customer release can be expected in the first quarter of 1974.

VS/GRASP will have pricing similar to that of the present version, and will offer extended features, such as paging statistics, in its optional job accounting module. POWER, which still requires as much core as ever, runs only in a nonpageable partition in the current release under DOS/VS, but the GRASP partition will be pageable under DOS/VS. If POWER is made pageable at its present size, it

Since the time relationships between job processing and printer output are different with GRASP-Spooling, forms notification is provided at print time. GRASP-Spooling also provides forms alignment masks for program beginnings and at any points in mid-program requiring forms changes. The alignment masks can be used repetitively, simply by pressing START and STOP on the printer. It is also possible to backspace through the printer record on the spool; this can simplify restarts.

GRASP-Spooling, unlike POWER, is an asynchronous spooler. That is, printer output can begin whenever the first record is ready, instead of waiting until a job is entirely processed. SDI calls this "early printer start." This effect, which decreases turnaround time, is accomplished using a cyclic disk concept, where the disk is a buffer ahead of the printer and the records on disk are overlayed in a round-about queue fashion. The user determines the size of the disk buffer, which can be as little as 1/4 as much as POWER requires. (The IBM package needs an extent equal to the printer file size.) The GRASP-Spooler buffering is 2-level, to main storage and then to disk. Under GRASP-Spooling, printers usually run at rated speed. This is enhanced by automatic right-hand print line blank truncation.

GRASP-Spooling "pseudo devices" can save users the cost of readers, punches, and/or printers by spooling second and third partitions, for example. A pseudo-device is really just another disk extent with its own address, say Printer O1E.

GRASP-Spooling can handle multiple print files, whereas POWER cannot handle two printers in any one partition. A GRASP user can, for example, print invoices and accompanying labels simultaneously.

GRASP-Spooling has automatic warm restart. It asks the operator if any queues exist during restarts.

Tape spooling can be valuable as backup when a printer is down. The output, spooled from disk, goes to tape for later transcription when the printer is running again.

Multiple copies of output is an OS/HASP feature that GRASP-Spooling users can use; but they must then let the disk extents go to full-size.

Procedure Libraries permit JCL on disk to be invoked via one card. This is an OS feature.

IBM 1400 emulation jobs can be spooled; this is impossible with POWER.

The console typewriter can be spooled. This can be valuable as reader backup.

Class Queues allow jobs in each partition to be executed according to a priority scheme. And in a situation where, say, a local user and a remote user are submitting jobs to the same partition, the Class Queues not only can thus resolve conflicts through the assignment of differing priorities, but, if the same priorities are assigned, the jobs will alternate until a queue is exhausted.

Automatic Partition Selection sends queued jobs to the proper partitions.

GRASP-Spooling Remote Terminal Support can let an IBM 2780 (or equivalent) look to the system like a 2540 Card Read Punch. It lets the user enjoy RJE (remote job entry) without having either a 2780 reader or BTAM, both of which are required by POWER to handle RJE. The trans-

GRASP-Spooling
Software Design, Inc.

should be at a considerable performance disadvantage to GRASP, since the latter would operate with less paging activity.

But a last lingering look at GRASP's pricing under the new policy causes us to want to warn SDI that some of its largest users could be forced into OS by the increases. One user complained about his system (which he calls the largest of its type in the Northeast United States) having its costs on two CPU's going up from $800 and $1,575 per month, respectively, to $1,100 and $2,200 per month. Another went from $650 to $875 per month. The larger user said that he's aware of POWER's buried costs that exceed GRASP's price, but he also notes that, for the 370/145, the recent IBM prices on expanded memory capacities reduce the cost per byte of additional memory by about half, thus tempting him to move from GRASP/DOS to OS or OS/VS1.

USER REACTION

Datapro's first annual User Ratings of Priopriet ary Software report (70E-010-40) showed that the 15 responding GRASP users gave the product extremely high ratings in all five of these critical areas: overall satisfaction, throughput/efficiency, ease of installation, documentation, and vendor technical support. Users interviewed since the recent unbundling and price restructuring still have nothing but praise for the product and its support from the vendor, but those who are now paying more are understandably disturbed. (We note that they're staying with the product, though.)

The GRASP DOS enhancements, led by GRASP-Spooling, have always demonstrated the ability to generate dollar savings through their use. Every multiprogramming DOS shop should take a close look at them. □

► mission recorder function of the Remote Terminal Support feature can take CPU snapshots for SDI's systems engineers to decode. Point-to-point and multipoint networks are supported on leased or switched lines, and a remote operator can be provided to allow remote operation without local operator intervention.

PERFORMANCE: GRASP performance is virtually an industry standard. Without exception, users report decreased CPU utilization, I/O devices running at higher speeds, and virtually trouble-free performance. All of this comes with much less main storage and disk usage than would be required with POWER. By using the FØ partition, system throughput can be further increased by having available a partition that would normally be occupied by the spooler itself.

HARDWARE/SOFTWARE REQUIREMENTS: Any DOS IBM System/360 or 370 computer. (Currently, the 360/22 through 360/65, the 370/125, specially equipped, and the 370/135 through 370/158 either can or must run under DOS.) A multiprogramming supervisor and the CPU Storage Protection feature (standard on some models) are also required. The basic GRASP-Spooling module requiries about 6K bytes of memory, and each spooled device, real or pseudo, requires about 2K bytes more. For example, GRASP-Spooling would require about 24K to hold spooling of 3 devices in each of 3 partitions, plus FØ, and with Job Accounting, Relocatibility (Load Libraries), and Partition Balancing thrown in (see Reports 70E-760-12, 70E-760-13, and 70E-760-15).

PRICING: The following are prices for items associated with GRASP-Spooling. All items are available on monthly payment plans, either under a month-to-month agreement or (with a one-sixth discount) under a 2-year commitment. System maintenance and new releases are provided at no additional charge. Installation costs $100, and a new version after the initial 90 days costs $50.

Feature Number	Description	2-Year Plan	Monthly Plan
0100	GRASP Basic Module	$127	$152
0200	GRASP II Basic Module	192	230
8200	Spooling Interface	54	65
8220	Each Spooled Device	32	39
2380	Device Equates	14	17
2863	Compression	8	9
3350	Extended Reporting Feature	22	26
3630	FØ Partition	54	65
3770	GRASP-to-GRASP Communication	54	65
3860	Horizontal Tab	8	9
5950	OS Emulation Interface	29	35
8281	UCS Translation Feature	11	13
9233	3211 Printer Support	29	35
8240	Tape Spooling	18	22
7800	Remote Terminal Support	90	108
6851	Each Communication Port	90	108
5866	Multipoint Line Control	126	151

The following are no-charge features: Core-Resident Directories, Cataloged Procedures, PCI Fetch, and Selective Tape Lister Support.

INITIAL DELIVERY: GRASP was introduced in Europe in 1968. By December 1970 it was in its third release. GRASP II, currently in use at more than 95% of the GRASP sites, was in its third release as of December 1971. Unbundling of GRASP took place in September 1973. A version for DOS/VS is scheduled for release in the first quarter of 1974.

CURRENT USERS: More than 1400 to date, about 1000 in the U.S. and Canada and the rest overseas. (The foregoing numbers are estimates, since Software Design policy does not permit release of such figures nor comment upon them.) ■

GRASP-Job Accounting
Software Design, Inc.

MANAGEMENT SUMMARY

GRASP-Job Accounting is designed to assist IBM System/360 and 370 DOS users in the areas of cost allocation, cost control, and configuration analysis.

Since the September 1973 unbundling of SDI's GRASP product line, the comprehensive job accounting module available with GRASP-Spooling is also offered to System/360 and 370 DOS users as a standalone package. The package, as a part of GRASP, has enjoyed an outstanding user reputation for its value in the areas of documenting and reporting for purposes of cost allocation, cost control, and configuration analysis.

In the first quarter of 1974, a new version of GRASP-Job Accounting can be expected. It will have some of its features enhanced, and will probably offer paging statistics and accounting techniques adapted for use with IBM's DOS/VS.

Cost allocations can be made on the basis of the analysis of machine resources used by each job that is presented in a System Utilization Report prepared by GRASP-Job Accounting. This data can be used as the basis for billing.

The same data in the report can bring to light problem areas in production environments that could be improved to increase efficiency. This is cost control.

To analyze a configuration, the data on channel loadings and CPU overlap presented by GRASP-Job Accounting can be used.

USER REACTION

At this writing, users of GRASP-Job Accounting on a standalone basis are unavailable for consultation. However, experienced GRASP users have in the past voiced approval and appreciation of GRASP's job accounting feature. Datapro sees no reason for this view to change. □

CHARACTERISTICS

SUPPLIER: Software Design, Inc., 880 Mitten Rd., Burlingame, California 94010. Telephone (415) 697-3660. See Report 70E-760-11 for other SDI locations.

BASIC FUNCTION: To assist users in improving the efficiency of DOS System/360 and 370 computers, control their operation costs, and equitably allocate the charges for the use of the systems. It can be used in spooling and non-spooling systems.

OPERATION: GRASP-Job Accounting creates a System Utilization Report that presents job names, dates, time-on, time-off, elapsed time, non-MPS duration, operator duration, operator efficiency, percent of CPU utilization, maximum core usage, number of forms used, remarks, and the busiest I/O unit. The non-MPS duration is the time the job would have run if it were running alone, i.e., the time attributable to its partition. Wait times of more than 2 seconds are noted and attributed to operator inefficiency.

Data about each job is captured as it is executed. Later, at the end of the job or job step (at the user's option), the data recording phase transfers the recorded information to disk. There, it is stored in binary format, in variable-length records blocked to 1128 for maximum economy of storage. One cylinder on an IBM 2311 disk can hold about 350 job accounting records. The analysis and reformat program, GRASPAC, can be used at regular intervals to print the recorded data and copy it to another tape or disk in a format suitable for subsequent high-level-language processing.

The resultant report can show, for example, the I/O activity for each logical unit within a job, with record counts for each spooled device and EXCP's (execute channel program commands, or supervisor calls) totalled by SYS numbers.

PERFORMANCE: According to user interviews, the GRASP-Job Accounting package performs as advertised.

HARDWARE/SOFTWARE REQUIREMENTS: A DOS System/360 or 370 computer with a multiprogramming supervisor and the GRASP II Basic Module. To use job step accounting, GRASP-Relocability is required (see Report 70E-760-13). Job Accounting requires about 4K bytes of storage.

PRICING: SDI pricing terms and the price for the GRASP II Basic Module appear in Report 70E-760-11. The basic accounting feature is offered at no additional charge.

Feature Number	Description	2-Year Plan	Monthly Plan
2322	Channel Usage Accounting	$7	$9
4324	I/O Devices Accounting	7	9
8310	System Units Accounting	NC	NC

INITIAL DELIVERY: Prior to July 1970.

CURRENT USERS: Exact figures are unavailable, due to the recent unbundling. However, a significant number of GRASP customers use Job Accounting. ■

GRASP-Relocatibility
Software Design, Inc.

MANAGEMENT SUMMARY

A major drawback of DOS multiprogramming is that the system is divided into partitions but no relocatibility function is provided. This means that a program can run only in one particular partition, and only at specific memory addresses. Thus, if a partition becomes available but the only programs that need to be run are cataloged for use in another partition, that "available" partition, for all practical purposes, does not even exist for the user. But the user is still paying for the memory and the I/O devices assigned to the unusable partition. He also pays for the time wasted and for the extra CPU time when he gets the partition he needs.

In an attempt to get some semblance of partition independence, some users catalog the same program more than once, under different names or in different libraries. But this practice is clumsy and inefficient, and it wastes disk space on a grand scale.

The answer is self-relocatibility, so that programs are no longer fixed as to where in main memory they must run. Self-relocatibility also permits partition core allocations to be changed at will, so that a partition's size can be adjusted to accommodate an occasional program that is larger than the partition's normal size. As long as the other concurrent programs are not thus forced to attempt to run in readjusted partitions too small to hold them, no problems will be encountered.

Finally, one and only one system disk pack will now be needed, supplemented as necessary by private load libraries. Changes need be applied only once, and a change in supervisor size does not mean a complete recataloging of every program.

USER REACTION

Until SDI's September 1973 unbundling move, GRASP-Relocability was an integral part of the GRASP spooling supplement to DOS. It has earned a reputation for efficient, bug-free operation, and can be expected to sustain that reputation both with other GRASP packages and as a standalone product. □

GRASP-Relocatibility gives IBM System/360 and 370 DOS users the powerful capability to make all their programs self-relocating. This eliminates wasted partitions and duplicate or triplicate catalogs.

CHARACTERISTICS

SUPPLIER: Software Design, Inc., 880 Mitten Rd., Burlingame, California 94010. Telephone (415) 697-3660. See Report 70E-760-11 for other SDI locations.

BASIC FUNCTION: Adds program self-relocatibility to spooling and nonspooling DOS System/360 and 370 computers.

OPERATION: GRASP-Relocatibility builds an in-core directory (in the partition occupied by the GRASP II Basic Module) of all modules to be fetched and/or loaded by a main program to speed phase loading. The greatly reduced fetch/load times can dramatically improve the performance of heavily overlayed programs, such as assemblers, sorts, and inquiry-type user programs.

I/O areas and other "DS-type" areas need not be stored on load libraries, resulting in a generally 10-20% smaller load library than the corresponding Core Image Library.

The system becomes independent from changes in supervisor size and partition boundaries. Programs can run in any partition, even though they are cataloged only once.

PERFORMANCE: Please refer to the "User Reaction" section.

HARDWARE/SOFTWARE REQUIREMENTS: A DOS System/360 or 370 computer with a multiprogramming supervisor and the GRASP II Basic Module. The PCI Fetch feature is advisable. About 6K bytes are required.

PRICING: SDI pricing terms and the price of the GRASP II Basic Module appear in Report 70E-760-11.

Feature Number	Description	2-Year Plan	Monthly Plan
7840	Relocatibility (Load Libraries)	$90	$108

INITIAL DELIVERY: March 1971.

CURRENT USERS: Exact figures are unavailable, due to the recent unbundling. However, a significant number of GRASP customers use Relocatibility. ■

GRASP-Resident Transients
Software Design, Inc.

MANAGEMENT SUMMARY

By placing DOS operating system transients in main memory, GRASP-Resident Transients can save commercial DOS System/360 or 370 users scores of processing hours per month.

The value of a resident transient area depends greatly upon the application and CPU speed, but mostly upon the availability of main storage. GRASP-Resident Transient Area Support, a unique program product, uses a lot of main storage for a DOS facility: normally about 12K bytes, and never less than 10K bytes.

However, its use can yield throughput performance improvements of up to 7% for sufficiently fast CPU's, provided that those CPU's have memory space available that would otherwise be unused.

DOS requires time to perform its own functions, and many of those functions must be called from disk. Try pushing REQUEST on the console typewriter of a DOS system and note the delay caused by fetching the operator transients before the response is received. Multiply this by the hundreds of thousands of such fetches per month in a normal commercial system, and it can add up to scores of hours.

On the other hand, the memory space must be available. If user partitions must be reduced in size to the point where some programs won't fit into them in order to get resident transient support on the system, then it is inadvisable to use the facility. It takes a lot of saved milliseconds to make up for a partition that's idle for an hour.

USER REACTION

GRASP-Resident Transients has been an optional feature of the highly regarded GRASP spooling systems, and is said by its users to be bug-free and paying dividends where it is intelligently used. □

CHARACTERISTICS

SUPPLIER: Software Design, Inc., 880 Mitten Rd., Burlingame, California 94010. Telephone (415) 697-3660. See Report 70E-760-11 for other SDI locations.

BASIC FUNCTION: Elimination of some system overhead in DOS System/360 and 370 computers by placing operating system transients normally on disk into an area of main memory.

OPERATION: GRASP-Resident Transients operates with complete user transparency.

Presently unused main storage, or storage saved by installing GRASP-Spooling to replace POWER, can be used to keep a stack of the most recently used transients and a stack of the most recently referenced directory entries. This reduces the number of supervisor functions resulting in calls to disk, and thus saves time.

PERFORMANCE: Performance gains usually do not appear unless the system is a 360/40, a 370/135, or larger. Performance will not be adequate if less than 10K bytes is allocated to the area for resident transient storage, and diminishing returns tend to result if more than about 18K is used.

HARDWARE/SOFTWARE REQUIREMENTS: A DOS System/360 or 370 computer with a multiprogramming supervisor and the GRASP II Basic Module. About 12K bytes of main storage is also needed to store the transient routines.

PRICING: SDI pricing policies and the price for the GRASP II Basic Module appear in Report 70E-760-11.

Feature Number	Description	2-Year Plan	Monthly Plan
7440	Resident Transients	$29	$35

INITIAL DELIVERY: June 1973.

CURRENT USERS: Exact figures are unavailable, due to the recent GRASP unbundling. ■

GRASP-Partition Balancing
Software Design, Inc.

MANAGEMENT SUMMARY

GRASP-Partition Balancing is a unique product available only from SDI. A key component of many GRASP spooling systems, it is now offered on a standalone basis. It can increase the throughput of a 2- or 3-partition DOS system by 10 to 15%, and greater increases have been reported.

SDI has the perfect analogy to the inefficient situation in which a CPU-bound job has a higher assigned priority than an I/O-bound job. They say it's like feeding a St. Bernard (CPU-bound) and a Chiuhahua from the same dish in the wrong order. The larger dog consumes all the food, leaving none for the little fellow. If the order were reversed, though, the samller dog would satisfy all his needs with hardly any noticeable effect on the larger animal's meal.

GRASP-Partition Balancing automatically monitors the CPU activity and constantly rearranges priorities to optimize the system throughput. The constant monitoring can adjust priorities of programs whose CPU needs vary during a run.

The user benefits from greater scheduling flexibility. If an important job is running in the lowest-priority DOS partition, normally the only way to ensure its prompt completion is to stop other partitions, sometimes cancelling active jobs. GRASP-Partition Balancing eliminates this need.

USER REACTION

Many GRASP-Spooling users have been using GRASP-Partition Balancing, and they report complete satisfaction based on throughput increases. Some users feel that it is a "must" for efficient operation of a DOS multiprogramming system. □

GRASP-Partition Balancing provides automatic monitoring, with operator override, of CPU activity on a DOS System/360 or 370 multiprogramming system. Its purpose is to rearrange priorities for optimum throughput.

CHARACTERISTICS

SUPPLIER: Software Design, Inc., 880 Mitten Rd., Burlingame, California 94010. Telephone (415) 697-3660. See Report 70E-760-11 for other SDI locations.

BASIC FUNCTION: Automatic CPU monitoring and program priority adjustment in a DOS multiprogramming system in order to increase overall throughput.

OPERATION: GRASP-Partition Balancing functions automatically, with operator override provisions. It continuously monitors the CPU activity of programs running in a multiprogramming system, lowering the priority of CPU-bound jobs and raising the priority of I/O-bound jobs.

PERFORMANCE: A typical 2- or 3-partition system will normally experience a 10 to 15% increase in overall throughput.

HARDWARE/SOFTWARE REQUIREMENTS: A DOS System/360 or 370 computer with a multiprogramming supervisor and the GRASP II Basic Module.

PRICING: SDI pricing policies and the price of the GRASP II Basic Module appear in Report 70E-760-11.

Feature Number	Description	2-Year Plan	Monthly Plan
6520	Partition Balancing	$47	$56

INITIAL DELIVERY: December 1971.

CURRENT USERS: Exact figures are unavailable, due to the recent unbundling. However, a significant number of GRASP customers use Partition Balancing. ■

GRASP-Dynamic Device Allocation Software Design, Inc.

MANAGEMENT SUMMARY

GRASP-Dynamic Device Allocation, used on a multiprogramming DOS System/360 or 370, can eliminate the need to acquire additional peripheral devices and/or reduce inefficient, clumsy operator requirements.

GRASP-Dynamic Device Allocation eliminates a major restriction of IBM's DOS: that devices be assigned to particular partitions. Thus, under DOS, once a job is running in partition F1, it cannot use the printer or tape unit it may need if that unit is assigned to partition F2 or BG.

The SDI product, unique in scope, not only removes this restriction, but it also provides automatic volume recognition (AVR). AVR permits premounting of tape and disk volumes, since the system is now equipped to locate and recognize the proper volumes automatically. The user can mount a tape reel or disk pack on any available drive, without regard for the drive address specified in the program.

GRASP-Dynamic Device Allocation nicely compliments GRASP-Relocatibility (see Report 70E-760-13) to provide true partition independence.

USER REACTION

GRASP-Dynamic Device Allocation and AVR were formerly optional features of the GRASP spooling systems. They have a solid performance and reliability reputation among users. □

CHARACTERISTICS

SUPPLIER: Software Design, Inc., 880 Mitten Rd., Burlingame, California 94010. Telephone (415) 697-3660. See Report 70E-760-11 for other SDI locations.

BASIC FUNCTION: Frees IBM's DOS for System/360 and 370 computers of the restriction that devices must be assigned to specific partitions. It also provides automatic tape and disk volume recognition, and makes it easier to use address-incompatible backup facilities.

OPERATION: Resource Information Blocks (RIB's) are loaded as part of the startup procedure, and tape and disk allocations are made from a table based on this information. Also, Device Allocation Blocks are maintained independently for each partition.

For Automatic Volume Recognition (AVR), a scan is made of all tapes or disk packs on all ready, unassigned units (at load point for tapes). If the appropriate volume ID is found, assignment is made to that unit. If the volume ID is not found by the scan, a device is assigned based on the RIB table information. Partition preferred and intervention units are exhausted first, and the unit that is finally allocated will be assigned and unallocated in all partitions. In the case of tape, the allocated unit will be able to handle the particular density of tape. Otherwise, an appropriate message is generated at the console.

PERFORMANCE: To achieve maximum benefit, a procedure approach should be followed, making one job out of many steps. This offers additional benefits where successive steps depend on the execution of earlier steps. If a step is cancelled, the remainder of the job is flushed. Throughput gains of varying amounts can be achieved, and devices can sometimes be eliminated.

HARDWARE/SOFTWARE REQUIREMENTS: A DOS System/360 or 370 computer with a multiprogramming supervisor and the GRASP II Basic Module.

PRICING: SDI pricing policies and the price of the GRASP II Basic Module appear in Report 70E-760-11.

Feature Number	Description	2-Year Plan	Monthly Plan
2400	Dynamic Device Allocation	$36	$43

INITIAL DELIVERY: January 1973.

CURRENT USERS: Exact figures are unavailable, due to the recent unbundling. However, a significant number of GRASP customers use Dynamic Device Allocation. ■

GRASP-Pseudo Clocks
Software Design, Inc.

MANAGEMENT SUMMARY

GRASP-Pseudo Clocks provides transparent software simulation of an interval timer (or two), enabling the user to have an independent timer for each task in the system. He can thus have concurrent time-dependent jobs operating in a DOS system, whether the system has spooling or not.

Pseudo Clocks is a unique SDI product. It can let a DOS user have two CICS partitions, for example.

USER REACTION

GRASP-Pseudo Clocks has been a feature available with GRASP spooling in the past. It has found use at a number of installations to permit simultaneous execution of two or more time-dependent programs. A common example of this is to use terminal polling routines that "wake up" to poll terminals from time to time while batched jobs run in other partitions. The program is said by its users to be well debugged. □

GRASP-Pseudo Clocks eliminates the IBM DOS restriction that limits the user to only one interval timer. Thus, it lets the user operate more than one concurrent time-dependent program.

CHARACTERISTICS

SUPPLIER: Software Design, Inc., 880 Mitten Rd., Burlingame, California 94010. Telephone (415) 697-3660. See Report 70E-760-11 for other SDI locations.

BASIC FUNCTION: Provides transparent software "pseudo clocks" in an IBM System/360 or 370 computer under DOS, thereby eliminating the normal restriction of one timer per system. This allows running of simultaneous time-dependent tasks.

OPERATION: Operation is transparent. For example, the timer facilities example given in IBM's DOS Systems Programmer's Guide (IBM form number GC24-5073), as "STXIT EXAMPLE" in Appendix C can be executed simultaneously in all three partitions with this program.

PERFORMANCE: Essentially identical to a situation in which two or three hardware timers existed.

HARDWARE/SOFTWARE REQUIREMENTS: A DOS System/360 or 370 computer with a multiprogramming supervisor and the GRASP II Basic Module.

PRICING: SDI pricing policies and the price of the GRASP II Basic Module appear in Report 70E-760-11.

Feature Number	Description	2-Year Plan	Monthly Plan
6370	Pseudo Clocks	$54	$65

INITIAL DELIVERY: June 1972.

CURRENT USERS: Exact figures are unavailable, due to the recent unbundling. ■

MMS General Ledger System
Software International Corporation

MANAGEMENT SUMMARY

The purpose of an accounting system is to accumulate data (normally costs) about a company's business operations in a manner that allows an accurate picture to be drawn for evaluating the effectiveness of past operations and providing a base to plan and predict future operations. This task can be thought of in two parts: one is collecting and accessing the information; the second is determining what grouping of information is required. The MMS General Ledger System is a way of implementing the first part; the second part is up to you.

Basically, the MMS General Ledger is a group of COBOL programs operating on disc-based files containing transactions, account balances, relationships among accounts, and report specifications. File structures and data accessing are based on the IBM Bill of Materials Processor, an applications program designed to facilitate maintaining multiple relationships among data items. The system has also been installed utilizing IBM's IMS version 2.

The handling of accounts is particularly flexible, allowing almost any imaginable level of detail to be maintained. Subsidiary accounts are automatically posted from detail transactions, and any account balance is accessible for reporting purposes. Relationships among accounts are maintained in a separate file. This flexibility allows cost centers to be identified strictly according to the organization chart, by groups across organizational boundaries, or in both ways.

MMS has also developed a flexible report writer. A limited number of report formats can be defined as far as column headings are concerned, but an unending variety of reports can be defined in terms of the accounts or groups of accounts to be included in each report. This flexibility permits designing reports which accurately reflect just the information each manager requires. An unfortunate tendency in automated ledger-keeping is supplying too much information in the management reports, detailing costs and comparisons with budgets. Inundated with too much information, some managers find it difficult to isolate those costs they can effectively do something about. Overly detailed record-keeping can also lead to an unfair evaluation of a particular cost center's performance by including costs over which the manager has no control. There are many schools of thought as to this, however. After all, somewhere sufficient profit has to be achieved to pay the president's salary. If you can solve the problem of just what your reporting needs are, chances are the MMS General Ledger System can satisfy them.

Other proprietary packages from SIC, including Accounts Payable and Accounts Receivable, can be tied into the General Ledger System without too much difficulty. SIC's marketing efforts have been principally aimed at locations east of the Mississippi, with areas outside the Northeast being handled by an agent network. There are a number of installations in the Middle West and West Coast, and expansion of direct marketing to Canada and Europe is now under way.

The MMS General Ledger System features flexible handling of accounts and report generation to facilitate using the data base created by storing a company's accounts on disc. It runs on IBM System/360 or 370 configurations as small as 32K bytes and two 2311 drives.

CHARACTERISTICS

SUPPLIER: Software International Corporation, a subsidiary of Manufacturing Management Sciences (MMS), Inc., 279 Cambridge Street, Burlington, Mass. 01803. Telephone (617) 272-2970.

BASIC FUNCTION: To automate posting of entries to general ledgers, and to generate management reports, in addition to conventional financial accounting reports, through a specialized, flexible report generation technique.

OPERATION: Seven master files are maintained on disc storage. All current account information is maintained in the Account Master file, along with two transaction history files: Monthly and Yearly. One master file, the Allocation Master, includes details for allocating indirect costs to accounts. Subsidiary relationships are defined in the Subsidiary Master; this file simplifies the automatic posting of entries upward in the chain of accounts. The other two master files deal with reports. One, the Report Master, identifies all reports that are to be produced, their frequency of production, and the distribution list. The other, the Report Line Master, details the report title, comments, selection and grouping of accounts to be included in each line, and the totals to be taken.

The MMS General Ledger System is written COBOL and includes a total of 37 programs. The structure of the files is based on the IBM Bill of Materials Processor or IBM's IMS (2)—both are fully supported for DOS but available only as Type 3 programs for OS. Under OS, IBM's CFMS—the OS version of DBOMP—is also fully supported. Figuratively speaking, each report is treated as an "assembly" consisting of multiple report lines, each of which may be a "subassembly" of one or more accounts. All account information is accessed randomly when pulling together a report.

The concept behind the system is to allow access to the data base represented by the company's ledgers for generating reports to aid managers in their decision-making processes. Naturally, maintenance of the ledgers is an integral part of this concept.

The normal processing flow accumulates transactions daily on the Monthly Transactions Master file; accounts are posted weekly. At the end of the month (accounting period), indirect costs are allocated, history files are updated, and the reports are prepared. Individual accounts can be examined at any time by running a series of inquiry programs; recent transactions are picked up from the Monthly Transactions file.

In general, almost any size of account numbers can be accommodated, including alphabetic characters. The Relationship Master file permits subsidiary accounts to be maintained to any level. For example, one customer uses a 22-digit account number, which identifies individual projects. The system imposes no restrictions upon the assignment of account numbers; any number can be an income or expense account.

The overall system consists of eight cycles: file creation, file maintenance, inquiry of Account and Allocation files, editing of Report Line and Allocation files, daily collection

MMS General Ledger System
Software International Corporation

▷ The MMS General Ledger System is real; it is installed and running at numerous locations, producing high-quality reports. DATAPRO 70 contacted a number of users, and in every case the package was living up to expectations and operating quite effectively. □

➤ of transactions, weekly posting of accounts, end-of-period processing, and end-of-year processing.

MASTER FILES: Account data is contained in the Account Master, Monthly Transaction Master, and Yearly Transaction Master files. The Account Master contains the account identifying information, fixed and variable budget information, current balances by month, and last-year balances by month. Through the Yearly Transaction Master, detail information can be retained as long as desired. The system is oriented toward using the information from the Account Master and Monthly Transaction Master to pull together reports. Period balances are available for previous years, and several users have up to 5 years of data on-line in the yearly Transaction Master(s). In general, the system is oriented toward having all data on-line at the same time.

The exact content of the Account Master is dependent on the installation. MMS prefers to keep record lengths the same and is willing to modify field names and lengths within this basic limitation; record lengths can be changed, but the modifications required to the system are more extensive than for field lengths.

INPUT: Input to the system consists either of transaction cards, file maintenance cards, and inquiry cards, or of tape or disk input produced by other accounting subsystems. Exact content of the transaction cards is dependent on the installation, but in general is minimal. Date, amount, credit/debit identification, and account are about the extent of the input data. Account descriptions are picked up from the Account Master file or from the Report Line Master file for reporting purposes. Transactions are normally batched each day. The credit/debit total is input on a separate card and compared with a computed total for each batch.

OUTPUT: The edit and listings transaction registers provide an adequate audit trail. These form the only standard reports from the system. All others are tailored for each installation.

Each report can be thought of as consisting of four parts: heading and comments, horizontal format, vertical format, and totals. The horizontal format is the arrangement of columns, such as budget dollars, actual dollars, previous years actual dollars, etc. The vertical format is the account or group of accounts that is reported for each line.

The horizontal formats are decided at installation time. Changes can be made later, but they will require some fiddling with the system. The system can accommodate up to nine different formats. Few users exceed the requirement for four or five formats.

For standard reports such as balance sheet, income and expense statement, etc., the vertical formats can be set up at installation time with little need for ever changing them. An almost limitless number of other reports can be created as the need arises.

Reports are created by using specialized forms to specify the requirements. Separate forms are provided to specify the report heading and comments, the accounts that make up each line of the report along with the stub entry for the line, and totals. Up to 99 different totals can be accumulated. By controlling when each accumulator is cleared and at what point in the report it prints, many levels of subtotals can be obtained. In addition, printing of the detail balances comprising a total can be suppressed to obtain working totals. The contents of accumulators can be combined by addition, subtraction, multiplication, and division to provide an unusually powerful reporting capability.

After the report specifications are generated, they are punched into cards and entered into the system. The specifications, including the frequency of preparation and distribution list, are stored in various files. Production of the reports is handled by a single program accessing the necessary files.

HARDWARE/SOFTWARE REQUIREMENTS: The MMS General Ledger System is customized by SIC for each user's installation, and requires a 32K IBM System/360 or 370 with at least two 2311 or 2314 Disk Drives under DOS. The resident main memory requirement is 24K bytes under DOS. The system also runs under OS/MVT or MFT in a partition or region of from 40K to 80K bytes.

PRICING: The system is available for lease under several options. A "single-payment" plan for a full DOS installation calls for a payment of $7,000 at contract time, $7,000 at master-file load time, and $7,500 when the system becomes fully operational, for a total of $21,500. After 5 years, continued use costs only $500 per year. This arrangement includes a 12-month guarantee against all program defects and up to 15 days of service for installation, education, and consulting (modifications).

The payment for a full DOS installation can also be spread over several years. In all cases, there are charges of $3,000 at contract time and $3,000 at master-file load time. Subsequent monthly payments are $450. The warranty against program defects holds for the entire period of the lease. Substantial discounts are offered for multiple installations within the same company.

A limited system for DOS is also available. This system provides for only the Account Master file but includes full report-generation capability. It is available in the full MMS array of leasing options. The "single-payment" cost is $10,000. Individual cost items are about 60 percent of those of the full DOS system. Ten man-days of installation assistance are included.

The MMS General Ledger system for an OS installation runs about a third higher than the full DOS system. The "single payment" charge is $26,500, payable in installments of $9,000, $9,000, and $8,500. For longer-term arrangements, the two initial payments are $4,000 each, and monthly payments are $550. Thirty-five man-days of installation assistance are included. The IMS (2) version is available for $40,000 under pricing arrangements similar to those of the OS version.

INITIAL DELIVERY: December 1969.

CURRENT USERS: 30 as of April 1972. ■

MMS Accounts Receivable System
Software International Corporation

MANAGEMENT SUMMARY

The MMS Accounts Receivable System is designed to fulfill the need for timely and accurate credit information. It is a base on which the user can build uniform credit, accounting, customer billing, and cash projection policies. This means, in a credit manager's terms, that the system monitors changing credit limits, average days outstanding, unearned discounts, unauthorized deductions, and past due accounts. The system automatically handles the repetitious accounting functions of receivable control and cash application, variable terms, repeating transactions, revolving charges, variable aging, and automatic chargeback.

The MMS Accounts Receivable System's major advantages in performing these tasks are a multicompany processing capability, provision for open item or balance forward methods, automatic discount and due date calculations, tracking of unauthorized deductions, and cash projection based on customer payment history. An On-Line Cash Application and Credit Inquiry feature is also available upon request.

The system's latest version represents advanced yet simplified file handling. To a user, this means flexibility in handling the requirements of different customers. A large number of customers with a few outstanding invoices each can be handled as easily as a few customers with a large number of outstanding invoices each. The system does this by maintaining separate files on disk for customers and for transactions and using a chaining technique to link related information. That is, a customer record contains a pointer to the first accounts receivable record, which in turn contains a pointer to the next one, etc. The entire system is based upon this COBOL file organizing and chaining technique, which allows for rapid data retrieval and exception reporting. The chaining technique also reduces the need for constant file reorganization.

USER REACTION

Datapro interviewed by telephone six users of the MMS Accounts Receivable System. They rated the system as follows:

	Excellent	Good	Fair	Poor	WA*
Overall satisfaction	2	4	0	0	3.3
Throughput/efficiency	1	4	1	0	3.0
Ease of installation	2	1	0	0	3.7
Documentation	0	5	1	0	2.8
Vendor support	5	1	0	0	3.8
Training	0	3	1	1	2.4

*Weighted Average on a scale of 4.0 for Excellent.

These users represented the following computer configurations: IBM 360/30–1, 360/40–1, 360/50–1, 360/65–1, and 370/135–1; UNIVAC Series 70/45–1. The operating

This comprehensive, data base-designed system uses advanced file handling techniques which allow the user control and flexibility in processing cash payments and controlling outstanding receivables. It will run on IBM System/360 or 370 configurations with at least 64K bytes of memory and two 2311, 2314, or 3330 type disks.

CHARACTERISTICS

SUPPLIER: Software International Corporation, an affiliate of Manufacturing Management Sciences (MMS), Inc., Elm Square, Andover, Massachusetts 01810. Telephone (617) 475-5040.

BASIC FUNCTION: To automate the keeping of accounts receivable records. The entire system is based on a unique COBOL file organization and chaining technique which allows retrieval of these records and exception reporting. A total of 51 programs written in ANS COBOL comprise the system.

OPERATION: The Accounts Receivable System is based on two major files chained together using the MMS COBOL Chaining Technique. These files are the Customer Master File and the Accounts Receivable File. The Customer Master File contains one master record for each customer. The record contains a disk address that provides linkage to all associated transactions on the Accounts Receivable File. The record may contain (besides customer number, name, and address) balance forward/open item indicator, D & B code, last activity date, and payment history. The Accounts Receivable Master File contains one record for each invoice, memo, adjustment, and cash transaction posted by the system. All transactions pertaining to a customer are linked to that customer's record in the Customer Master File, permitting an unlimited number of transactions for each customer. The records in this file will contain data fields such as: transaction type, terms code, "as of" date, authorized discount, amount due, and remittance data.

Two supplemental files, the Cross-Reference Master File and Accounts Receivable Transaction File, are also used. The Cross-Reference Master File contains one record for each record on the Accounts Receivable Master File; this record contains the transaction number, transaction type, customer number, and disk address of the corresponding accounts receivable record. The file is used to maintain invoice integrity and assist in the application of cash. The Accounts Receivable Transaction File is used to enter invoices, memos, and adjustment transactions into the accounts receivable system via tape or disk from various sources such as order entry and billing systems. Optional data fields are available in the records for information and control as determined by the user.

The MMS Accounts Receivable System has three basic cycles—a posting cycle, an "as required" cycle, and a period-end cycle.

In the edit phase of the posting cycle, due dates, discount dates, and discount amounts can be calculated based upon user-specified terms if they are not supplied by the originating source. All necessary editing is performed on each transaction to insure that only transactions containing

▷ systems were IBM DOS–4, IBM DOS/VS–1, and UNIVAC DOS–1.

The system performed as advertised immediately for four of the users, and eventually for one. The sixth user had just taken over the system and did not feel qualified to reply. Four users had made modifications to the system and two had asked the vendor to make them. In one case, the vendor even assisted the user in modifying the system to allow the addition of notes and leases. The process of modification during installation also acted as an excellent review of the users' accounts receivable procedures. One user found during installation that the system would not accept more than one invoice with the same number, and this was instrumental in uncovering invoices with this error in the user's file.

The vendor's support during installation and follow-up drew favorable comments from all the users contacted. They felt that the vendor performed in a highly professional manner. Among the selection criteria the users cited in choosing the MMS Accounts Receivable System were price, ease of use, prior experience with Software International's other products, and the vendor's continuing efforts in developing and improving the system.□

➤ valid data are posted by the system. For example, invoices or memos containing invalid transaction numbers, improper customer numbers, or invalid dates and amounts will cause the transaction to be rejected. Individual transactions or entire batches containing errors can be retained by the system on a suspense file, if desired, until corrective action is taken.

During the update phase, a record for each invoice, memo, and cash transaction (accepted by the edit) is added to the Accounts Receivable Master File. Also, the accounts receivable balance, sales, or remittance history is updated for the appropriate account on the Customer Master File. A corresponding entry is made to the Cross-Reference Master File for each record added to the Accounts Receivable Master File. The Cross-Reference Master File provides the edit phase with an efficient method of detecting duplicate invoice or memo numbers and supplying cash transactions with missing customer numbers when only the invoice number has been provided. Specific transactions or all transactions can be transferred between accounts during the cycle. Chargeback transactions can automatically be generated for short payments, overpayments, and service charges if desired.

Exception reports are generated during the report phase when unearned discounts, unauthorized deductions, service charges, and credit-limit-exceeded conditions are detected. The Accounts Receivable Register and Cash Register provide the user with a detailed audit trail of transactions posted by the system. Reports such as the Accounts Receivable Maintenance, Edit, Batch Control, and Summary of Suspense Items reflect transaction control and highlight conditions requiring corrective action.

During the "as required" cycle, reports can be produced for the entire Accounts Receivable File or can be run selectively through inquiry control reflecting the current status of each account. The reports are normally generated at the end of the period, but, as the cycle name implies, can be requested as required by the user. Detailed and Summary Aged Trial Balance Reports are produced with flexible aging capability through seven aging periods. If desired, a Past Due Aged Trial Balance and a Forward Aging of the Accounts Receivables can be printed. Reports such as the Unapplied Cash Listing, Unresolved Deduction Listing, Customer's Exceeding Credit Limit, Unauthorized Discount, and Accounts Receivable Open Item Listing will highlight the exception conditions. The cycle provides a Cash Application Document for posting cash and a Cash Forecast Report projecting how much cash can be expected to be received, and when. Statements and/or dunning letters can be issued to customers on an exception basis.

The system, at the end of the period, produces a report by control date and a report by department or division reflecting summary totals for each type of Accounts Receivable transaction processed for the period. Also, a summary of allowed deductions by type of deduction reported is produced. The main function of the cycle is to purge the Accounts Receivable Master File of records containing a zero balance amount due because of payment or adjustment entries. The Accounts Receivable Purge Report provides an audit trail of records removed by the system. All purged records are retained on a Detailed History File under user control and can be printed on the Accounts Receivable History Report if desired. In the final phase of the cycle, a Customer Status Report can be generated, and history data on the Customer Master File can be transferred to the Customer Summary History File as required.

HARDWARE/SOFTWARE REQUIREMENTS: The system is designed to run on an IBM System/360 or 370 computer with at least 64K bytes of main memory and two 2311, 2314, 3330, or equivalent disk drives. The system is completely written in ANS COBOL. The available versions are: IV-A (for DOS), IV-B (for OS), V-A (for DOS/TOTAL), V-B (for OS/TOTAL), and VI-A (for IMS-2). On-Line Cash Application and Inquiry is an optional feature for the DOS, OS, DOS/TOTAL, or OS/TOTAL version.

PRICING: The system is available on a four-year lease for the following single payments: System IV-A (DOS)–$16,500; IV-B (OS)–$19,500; V-A (DOS/TOTAL)–$19,500; V-B (OS/TOTAL–$23,000; VI-A (IMS-2)–$32,500. The On-Line Cash Application and Inquiry feature is priced at $5,000 under DOS or DOS/TOTAL and $6,000 under OS or OS/TOTAL.

At the end of the four-year lease, the system can continue to be leased for the following annual charges: System IV-A–$660/year; IV-B–$780/year; V-A–$780/year; V-B–$920/year; VI-A–$1,300/year. The above annual charges guarantee full maintenance against all program defects and all future enhancements for as long as the system is leased.

For multiple installations, the second system can be leased for four years at 50 percent of the base price. The third and subsequent systems can be leased for four years at 25 percent of the above annual rate. After the fourth year, each system can continue to be leased for the applicable charges outlined above.

INITIAL DELIVERY: September 1969.

CURRENT USERS: 35 as of February 1975. ■

MMS Accounts Payable System
Software International Corporation

This comprehensive, data base-designed system uses advanced file handling techniques to control and report on vendor payments, discounts, and cash commitments. It can be used on IBM System/360 or 370 computers with at least 64K bytes of memory and two 2311, 2314 or 3330 type disk drives.

MANAGEMENT SUMMARY

The MMS Accounts Payable System is designed to fulfill the need to control cash disbursements, future cash commitments, and the monitoring of who gets paid and when. It does this by maintaining audit trails throughout the system and by selective examination of commitments in advance of payment, by due date or by vendor. Augmenting these controls are summary information on the Vendor and Date Master Files and detailed histories of transactions, disbursements, and distributions. The entire system is based upon a special COBOL file organizing and chaining technique, which allows for rapid data retrieval and exception reporting. To the user, this means the flexibility to control cash reporting by either vendor or date.

The system can produce a tape with the records necessary for a complete check reconciliation procedure. A user may choose to have his bank do this processing or he may have the system do it. If, for example, the bank discontinues the service, all that is required is a simple file maintenance change to the system for it to assume the job.

Other features of the system generation of input to a general ledger system (such as MMS's own GL package—Report 70E-762-01) and the capability to incorporate independent financial controls, such as the insertion of balancing dollar amounts in the system's job stream. The company's marketing efforts place emphasis on its willingness to adapt such features to individual needs.

USER REACTION

Datapro interviewed five users of the MMS Accounts Payable System. They rated the system as follows:

	Excellent	Good	Fair	Poor	WA*
Overall satisfaction	2	3	0	0	3.4
Throughput/efficiency	1	2	2	0	2.8
Ease of installation	1	2	2	0	2.8
Documentation	2	1	2	0	3.0
Vendor support	3	2	0	0	3.6
Training	2	1	2	0	3.0

*Weighted Average on a scale of 4.0 for Excellent.

The users represented the following computer configurations: IBM 360/30–1, 370/145–1, 370/155–1, and 370/158–2. Their operating systems were: DOS–2, OS–2, and OS/VS–1.

The system performed as advertised immediately for two of the users and eventually for the other three; however, these three made extensive system modifications. All of the users interviewed commented on the vendor's willingness to tailor the system to their needs. Such needs ranged from accommodating their own general ledger systems to assisting in the installation of a user-written front-end processor, which was written to overlay financial controls (dollar amounts and record counts) into the system and balance these to control figures manually supplied by the accounting department.

CHARACTERISTICS

SUPPLIER: Software International Corporation, an affiliate of Manufacturing Management Sciences (MMS), Inc., Elm Square, Andover, Massachusetts 01810. Telephone (617) 475-5040.

BASIC FUNCTION: The automation and control of vendor payments, discounts, and cash commitments. The system provides complete prepayment information on vouchers by payment due date and/or by vendor prior to check writing.

OPERATION: Three principal master files are maintained on disk: who is to be paid—the Vendor File; what is to be paid—the Voucher File; and when—the Date File. A COBOL file organization technique associates these interactive data files. Any information referencing any part of the Accounts Payable System within any time period can be obtained simply by designating the time interval in the Date File.

The MMS Accounts Payable System is designed to handle multicompany, multibank processing. The system has five basic cycles: a daily input cycle, an "as required" cycle, a check processing cycle, an optional bank reconciliation cycle, and a period-ending cycle.

During the daily cycle, vouchers, credit memos, adjustments, and manual voucher transactions (checks written outside of the system) are validated and processed by the system. The transaction source can be cards, tape, or disk, depending upon the environment.

In the edit phase, all necessary validations are performed on each transaction to insure that only valid transactions are entered into the system. All batches containing rejected transactions are placed on a rotating suspense error file, where they are retained until corrective action is taken to relieve each error condition.

All accepted transactions are passed to an update phase which adds these transactions to the Voucher File. As the vouchers are added, the associated Vendor and Date Master records are updated to reflect their new status and balances. All accepted transactions are also matched against a Vendor Invoice file, generating a Duplicate Invoice Check report. This provides the ability to stop duplicate invoice payments.

During the "as required" cycle, various reports may be requested. These reports are the Vendor Master Listing, Vendor Labels (or Rolodex), and various cash requirement reports. File maintenance of the master files is also included in this cycle. File maintenance of dollar fields is not permitted.

During the check cycle, under the direction of user control, checks and remittances are written for those vouchers to be considered for payment. Numerous audit controls are displayed for control purposes. A Check Register containing manual checks entered since the last check run and checks produced during the current cycle is generated. These newly issued checks are now added to the outstanding check file for subsequent reconciliation.

During the optional bank reconciliation cycle, standard reconciliation procedures are followed to relieve cleared,

MMS Accounts Payable System
Software International Corporation

Ease of operation and reliability were stressed as the most important features of the system by another user. He cited the fact that he had been running it since May of 1974 and had experienced no program ABENDs.

One area—that of training for non-EDP personnel—came in for sharp criticism by four of the five users. They felt that they had to give the necessary training to their accounting departments, and if extensive modifications took place during implementation there were severe operational problems.

Overall, the users were enthusiastic about the MMS Accounts Payable System and the outstanding support given by the vendor. For anyone considering an accounts payable package, this is one that deserves your attention.□

canceled, or voided checks. If reconciliation is to be done by the user's bank, a tape is generated to be forwarded to the bank for its processing.

During the end-of-period cycle, strict accounting procedures are followed to insure complete and accurate recording of current liabilities and disbursements for balance sheet purposes. A report is produced to show the status of vendors on the current file. A purge of all closed vouchers and an update of a Year-To-Date Paid Voucher File is performed. A detail Distribution Report is produced, and journal entries are passed to a general ledger accounting system. The system provides for overlapping period processing, allowing for adjustment and closing preparation while a new period is starting.

MASTER FILES: The Vendor Master File contains one master record for each vendor in the system. Each record contains such data as: vendor's number, name, address (plus two optional addresses), and a contact name; 1099 indicator; standard industry code; activity date; A/P payments—year and month-to-date; A/P outstanding commitment; discounts taken—year and month-to-date; outstanding discounts; payments made last year; discounts taken last year; discount terms—days; discount terms—percent; cross-reference; bank; hold indicator, and temporary vendor. In addition, each Vendor File record is chained to the list of vouchers associated with that vendor.

The Date Master File contains the following summary information associated with particular due dates: total outstanding amount due, total outstanding discounts, number of open vouchers, and number of paid vouchers. Each Date File record is chained to the list of vouchers associated with that due date.

The Voucher Chain File is composed of records which contain all relevant information relating to a particular transaction with a particular vendor. In addition, each Voucher File record is chained to the list of vouchers associated with that voucher's vendor, as well as to the voucher list associated with that voucher's due date. Each record is also chained to the vendor and due date associated with that record. This extensive system of file chaining provides the user with great flexibility in obtaining information from the Accounts Payable System.

The remaining major files in the system are the Outstanding Check File, Year-to-Date Distribution History File, Year-to-Date Paid Voucher History File, and Vendor Invoice File.

INPUT: The major inputs to the system consist of: processing transaction input, voucher changes, check run request forms, and control cards. The system requires that all liabilities be recorded as soon as they are incurred. An Accounts Payable Voucher Apron is used for recording the voucher information. Different types of vouchers are used. For example, a manual check issued outside the system is handled through a transaction code 33, which will result in disbursement and distribution, but not in a check commitment. The user is provided with a series of these transaction codes and subcodes to use in conjunction with one or several preprinted input documents.

A check run is initiated by entering on a request form the accounting period to be paid, discount if passed, and all dates to be paid. The user has the ability to not pay vendors and/or divisions if he wishes to withhold payment. The system has a multibank capability and the ability to enter different starting check numbers for the different banks in a run. There are forms provided to initiate the requests for commitments, showing commitment in detail or summary by vendor and/or by period. Control cards are the last major source of input. These are used to trigger the runs that create the Vendor and Date Master Files or the reporting and processing runs.

OUTPUT: The output reports can be classified as belonging to the following types: entry edit listing, cash commitment status (checks and remittances), directory of vendors, end of period, and check reconciliation.

Input entries are checked for the presence of required fields and formatting conventions. Errors are shown on the Edit Report by means of asterisks,, and English-language messages are displayed. Control is maintained by batch through the matching of computed dollar amounts to submitted batch total amounts. Two reports, a Batch Suspense Report (listing all error batches placed on the Suspense File) and a Voucher Add Report (listing new commitments), complete the entry edit listing group.

A number of status reports can be printed on an "as required" basis. The system provides the following cash commitment status reports: Cash Summary by Date, Cash Commitment in Detail by Date, Cash Commitment in Detail by Vendor, Outstanding Check List, and Payment Made Month-to-Date by Vendor. Checks and remittances are printed, and a Check Register is produced. Part of this run includes an Exception Report identifying checks written which exceed a predetermined amount and checks voided because of vendor debit condition.

Programs are also included for producing vendor directories alphabetically by vendor name or numerically by vendor number. Output can be labels, Rolodex or a vendor book. The end-of-period processing (usually monthly) produces a Vendor History Report, a Paid Voucher Report, and a Distribution Report.

The check reconciliation run results in a journal of Card Input listing the check transaction types (1 = New Check, 2 = Cancel, 3 = Clear, and 4 = Stop), supplemented by a control count on items and dollar amounts. A Check Reconciliation report provides a complete reconciliation summary displaying each step along with the dollar amounts and item counts. Finally, an Outstanding Check Journal completes the check reconciliation outputs.

HARDWARE/SOFTWARE REQUIREMENTS: The system is designed to run on an IBM System/360 or 370 computer with at least 64K bytes of memory and two disk drives. Its 25 programs are written in ANS COBOL. The system is offered in separate versions for use under DOS, OS, IMS-2, DOS/TOTAL, or OS/TOTAL.

PRICING: The system is available on a four-year lease for the following single payments: System VI (DOS)—$14,500, VII (OS)—$18,500, VIII (IMS-2)—$30,000, IX (DOS/TOTAL)—$17,500, and X (OS/TOTAL)—$23,000.

At the end of the four-year lease, the user can continue to lease the system for the following annual charges: Sytem VI—$580/year, VII—$740/year, VIII—$1,200/year, IX—$700/year, and X—$920/year. The above annual rates guarantee full maintenance against all program defects and provision of all future enhancements for as long as the systems are leased.

For multiple installations, the second system can be leased for 4 years at 50 percent of the base price. The third and subsequent systems can be leased for 4 years at 25 percent of the base price. After the fourth year, each system can continued to be leased for the above annual rates.

INITIAL DELIVERY: June 1969.

CURRENT USERS: 43 as of April 1975, divided among 12 OS, 21 DOS, and 10 VS users. ■

SYMBUG
Standard Data Corporation

SYMBUG, an interactive COBOL debugging system for IBM System/370 computers running under VM/370 with CMS, is compilation-independent, eliminates the need for programmer knowledge of machine or hexadecimal language, provides write-protected file scanning, and can permit separate testing of main programs or subprograms.

MANAGEMENT SUMMARY

SYMBUG is an interactive COBOL debugging system for IBM System/370 computers running under VM/370 (Virtual Machine facility, see Report 70C-491-04) with the Conversational Monitor System (CMS, *ibid.*). It is a program available now that competes with IBM's COBOL Interactive Debugger, announced at the November 1973 GUIDE meeting for delivery February 15, 1974. SYMBUG was beta-tested in the summer of 1973 and already has 6 users.

SYMBUG obviates any need for programmer knowledge of machine language or hexadecimal code, since all references to COBOL data items are the same as those used in the COBOL program. It thus facilitates debugging at the source language level, which can dramatically reduce program debugging time.

SYMSCAN, a standard subsystem of SYMBUG, allows the COBOL programmer to scan any file accessible to his system, without allowing him to violate the file's integrity by changing it in any way.

An optional pair of SYMBUG subsystems, MAINTST and SUBTEST, respectively allow the COBOL programmer to independently test main programs without their called subprograms, and to test subprograms without having a main calling program.

No modifications to either the CMS nucleus or the COBOL compiler are necessary to install SYMBUG, and the package can be installed in one day. SYMBUG permits the COBOL programmer to interact with his COBOL program under any of three conditions: (1) whenever execution of the COBOL program results in a program interrupt, (2) at a pre-established breakpoint or upon encountering an execution interruption, and (3) when the user causes an external interrupt. At these times, the facilities of SYMBUG are available to the user through SYMBUG's command language.

In summary, SYMBUG allows the COBOL programmer to monitor and direct the flow of his COBOL program at execution time. This facility can be very valuable during the development of complex programming systems. The programmer using SYMBUG must be familiar only with CMS and SYMBUG's command language, and, of course, with the IBM ANSI COBOL or OS COBOL F compiler. The programs using the SYMBUG execution-time monitor must be compiled with the SYMBUG option.

USER REACTION

According to one user, SYMBUG is the best interactive COBOL debugger on the market. This expert user likes the ability to save commands on a disk file as indirect commands, the ability to loop through breakpoints, the

CHARACTERISTICS

SUPPLIER: Standard Data Corporation, 1540 Broadway, New York, New York 10036. Telephone (212) 586-3100.

BASIC FUNCTION: Interactive COBOL debugger and execution-time monitor for IBM ANSI COBOL and OS COBOL F programs on System/370 computer systems running under VM/370 (virtual machine facility) and using the CMS (Conversational Monitor System) facility. A standard provision also enables the COBOL programmer to scan any file normally accessible to his program, and optional subsystems (MAINTST and SUBTEST) permit, respectively, the ability to test a main program independently of its subprograms and to test subroutines without the main calling program.

Summarized, SYMBUG permits a COBOL programmer to monitor and direct the flow of his COBOL program during execution, and obviates any need on his part to know machine code or hexadecimal arithmetic; he need only know his compiler, CMS, and SYMBUG.

OPERATION: SYMBUG is installed, usually in one day, following the procedures in the installation manual. Programs using SYMBUG are compiled with the SYMBUG option. SYMBUG is activated via the CMS SYMBUG command or through a program interrupt. The CMS SYMBUG command will load the text files specified and enter the SYMBUG environment, after which the user can issue any SYMBUG command. Thus, the COBOL programmer can stop object-program execution from his terminal at any time and enter any of the SYMBUG commands.

The programmer can also interact with his COBOL program whenever its execution results in a program interrupt, at a pre-established breakpoint, or when an execution interruption is encountered.

SYMBUG commands can be grouped and executed as a group, and control can be passed to a group at any time by using the TO command. SYMBUG commands can use arithmetic and logical expressions as arguments, and these expressions follow normal COBOL conventions. Also, an expression can contain subexpressions (i.e., expressions enclosed in parentheses).

Breakpoints can be set at a NEXT executable verb so that the programmer can trace the branching logic in his COBOL program. Expert programmers can easily use the machine-language debugging environment (DEBUG), because SYMBUG has facilities to transfer to and from that environment.

With SYMBUG, the COBOL command is implemented via an EXEC file; to compile a COBOL program in the EXEC file, EXEC COBOL is used instead of COBOL. Meanwhile, specifying the SYMBUG compile option will create a file with the same filename as the COBOL source file and a filetype of SYMBUG on the user's A-disk (which was created at compilation time by specifying the SYMBUG compile option). The filename must be the same as the program-ID. The format of the SYMBUG command is thus: SYMBUG fn, fn2 . . .fn15.

SYMBUG
Standard Data Corporation

▷ HELP command (which explains the debugger in case the manual isn't handy), the ability to issue CMS commands during executions, the NEXT command, (which traces a program breakpoint after an IF statement to the next executable verb), and the ability to trace the calling sequence of a program.

SYMBUG also earned mention as the only debugger that lets a user resume program execution at any verb on a line of COBOL code. It also won praise for its SUBTEST feature (which has an analogue in only one other interactive debugger), a feature that is quite complicated in nature; and for SYMSCAN, which uniquely lets the user find the line number in a program so that he can set a breakpoint there. The EQUATE command was another favorite, since it allows the user to employ symbols in place of the long data names that COBOL programmers often tend to use.

Users who have had SYMBUG running state that the package is "clean" (well debugged) and that the vendor's record for responsiveness is very good. A user who took part in SYMBUG's development states that a design objective was to make the package as close to COBOL as possible. The ease with which COBOL programmers tend to learn the package's main features reveals that the objective has been met.

SYMBUG has also been adopted by sophisticated users who are experienced with CP-67 on the IBM 360/67 (the VM/370 predecessor). The package was chosen over IBM's announced product because it: (1) runs under VM/370 CMS now, and (2) gives users the facilities they've been asking for.

One user reports that SYMBUG was delivered in "above average" condition (this was the second user of the package), and that software fixes of the original bugs were "exceptional" and delivered within a matter of hours. This is because, the user says, the package is written to be easy to expand and/or maintain. Standard Data is currently working on a FORTRAN version of SYMBUG, and also on interfacing SYMBUG with TSO under OS/VS. □

➤ Issuing the SYMBUG command causes the files 1-15 (15 is the maximum, unless the TXTLIB feature of CMS is used) to be loaded and the SYMBUG environment to be entered. To execute the SYMBUG command from an EXEC file, the command EXEC SYMBUG is used.

To start testing with a different module, the COBTEST command is used, with the format COBTEST fn, where fn is the filename of the module to which subsequent SYMBUG commands will refer. The COBTEST command uses the SYMBUG file on the user's A-disk to automatically create a symbol table in main memory that is calculated to the correct size, depending on the size of the program. Once the symbol table for a module has been created, any SYMBUG command referring to that module can be issued, and the START command begins execution.

The SYMBUG environment accepts direct and indirect commands. The format of a direct command is any valid SYMBUG command followed by its associated arguments. The format of an indirect command is a line number (1 to 4 digits) that identifies the following command (in direct command format) as an indirect command; and the command can be any valid command except the ACCEPT command of SUBTEST. The indirect commands are stored in the order of their associated line numbers. Indirect commands can generate or reference other indirect commands.

SYMBUG variables can be Data Division data names, Index names, special registers, or SYMBUG internal variable counters. Data names used as variables can have up to three levels of qualifiers or subscripts, but a subscript expression cannot contain any subexpression.

Index names can be displayed or used as subscripts of data names. They must appear in the INDEXED BY clause for the data name being subscripted. Special registers accessible as data items in the COBOL programs are: RETURN-CODE, SORT-CORE-SIZE, SORT-FILE-SIZE, SORT-MODE-SIZE, SORT-RETURN, and TALLY.

There are 16 SYMBUG internal variable counters, and they can be used in any SYMBUG command. They are numeric data items with a PICTURE of S9 (16) and a usage of COMPUTATIONAL, and they contain an initial value of zero. Literals accepted in the SYMBUG environment can be numeric, alphanumeric, or hexadecimal.

There are 42 SYMBUG commands, not counting the two optional commands MAINTST and SUBTEST. They are straightforward and easy to learn, and all of the commands over two letters long have one- or two-letter abbreviations. Examples of the commands are: AT, which sets breakpoints where processing will be suspended and optionally executes other SYMBUG debugging commands; CMS, which executes CMS commands from the SYMBUG environment; FOR, which repetitively executes a SYMBUG request; HELP, which provides information about any SYMBUG debugging commands; IF, which evaluates a logical expression for a comparison and then conditionally executes a SYMBUG debugging command; SAVE, which saves the indirect commands in a spectified field; TRACE, which traces execution of paragraphs, lines, or specified verbs, as well as the program calling sequence; and WHERE, which translates hexadecimal addresses into paragraph names, line numbers, and verbs.

PERFORMANCE: According to users, SYMBUG is a well-debugged program that is functioning exactly as the vendor represents it. Users also state that it has, in general, very little overhead associated with its use; this means it doesn't cost very much to run a program with SYMBUG on, but without actually using it. However, the WHEN command, which allows the user to monitor a field for any change, should be used sparingly, because it can have quite a bit of overhead associated with it.

HARDWARE/SOFTWARE REQUIREMENTS: An IBM System/370 computer running under VM/370 with the CMS facility, plus either the IBM ANSI COBOL or OS COBOL F compiler.

PRICING: Under permanent license, SYMBUG costs $10,000 and the MAINTST and SUBTEST options together cost $3,500. A lease license for SYMBUG costs $326.79 per month, with the options costing $114.37 per month. Licenses include one day of training and first-year maintenance (error correction and all updates not charged as separate options). After the first year, maintenance costs $240 per year. Leases are cancellable on 9 months' written notice, and lease/purchase conversion credits are 100% during the first 3 months, 80% from 3 months to a year, and 50% thereafter. The vendor has not yet established rental prices, but states that rental will include maintenance and updates, be cancellable upon 30 days' notice, have no equity build-up, and involve separately priced training at $250 per day plus expenses. All plans include a 30-day free acceptance trial.

INITIAL DELIVERY: September 1973.

CURRENT USERS: 6 as of December 1, 1973. ■

PRO/TEST
Synergetics Corporation

MANAGEMENT SUMMARY

Somewhere, sometime, a data processing department must put its work to the acid test–does it work? To do this, the programmer must have data files to serve as the input upon which the program works, and a procedure for certifying that the output meets required specifications.

Two traditional approaches have arisen to create the test data files. One is to copy selected portions of a live file. The other is to create imaginary data records. Synergetics' PRO/TEST covers both bets. As to certifying program output, some degree of manual effort will inevitably be required to compare the test results with predicted or prior output. But that portion of the checking process that involves form as opposed to content–e.g., verification of formatting, editing, and field definition criteria–can be automated. It is this group of functions that is handled by PRO/TEST.

The name PRO/TEST actually refers to three different programs. The original Test Data Generator, introduced in the Fall of 1969, creates dummy test files entirely from input specifications. The more recent File Processor–first delivered early in 1971–is a file stripper with powerful capabilities for culling records from a master file or data base and/or modifying the selected records. The File Checker verifies the output format of a program being tested.

The Test Data Generator is available for IBM System/360 or 370 computers and also for the Honeywell Series 200, 2000, 600, and 6000 systems; about one-fourth of the current installations are using Honeywell equipment. The File Processor and File Checkers are available for the IBM System/360 or 370 only.

The comprehensive Test Data Generator enables construction of just about any hierarchical file structure imaginable. Field generation capabilities include relational and arithmetic operations along with generation of sequenced, range, and random values. The output file can be on magnetic tape, disk (sequential or indexed sequential), or punched cards.

The record selection facilities within the File Processor allow considerable latitude in selecting records, as well as in specifying the processing for each type of record. Different selection criteria can be intermixed in processing one file. Once the records have been selected, the data fields can be manipulated along with generated data from the Test Data Generator program to obtain the desired output. The File Processor opens the door for applications beyond the creation of test data, such as file conversions and selective file listings.

The File Checker accepts parameter cards that describe the output files to be produced by a particular program. That program's output is then fed into the File Checker, where it is matched against the original record design specifications and a verification of format is made. The File Checker is also suitable for use in a service bureau or ▷

PRO/TEST consists of two test data generator programs–one that creates data based upon input parameters and another that strips a live file to cull/modify existing records–and a validation program that checks the output formats of programs under development. All are available for IBM System/360 or 370 computers running under DOS, OS, or their VS counterparts. The basic Test Data Generator program is also available for most Honeywell computers.

CHARACTERISTICS

SUPPLIER: Synergetics Corporation (a subsidiary of Search, Inc., Hartford, Conn.), One Garfield Circle, Burlington, Massachusetts 01803. Telephone (617) 272-3450.

BASIC FUNCTION: To generate data files for testing programs and to validate the form (rather than the content) of the programs' output. The Test Data Generator works entirely without input files; the File Processor works from an input file and can also generate data, as well as handle many types of single-file manipulation applications. The File Checker validates the format of the output files produced by the program under development.

OPERATION: All three PRO/TEST modules are self-contained, batch-oriented programs with parameterized control cards. The principal differences between the two programs that generate test data are in the determination of file structure and the source of data for the output file. The File Checker works with the output of programs under development and ensures that the format agrees with the anticipated results described by the programmer.

In general, the Test Data Generator determines file structure (the grouping of record types into sets) by generation statements (including arithmetic calculation) in the input parameter cards; all data is generated by statements in the parameter cards. The File Processor determines file structure based on the structure of the input file; data passed into the output file can be selectively obtained from the input file, generated by statements, and/or calculated from input or generated data. Both PRO/TEST modules contain extensive facilities for determining the content of the output file based on comparisons of data fields with constants, other data fields, or calculated values, and the two modules are highly complementary.

Once a file has been generated by either test data module, the job is finished; there are no direct provisions for linking file generation with a user-program compilation or execution. A set of five types of parameter cards contains the specifications for either module. An Applications Data card, containing identifying information such as system description, date, and remarks, and a File Description/Options card, containing a description of the input (File Processor) and output files along with processing options such as report printing, output device selection, memory size available, etc., are common to both versions. The third card type is Generation Control (which specifies record grouping for the Test Data Generator) or Selection Control (which specifies record selection criteria for the File Processor). The other two types of cards, Record Identification and Field Description, are also common to both programs.

The elements of a Test Data Generator run are the input parameter cards, output listings (Audit Report, Analysis Report, and File-Print Report), and the output data file. Up to 8 million records can be generated, which should be ➤

PRO/TEST

Synergetics Corporation

▷ contract programming environment where there is a need to audit files produced by a program not directly under the user's control. This administrative management aspect of the File Checker makes it a versatile and handy tool indeed.

The PRO/TEST programs are free-standing; i.e., not linked to the compilation of the application program. This provides the additional flexibility of being able to produce files for different source languages. However, it has the disadvantage of not being able to draw on the program's file and record structure statements to reduce coding and ensure accuracy. Thus, there are some significant tradeoffs to consider. PRO/TEST will require a little more time to code than a compiler-linked data generator, but test data can be generated before the program is finished, which reduces the overall time required. The value of PRO/TEST for acting as a consistency check must be compared with the convenience of having data fields already laid out and named.

The principal strength of PRO/TEST is the flexible structure of the various specifications and directives. The feel of writing the PRO/TEST specifications is similar to programming itself, rather than filling in what seem to be arbitrary parameters. The format is straightforward; specifications for generating a file are quite intelligible, even after they are written. This promotes the use of PRO/TEST to generate and modify control files that will be used throughout the life of an application program to test program changes.

Aside from fulfilling a necessary step in the job of getting a program on the air, preparation of test data can bring some very useful information to the front about the relationships among the various records and data fields. Synergetics strongly recommends that test data be generated as a separate task from the programming; this serves to give a check on the consistency of the file structures and program definitions.

Users contacted by Datapro are enthusiastic about PRO/TEST and have thought up a variety of interesting uses for it. This is definitely one proprietary package that is worthwhile discussing with existing users before reaching a decision, just to get the true flavor of what it can be made to do. For example, one user generates segmented files. Another inputs parameters via a terminal. A third user creates data to audit implemented applications.□

➤ adequate for all occasions. The key concepts are the specification of file structure and data generation. The structure and content of individual records is determined by the Field Description cards and is discussed below. The structure of the file, on the other hand, is specified by the Generation Control card and is concerned with the groupings of different types of records within the file. For example, a master header record can be followed by one or more groups of different types of transaction records. Control is achieved by record names. The number of occurrences of each type of record within a group can be a fixed value or a value varying between limits, to fully exercise record processing logic in the target test program.

The data generation statements control the value assigned to a data field in successive records of the same type. The coding for each record type consists of a Record identification card giving the record name, followed by a series of Field Definition (FLD) cards. In general, there is one FLD card for each field and each element of a conditional statement.

Fields are identified by their beginning and ending locations relative to the beginning of the record. Three types of fields can be specified: generative, conditional, and computational. Any field can be assigned a constant (literal), range, or random value; data formats include alphabetic, alphanumeric, decimal numeric, packed decimal numeric, and binary numeric.

A generative field is explicitly defined in a statement. Typically, a set of values are specified and successive members of the set are assigned to the corresponding fields in successive records; the set can be recycled when the end is reached, or the last value can be held.

The operand of the field specifications is a data value of the appropriate type (alphabetic, decimal, etc.); alphanumeric items are used with automatic conversion to packed decimal or binary as required. The length of the data value to be assigned need not be as long as the field. In general, numeric entries are right-justified and zero-filled, while alphabetic entries are left-justified and space-filled. If no operand is specified, the data field will be automatically filled with a default character of appropriate type.

The value assigned a field can also be based on a comprehensive set of conditional operations. Data fields can be compared with a literal or another data field to determine the results. Relationships permitted include equal, greater, less, equal or greater, and equal or less. A data field can also be tested for certain conditions such as positive, negative, numeric, alphabetic, spaces, and zeros. The NOT relationship can be combined with any of the above to facilitate specification of logical relationships. In general, if a conditional specification is satisfied, a data field is generated according to its specifications; if not, then spaces or zeros are inserted according to the data type. Conditionals can be chained with AND and OR operators to allow complex logical relationships to be specified. Through an ELSE facility, an alternate data value can be generated.

Computational data fields can be dervied by adding, subtracting, multiplying, or dividing two data fields or a data field and a constant. All data conversions to packed decimal for computation are automatic. The result is in packed decimal, but specifying other formats is easy. The results of calculations are stored in one of ten 15-byte total areas. These areas can easily be used in the specification of data values output in the record.

Alphanumeric items can be stored temporarily in one of ten 15-byte save areas; the programmer also has easy access to these. This facility simplifies setting up constants to be used in several different records.

A total area can be reset to zero after each record or can be continued across several records. A save area remains unchanged until a new value is loaded.

The random specification always generates the same result from the same parameters; subsequent generations of a test file will then be identical to the original.

A repeat specification can be used with any field definition to force the same data value to be inserted in the specified number of successive records; each member of a set is repeated the specific number of times before going to the next number.

A very useful feature is the capability to copy parts of a previous record definition. The previous record is also named in the Record Identification card; only the parts that are different need to be coded. A maximum of 500 individual record names can be accommodated.

After a record is generated, a user subroutine can be entered for additional processing. Upon leaving the routine, ➤

PRO/TEST

Synergetics Corporation

several different actions can be specified. The current record can be written to the output file or deleted; control can then return to the Generator program for the next item generation or back to the subroutine. These actions can be set at the time the item is generated, based on the then-existing situation. This capability permits additional processing beyond the specification capabilities of PRO/TEST and additional control on records generated or deleted.

The File Processor is similar to the Test Data Generator, except that the file structuring facilities (Generation Control) are replaced with a powerful capability for selecting records (Selection Control) from an existing file. Live data is run through an "input select" facility that culls out records meeting user-specified criteria. Selected records are read into an input area accessible by the programmer. Fields selected from the input area or generated using the same facilities available to the Test Data Generator (using input fields as elements if desired) are placed into an output area. Additional total and save areas (a total of 20 each) are provided. Fields in the output area are accessible for further manipulation through the total and save areas. The key concepts are record selection and file generation.

Additional provisions allow selection based upon the following criteria:

- Record count; e.g., every fifth record between the 50th and 600th records in the input file.
- Sets; similar to record count except that a key field is used to identify sets of similar records to count; all records with the same key value count as one selection.
- Keys; all records whose key-field value lies between two limits will be selected.
- Contents; conditional statements relating input fields, constants, accumulated data, and generated data can be used to determine record selection; control is achieved through record names; this feature is normally used to control processing of different record types in the same file.
- Volume; limits the number of a particular type of record that will be selected.

Within the Selection Control card, one or more record names can be specified. These key to the Record Name card preceding a group of Field Description cards. More than one record selection request can be coded (up to 9,999) for the same file, which permits changing how records are selected as you progress through a file.

These facilities allow selection of all or representative records from files with hierarchial organization. Special facilities are provided for generating a record at the end of a set or file; these are normally used for summary totals and similar quantities.

Other facilities present in the File Processor include a field-type "scan" for editing (similar to the COBOL EXAMINE), an improved "edit" similar to that in Assembler language, table handling capability, and an output format control for use instead of direct file listing to review results.

Both the Test Data Generator and The File Processor can produce a File-Print Report and Audit and Analysis Reports. The Audit Report is a listing of the input parameter cards formatted to make reading easy; errors are noted by diagnostic codes. The Analysis Report lists the processing options in effect, including defaults, and the characteristics of the requested file.

PERFORMANCE: A11 PRO/TEST modules normally run at I/O speed, with all manipulation performed in main memory for the Test Data Generator and the File Processor (i.e., no working files on tape or disk are used). The File Checker uses one temporary working file, but still runs at about I/O speed.

Output for the test data generators can be placed on mgnetic tape, disk, data cell, or punched cards. Record format can be fixed or variable in length (tape only), blocked or unblocked, or undefined. Maximum record length is a generous 32,767 bytes. For disk files, the file organization can also be sequential or indexed sequential (complete with index generation). Segmented files such as those created with IBM's Data Language/1 (see IMS-2, Report 70E-491-01) can be constructed. Magnetic tape files can be written with no, standard, or non-standard labels, with header and trailer records, and with or without tape marks. Multi-reel files and multi-file reels can be generated. Rewind, no rewind, or unload can be specified at the completion of a file generation.

HARDWARE/SOFTWARE REQUIREMENTS: The Test Data Generator can be run on an IBM System/360 or 370, Model 30 or larger, with 65K bytes of core storage under DOS, OS, DOS/VS, or OS/VS. The minimum practical main memory requirement for the File Processor or the File Checker is about 48K bytes. One device or device area is required for the input file (File Processor) and one for the output file in addition to systems residency requirements.

The Honeywell version of the Test Data Generator will run on a Series 200 (or 2000), Model 110 or larger, with a minimum practical main memory requirement of 24K characters of core storage, or on a Series 600 (or 6000) with a minimum practical main memory requirement of about 24K words of core storage.

Larger main memory requirements for all the systems above may be imposed by large input blocking factors.

PRICING: Purchase price for the Test Data Generator is $5,750 for IBM systems, $7,500 for Honeywell Series 600 or 6000 systems, and $3,900 for Honeywell Series 200 or 2000 systems. The File Processor can be purchased for $4,750, and the File Checker costs $2,900. If any two or all three programs are installed, an overall discount of 25 percent is given. These prices include installation, documentation, forms, training (six-hour class/workshop), and full maintenance for five years.

Rental plans are available on a month-to-month basis with a one-time installation charge. For IBM systems, the installation charge for the Test Data Generator is $220, with monthly rental payments of $230; the File Processor installation charge is $310, with monthly rental payments of $190; and the File Checker installation charge is $34, with monthly rental payments of $116. A purchase option plan is also provided; contact Synergetics for details. Rental prices for Honeywell Systems, and lease terms of up to 5 years for all systems, are also available.

INITIAL DELIVERY: Test Data Generator – Summer 1969; File Processor – January 1971; File Checker – December 1971.

CURRENT USERS: About 70 for the Test Data Generator, 21 for the File Processor, and 5 for the File Checker. ■

Quikjob I and II
System Support Software, Inc.

MANAGEMENT SUMMARY

Quikjob I and Quikjob II are report and utility program generators for IBM System/360 and 370 DOS, OS, and VS systems that rival, respectively, Dylakor's DYL-250 and DYL-260 (Report 70E-400-01).

While both System Support Software's Quikjob and the Dylakor packages perform the same general functions for users and are similarly priced, the approach each company has taken to "quickie" utility program generation is distinctly different: Quikjob uses a language approach, while DYL is parameterized.

The language approach taken by Quikjob's vendor gives the user a free-form, English-appearing, logically sensible language made up of elements like those of an abbreviated COBOL Procedure Division. Using this approach, the user has a wider degree of program logic control. Meanwhile, standardization (of debatable value for one-shot program generators anyhow) is sacrificed. Whether it is easier for non-EDP personnel to use the simple Quikjob language or the DYL fill-in-the-blanks technique is a toss-up, and probably subject to personal preference. At least, with Quikjob, no special coding forms are needed.

System Support Software was founded in Ohio in January 1972. Quikjob, the company's first product, was tested for nearly one year, then initially installed at a customer's location in January 1973. A few months later, in response to customer feedback, Quikjob was enhanced with new and improved verbs and I/O options and re-dubbed Quikjob I. In July 1973, an improved package with the important capability of handling a second input file, and called Quikjob II, was released.

Quikjob is made available on a 30-day trial basis; it can be leased thereafter for successive 3-month periods. It is a load-and-go package that runs immediately after it is cataloged, a procedure that requires only minutes of computer time. The package adds, subtracts, multiplies, and divides EBCDIC and packed data, and can perform the latter two operations on binary data also. It performs rounding and decimal alignment while multiplying or dividing.

USER REACTION

Datapro owes its discovery of Quikjob to users responding to the recent Datapro survey of proprietary software users (Report 70E-010-40). In response to the survey, three Quikjob users rated the package either excellent in all categories (two users) or good in some categories and excellent in all others. All three respondents reported that the package performed as advertised immediately, and ▷

Quikjob, a "quickie" file manipulation program generator and report writer, offers a high degree of user logic control through a language similar to that of the COBOL Procedure Division. Quikjob I supports one input file, while the newer Quikjob II supports two.

CHARACTERISTICS

SUPPLIER: System Support Software, Inc., 1132 Donson Drive, Dayton, Ohio 45429. Telephone (513) 435-9514.

BASIC FUNCTION: Quikjob constructs and executes programs for reports, utility use, and file manipulation. Quikjob I operates with 1 input file, and Quikjob II can handle 2 input files.

Quikjob can simultaneously produce:

- A printed report with up to 9 total levels, plus a grand total,
- A punched card output deck, and
- Any combination of up to 3 tape and/or disk files that need not have any relation to each other or to the input file(s); the user has complete format control. Fixed, variable, and undefined (plus spanned for OS) record formats are handled. ISAM is read, created, and updated in OS. DOS reads and updates ISAM. Additionally, 2311, 2314-type, and 3330 disks can be used, and labelled and unlabelled tape files are supported.

OPERATION: Once cataloged from 800 or 1600 bpi tape or cards into the user's core image library, Quikjob is a load-and-go system. It runs at I/O speed. It has been used for the following functions: device to device file conversion (e.g., 2314 to 3330), test data generation, file editing and correction, file security backup and restore, report writing, file maintenance, IBM DITTO replacement, media conversion and data transcription, file printing for debugging, auditing, unit record (IBM 514 and 407) replacement, and data validation.

Quikjob uses a free-form language that is similar to an abbreviated COBOL Procedure Division. For example, GO TO, IF, PERFORM, ADD, PRINT, and MOVE are typical verbs. ACCUM is for accumulation reporting, GET reads an input record, HDR supplies report headings, and so on.

Table loading and subsequent lookup is possible Quikjob does not require elaborate file definitions, and no Data Division is required.

PERFORMANCE: The key performance criterion for a package of this type is "Can the job be coded and run quickly?" rather than any specification of execution speed. Users report that Quikjob gets the job done, but not all consider it advisable to use the package for production jobs – except, possibly, for ongoing special request jobs. ➤

Quikjob I and II
System Support Software, Inc.

that no modifications of the package were required. Intrigued by this unusually positive response, Datapro contacted one of the users, who kindly informed us where we could locate the vendor.

Using information supplied by System Support Software, we contacted several more Quikjob users. All seem to be quite pleased with the package and vendor. One user reported having placed Quikjob II on order (it was announced only two days before this report was written), and says that he uses Quikjob anytime it fits. Another user doesn't use the package for production, because he feels that one-shot programs don't pay off there as well as regularly coded programs.

One users reports having replaced DYL with Quikjob because his COBOL programmers preferred the COBOL-like structure of the latter.

In conclusion, Datapro feels that Quikjob is a package that should be investigated by any user concerned with requests that necessitate "quick and dirty" program generation. □

An important performance factor for Quikjob is the ease with which it can be learned and used; COBOL programmers will pick up the required knowledge almost immediately. Non-programmers can be taught to use the language in less than an hour for simple requests.

HARDWARE/SOFTWARE REQUIREMENTS: Quikjob runs on any DOS, OS, or VS IBM System/360 or 370. A minimum of 24K bytes of main storage is required under DOS, with more space being taken automatically as available for larger block sizes and a larger user table. Under OS, up to 45K bytes may be required. The operating system is not altered in any way.

PRICING: Quikjob I leases for $30 per month, or can be obtained on a permanent license for $2,300. Corresponding prices for Quikjob II are $50 per month and $3,300. Multiple-copy discounts are contracted on an individual basis.

There is no maintenance charge for Quikjob. Program fixes are mailed to all users at no charge, and new releases are offered to all users for only a materials and handling charge of $15.

INITIAL DELIVERY: Quikjob I – January 1973 (after nearly a year of on-site testing); Quikjob II – July 1973.

CURRENT USERS: Approximately 70. ■

Stage II
Tesdata Systems Corporation

MANAGEMENT SUMMARY

An interesting phenomenon has developed along with the use of high-level languages such as COBOL. Some of the facilities of the language which make coding easier make the resultant programs less efficient. In the days of assembly-language coding, a reasonable parallel could be drawn between the number of lines of code and the efficiency of the program (assuming, of course, that I/O operations were handled well). In COBOL, it isn't necessarily so.

Iterative loops, which look so neat on paper and agree so nicely with the blocks on flow charts, are not necessarily the most efficient way to process data. The savings in programming time are likely to be more than offset by losses in execution time. But that's not the bad part. Programmers conditioned to subscripted loops may use them where not necessary. In addition, many of the old techniques, such as binary searches, have fallen by the wayside simply because training is not available. Boiled down to its essentials, this line of discussion can be summarized by saying that: (1) compact source code does not mean compact, efficient object code, and (2) programmers do not always know the idiosyncrasies of the specific compiler that they are using (these vary from vendor to vendor – even for the same user-level version), and consequently do not know the best techniques for coding.

To these general observations, Tesdata adds the claim that the majority of IBM System/360 or 370 COBOL programs are CPU-bound, not I/O-bound as is traditionally believed. This is true because of inefficiencies in IBM systems software, particularly input/output control coding.

Thus, for users of IBM COBOL, there are three sources of inefficiency combining to use up a lot of processor time: programmer, COBOL compiler, and the systems control program(s) or operating system(s).

Tesdata Systems Corporation has been engaged since April 1972 in presenting seminars designed to inform IBM COBOL users about the relative execution times of various COBOL statements and to teach them effective techniques for optimizing the source code. The seminars (also available from Optimization Sciences, Inc. as Stage I) have been attended by about 800 participants and have received very favorable reviews.

Stage II essentially automates the identification of problem areas that might benefit from the application of techniques taught in the training seminars. Indeed, this training is essential to understanding the output of Stage II and is included in the price of the package.

▷

Stage II is a program for automating the scanning of IBM System/360 COBOL source code to identify potential places where improvements in efficiency can be made by applying techniques taught in Stage II seminars. The results can range from modest to dramatic.

CHARACTERISTICS

SUPPLIER: Tesdata Systems Corporation, 5330 Wisconsin Ave., Chevy Chase, Maryland. (Formerly Optimization Sciences, Inc.; then Clasco Systems, Inc.) Telephone (301) 652-9220.

BASIC FUNCTION: To determine areas of IBM System/360 COBOL source code that can be optimized, primarily with regard to improving execution time. A training seminar, included in the price, is required for maximum benefit.

INPUT: COBOL source program.

OUTPUT: Punched cards containing some recommended changes to the source program along with a listing indicating other possible areas of improvement. The printed listing contains the potentially most valuable areas for improvement. The listing identifies the lines of code and the kinds of modifications recommended, but the actual changes must be made manually.

PROCESSING: The Stage II program scans the COBOL source code looking for specific programming usages of statements, storage assignments, data formats, etc., that have proved profitable for optimization. Some are due to peculiarities or inefficiencies in the IBM compilers; some are due to inefficiencies in programmer logic. All techniques used are straightforward and do not depend on "tricks" that might not work if data values or ranges change over the period of use of the application program. Though a lot of individual items are checked, the principal areas of attack seem to be in the data formats assigned to variables and in the use of subscripted operations. Recommendations can include such items as employing COMPUTATIONAL-3 (packed decimal) formats for data items used in arithmetic operations, replacing iterative loops with straight-line coding, and specifying COMPUTATIONAL usage (binary) for variables used as subscripts. Many of the recommendations may seem prosaic when examined individually, but taken as a whole they form a comprehensive outline for improving the execution efficiency of COBOL programs.

PERFORMANCE: Stage II Version 3, the current version, takes about one-half as long to run as the compile time for the program being optimized.

Dramatic decreases in execution time have been achieved in some of the programs optimized during Stage II seminars and demonstrations. It is very difficult to estimate the savings to be expected in any particular installation. It will depend on how many "poor" techniques have been used by the programmers. It is probable that the majority of COBOL application programs can be improved by about 10 percent or more.

➤

Stage II
Tesdata Systems Corporation

▷ The techniques taught in the Stage II seminars do not represent any secret or magical approach, to quote Mike Kavanagh, developer of Stage II and presently Stage I instructor for Optimization Sciences. Indeed, many COBOL programmers have already discovered and are using one or more of the techniques. The seminars do, however, represent a comprehensive survey of the pecularities of the IBM COBOL compilers and loop programming methods. The Stage II training seminars, to paraphrase some of the attendees, provide a better insight into some of the things good programmers have discovered by empirical and/or intuitive methods.

Whether or not as many COBOL programs are compute-bound as Tesdata seems to think, improvement of processing speeds is desirable. The processing speed is just as pertinent to large System/360 and System/370 users, who are making use of multiprogramming, as it is to small installations (such as Model 30's and maybe Model 40's) where the inefficiences of the systems software and slow processor speeds may lead to compute-bound programs.

In addition to the potential for processing-speed gains ranging from modest to dramatic, main memory usage is affected. Notwithstanding Tesdata claims of saving "an average of 25 percent on production time" for Stage II mailorder processing using Optimail, typical overall savings of about 10 percent are to be expected, taking into account that on some programs you will lose a little.

Against these gains you must measure costs. To the actual cost of the package, you must add the time for the personnel receiving COBOL optimization training, the cost of executing the Stage II program for each COBOL source program to be optimized, and, most importantly, the time for the implementation of the recommendations. In one case, it took several man-weeks of a programmer's effort to implement the changes to a large program (about 100K). The cost of optimization for a program must be measured against the expected savings, taking into account the frequency of running the program. It doesn't really make sense to optimize a small program that is run only once a month, for example.

Tesdata offers sound advice along these lines: get the program running first and then worry about optimization, if the program is important enough to optimize.□

➤ The sample program distributed by Tesdata as part of its promotional literature is a good example of a program that can be improved dramatically. It is a cross-tabulation of sales results. Separate tabulations for 23 divisions are prepared within 50 states and within 140 territories. The data in each input record must be distributed to a set of state totals and to a set of territory totals. At the end a single, large report is printed. This is a good program for demonstration purposes because it neatly separates computation and output printing. The dramatic execution-time saving of 96 percent was achieved through the use of many devices, but the principal culprits in this program were two serial searches to properly identify the state and territory totals applicable for each record. Using the state and territory codes directly as subscripts rather than as a key to identify the proper subscripts to be used in the accumulation phase through a sequential series of comparisons with a set of constants held in memory allowed the elimination of many, many execution cycles. A simple correction, you say? My programmers don't do anything so silly, you say?

How many of the hundreds or thousands of COBOL programs in your shop have you personally checked in sufficient detail to really have confidence in your reply? This points up one subjective benefit of Stage II: the ability to quickly check programs to see that good programming practice has been adhered to.

HARDWARE/SOFTWARE REQUIREMENTS: Stage II will run on an IBM System/360 or 370 configuration under DOS or OS in a 54K to 80K partition or region. It will also run under DOS or TDOS on a UNIVAC Series 70 (formerly RCA Spectra 70).

A printer, card reader and punch, and three tape or disk work files are also required.

PRICING: The basic price of Stage II includes enrollment of 15 persons in a seminar given at the customer's site. Maintenance for Stage II costs $250 per year after the no-charge maintenance period.

Stage II processing is also available on a mail order basis through Optimail, at the same address as Tesdata. Optimail charges are $50 for handling plus $25 per program and $0.04 per card. Thus, for example, a 2000-card deck containing three programs can be processed thru Optimail for $205. Optimail is intended to be used for one-time optimization needs, for Stage II test usage (for a paid-by-the-customer demo), or by users whose systems are too small to permit the use of Stage II in-house.

	5-year License	
	First Installation	Each Additional Installation
Stage II (includes 2 years of Maintenance)	$3,600	$2,600
Stage II (includes 5 years of Maintenance	4,100	3,100
Renewal (5 years) without maintenance	1,800	1,300

INITIAL DELIVERY: Stage II seminars began in April 1972. (Stage I seminars were first offered in April 1971). First delivery of the Stage II program is expected shortly.

CURRENT USERS: About 800 persons representing over 300 companies and government agencies had participated in Stage I or Stage II seminars as of June 1972. There were no actual users of Stage II at that time, but Stage II has been used in the seminars to check the manual optimization results of participants.■

CASE
Tesdata Systems Corp.

MANAGEMENT SUMMARY

CASE is designed to simulate a particular computer configuration executing a specific group of programs. It draws on a library of software and hardware parameters, together with extensive input information concerning the configuration and processing tasks, to detail and summarize the performance of the system. CASE can simulate real-time or multiprogramming environments as well as straight batch operations.

CASE (Computer-Aided System Evaluation), one of the first simulation programs to compete with Comress Inc.'s widely used SCERT system, enables a user to simulate the performance of a proposed computer system by executing a program on a computer system he already has. Or it allows him to conveniently investigate the performance of his own computer system on projected workloads. The reasons for performing this operation can be to evaluate future processing needs, to evaluate equipment proposals, or to improve the operation of his present system by making judicious equipment changes.

CASE really has two parts: the Independent Processing Analyzer (IPA) and the Concurrent Processing Analyzer (CPA). IPA is for non-multiprogramming environments where jobs are handled sequentially; CPA is for multiprogramming and real time environments.

Use of IPA can be thought of in three steps:

1. Preparation of input specifications which detail the workload and equipment configuration.

2. Execution of IPA, which draws on the CASE Library for equipment and software performance factors and outputs reports telling how the system would perform the jobs.

3. Analysis of the reports by a trained analyst to interpret the results.

The definition of the workload can be made as simple as specifying a general business or scientific mix, or as complex as identifying operations to be performed on each field. In any case, file size and layouts (field sizes) must be determined.

CASE simulates specific equipment configurations. It will give an indication of how a specific component is utilized. It won't tell you which component is best—that's the role of the trained analyst.

The results of IPA are centered around "run delay," the portion of time the processor is inactive.

By making several runs with different configurations and different computer systems, a picture can be built up showing whose equipment and which configuration is best suited for a proposed workload.

However, it is not just a matter of comparing numbers. Some of the techniques used by CASE to allocate time, particularly idle time, are tricky and need the understanding of a trained analyst, who may wish to revise ▷

CHARACTERISTICS

SUPPLIER: Tesdata Systems Corp. (formerly CLASCO Systems Inc.), 5530 Wisconsin Avenue, Chevy Chase, Maryland 20015 Telephone (301) 652-9220.

BASIC FUNCTION: To analyze the processing time requirements of a specific set of programs running on a specific computer system configuration. In effect, CASE simulates the operation of a computer system. The user identifies the configuration, file details, processing details, and frequency of occurrence of all elements. CASE takes these inputs and draws from a library of hardware and software characteristics to estimate the processing time required for each element and to show the balance between the processing power of the mainframe and the speed of the peripheral devices.

The Independent Processing Analyzer (IPA) evaluates the performance of individual programs. The Concurrent Processing Analyzer (CPA) is used to analyze a multiprogramming computer configuration and identify potential bottlenecks, using as input the results generated by the CASE Independent Processing Analyzer.

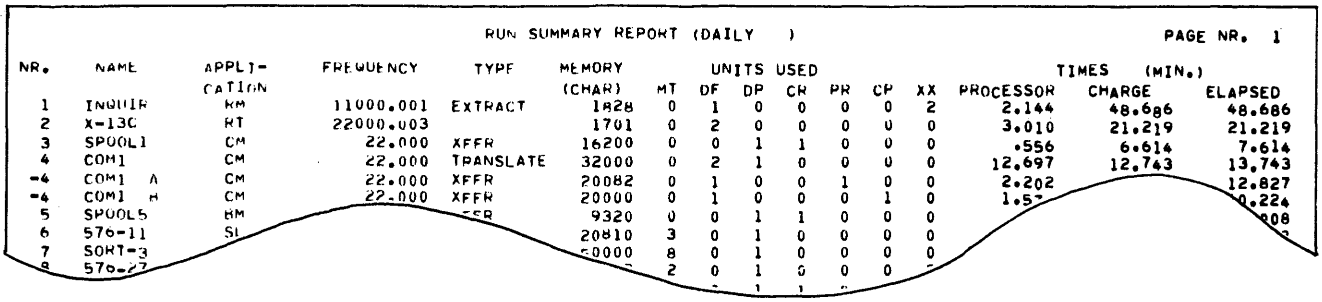
INPUT: The user fills out three forms (file specifications, run specifications, and configuration specifications) for a run involving the Independent Processing Analyzer (IPA).

Files are identified by name. The input form requires specification of device assignment, blocking factor, and number of records. Record length is identified by maximum and average length in characters and the number of numeric characters. Magnetic tape densities can be specified. Each file is designated as a master or non-master file.

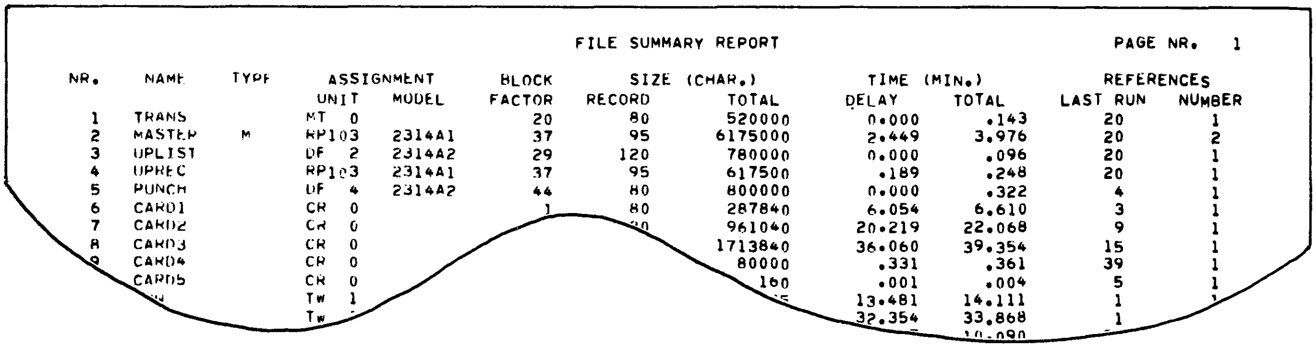
OS System Management Facility (SMF) data can be used as direct input to CASE to describe the system's files more precisely. Also, Data Language/1 (DL/1) parameters can be entered directly to simulate an IMS-2 DB/DC environment (see Report 70E-491-01). Data Base Description parameters for other systems can also be entered to describe non-IBM DB/DC environments.

The run specification form identifies all files for each run and specifies the file and processing activity. File activity is specified in terms of the percentage of records that will be active as well as the type of access (random, sequential, etc.). Processing activity can be defined at various levels. A general mix such as business or scientific can be called for, or basic processing functions such as move, ►

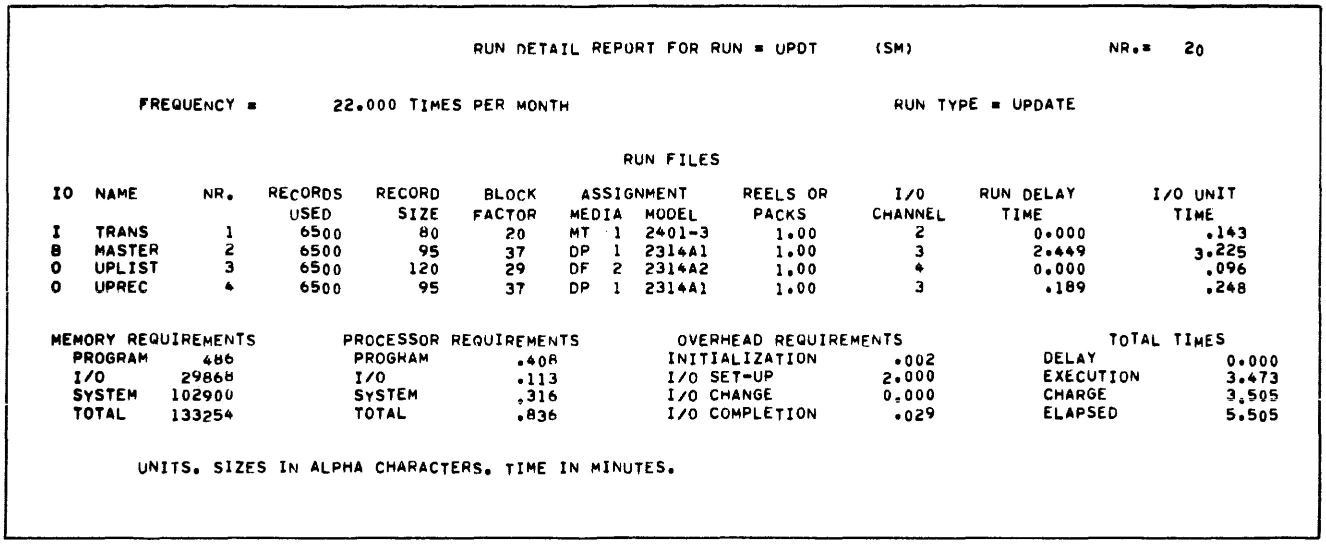
CASE
Tesdata Systems Corp.



```
                                        RUN SUMMARY REPORT (DAILY    )                                          PAGE NR.  1

NR.    NAME      APPLI-      FREQUENCY      TYPE       MEMORY          UNITS USED                          TIMES    (MIN.)
                 CATION                                (CHAR)   MT   DF   DP   CR   PR   CP   XX   PROCESSOR   CHARGE     ELAPSED
 1    INQUIR       RM       11000.001    EXTRACT        1828    0    1    0    0    0    0    2      2.144     48.686     48.686
 2    X-13C        RT       22000.003                   1701    0    2    0    0    0    0    0      3.010     21.219     21.219
 3    SPOOL1       CM          22.000    XFFR          16200    0    0    1    1    0    0    0       .556      6.614      7.614
 4    COM1         CM          22.000    TRANSLATE     32000    0    2    1    0    0    0    0     12.697     12.743     13.743
-4    COM1   A     CM          22.000    XFFR          20082    0    1    0    0    1    0    0      2.202                12.827
-4    COM1   B     CM          22.000    XFFR          20000    0    1    0    0    0    1    0      1.5                    .224
 5    SPOOL5       BM                                   9320    0    0    1    1    0    0    0                              08
 6    576-11       SL                                  20810    3    0    1    0    0    0    0
 7    SORT-3                                            0000    8    0    1    0    0    0    0
      576-27                                                    2    0    1    0    0    0
```



```
                                             FILE SUMMARY REPORT                                           PAGE NR.   1

   NR.    NAME     TYPE     ASSIGNMENT         BLOCK      SIZE (CHAR.)           TIME (MIN.)           REFERENCES
                           UNIT    MODEL      FACTOR    RECORD       TOTAL      DELAY       TOTAL     LAST RUN   NUMBER
    1    TRANS             MT  0                 20        80       520000      0.000        .143        20         1
    2    MASTER     M      RP103   2314A1        37        95      6175000      2.449       3.976        20         2
    3    UPLIST            DF  2   2314A2        29       120       780000      0.000        .096        20         1
    4    UPREC             RP103   2314A1        37        95       617500       .189        .248        20         1
    5    PUNCH             DF  4   2314A2        44        80       800000      0.000        .322         4         1
    6    CARD1             CR  0                  1        80       287840      6.054       6.610         3         1
    7    CARD2             CR  0                                    961040     20.219      22.068         9         1
    8    CARD3             CR  0                                   1713840     36.060      39.354        15         1
         CARD4             CR  0                                     80000       .331        .361        39         1
         CARD5             CR  0                                       160       .001        .004         5         1
                           TW  1                                                13.481      14.111         1
                           TW                                                   32.354      33.868         1
```



```
                                   RUN DETAIL REPORT FOR RUN = UPDT      (SM)                        NR.=   20


        FREQUENCY =        22.000 TIMES PER MONTH                                        RUN TYPE = UPDATE


                                                       RUN FILES

IO   NAME        NR.    RECORDS    RECORD     BLOCK     ASSIGNMENT      REELS OR       I/O      RUN DELAY           I/O UNIT
                          USED       SIZE     FACTOR   MEDIA  MODEL       PACKS      CHANNEL       TIME                TIME
I    TRANS        1       6500        80        20     MT  1  2401-3       1.00         2          0.000                .143
B    MASTER       2       6500        95        37     DP  1  2314A1       1.00         3          2.449               3.225
O    UPLIST       3       6500       120        29     DF  2  2314A2       1.00         4          0.000                .096
O    UPREC        4       6500        95        37     DP  1  2314A1       1.00         3           .189                .248

MEMORY REQUIREMENTS          PROCESSOR REQUIREMENTS            OVERHEAD REQUIREMENTS                      TOTAL TIMES
  PROGRAM      486             PROGRAM           .408          INITIALIZATION        .002          DELAY                 0.000
  I/O        29868             I/O               .113          I/O SET-UP           2.000          EXECUTION             3.473
  SYSTEM    102900             SYSTEM            .316          I/O CHANGE           0.000          CHARGE                3.505
  TOTAL     133254             TOTAL             .836          I/O COMPLETION        .029          ELAPSED               5.505


        UNITS. SIZES IN ALPHA CHARACTERS. TIME IN MINUTES.
```

These summary and detail reports represent, in abbreviated form, the output of the CASE Independent Processing Analyzer. In the Run Detail report, the data to the left of the Run Delay Time column is part of the input to the program. The delay time represents the portion of the I/O cycle during which the processor is not active. Proper interpretation of these results requires a trained analyst.

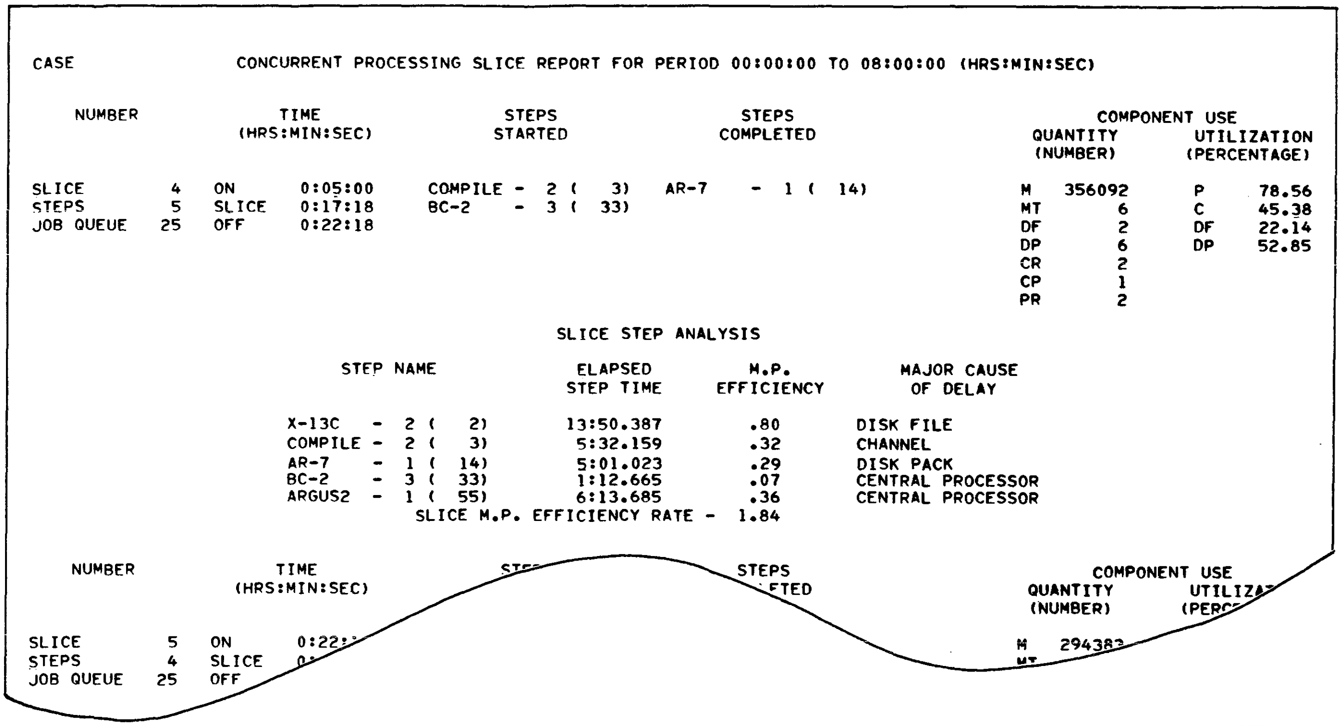
the figures before presenting them to management. Similarly, some of the allocations of processor time and considerations of disc access times are probably handled as well as can be done currently, but the analyst may want to add his own interpretations. Keep this in mind when you're looking at the illustrations in this report.

The CASE Concurrent Processing Analyzer (CPA) goes beyond the CASE Independent Processing Analyzer. CPA estimates the amount of time that will be used in a multiprogramming situation by a set of runs, some of

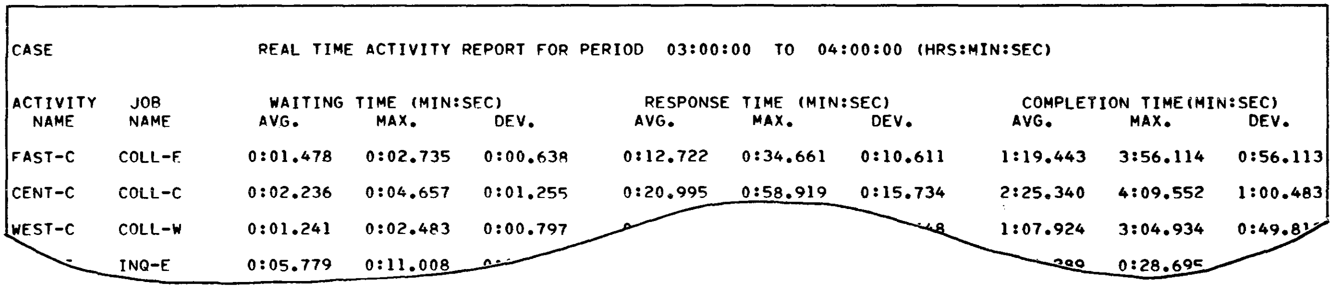
add, compare, etc. can be specified. The user can indicate the type and number of processing operations to be performed on each field. The various levels of processing specifications accommodate different levels of knowledge about the programs in the different runs. Also specified in the run form are detailed printing formats for output files.

The configuration specifications identify all components being used, such as processor model, memory size, magnetic tape units, operating system, and language processors. Typically, this list includes from 12 to 40 items.

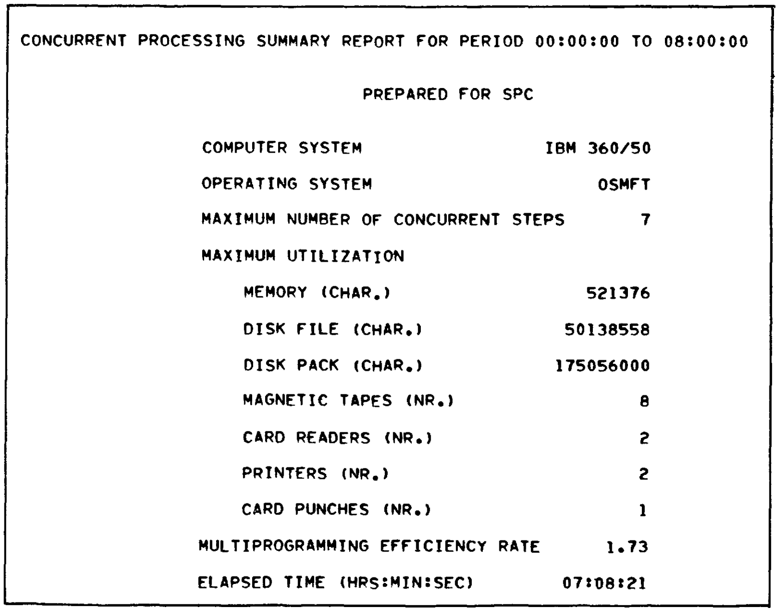
CASE
Tesdata Systems Corp.



```
CASE             CONCURRENT PROCESSING SLICE REPORT FOR PERIOD 00:00:00 TO 08:00:00 (HRS:MIN:SEC)

     NUMBER              TIME                 STEPS                 STEPS                          COMPONENT USE
                     (HRS:MIN:SEC)           STARTED              COMPLETED                 QUANTITY       UTILIZATION
                                                                                            (NUMBER)      (PERCENTAGE)

SLICE          4   ON       0:05:00    COMPILE -  2 (    3)    AR-7      -  1 (  14)          M    356092       P    78.56
STEPS          5   SLICE    0:17:18    BC-2      -  3 (   33)                                  MT        6       C    45.38
JOB QUEUE     25   OFF      0:22:18                                                           DF        2       DF   22.14
                                                                                            DP        6       DP   52.85
                                                                                            CR        2
                                                                                            CP        1
                                                                                            PR        2

                                         SLICE STEP ANALYSIS

                          STEP NAME            ELAPSED        M.P.            MAJOR CAUSE
                                              STEP TIME    EFFICIENCY          OF DELAY

                    X-13C    -  2 (    2)     13:50.387        .80        DISK FILE
                    COMPILE  -  2 (    3)      5:32.159        .32        CHANNEL
                    AR-7     -  1 (   14)      5:01.023        .29        DISK PACK
                    BC-2     -  3 (   33)      1:12.665        .07        CENTRAL PROCESSOR
                    ARGUS2   -  1 (   55)      6:13.685        .36        CENTRAL PROCESSOR
                          SLICE M.P. EFFICIENCY RATE -  1.84

     NUMBER              TIME             STE                   STEPS                          COMPONENT USE
                     (HRS:MIN:SEC)                                FTED                     QUANTITY       UTILIZA
                                                                                            (NUMBER)       (PERC

SLICE          5   ON       0:22:                                                           M    29438
STEPS          4   SLICE    0:                                                              MT
JOB QUEUE     25   OFF
```



```
CASE                     REAL TIME ACTIVITY REPORT FOR PERIOD  03:00:00  TO  04:00:00 (HRS:MIN:SEC)

ACTIVITY   JOB          WAITING TIME (MIN:SEC)               RESPONSE TIME (MIN:SEC)             COMPLETION TIME(MIN:SEC)
  NAME     NAME        AVG.       MAX.       DEV.          AVG.       MAX.       DEV.          AVG.       MAX.       DEV.

FAST-C    COLL-E     0:01.478   0:02.735   0:00.638     0:12.722   0:34.661   0:10.611     1:19.443   3:56.114   0:56.113

CENT-C    COLL-C     0:02.236   0:04.657   0:01.255     0:20.995   0:58.919   0:15.734     2:25.340   4:09.552   1:00.483

WEST-C    COLL-W     0:01.241   0:02.483   0:00.797                            48          1:07.924   3:04.934   0:49.81

          INQ-E      0:05.779   0:11.008                                                      289     0:28.695
```



```
CONCURRENT PROCESSING SUMMARY REPORT FOR PERIOD 00:00:00 TO 08:00:00

                             PREPARED FOR SPC

                  COMPUTER SYSTEM                      IBM 360/50

                  OPERATING SYSTEM                          OSMFT

                  MAXIMUM NUMBER OF CONCURRENT STEPS            7

                  MAXIMUM UTILIZATION

                      MEMORY (CHAR.)                       521376

                      DISK FILE (CHAR.)                  50138558

                      DISK PACK (CHAR.)                 175056000

                      MAGNETIC TAPES (NR.)                      8

                      CARD READERS (NR.)                        2

                      PRINTERS (NR.)                            2

                      CARD PUNCHES (NR.)                        1

                  MULTIPROGRAMMING EFFICIENCY RATE           1.73

                  ELAPSED TIME (HRS:MIN:SEC)             07:08:21
```

These reports are a portion of the output from the CASE Concurrent Processing Analyzer (CPA) for simulating multiprogramming and real-time (communications) processing. This format and the factors reported represent an improvement over previous CASE versions, but a trained analyst is still required for making adjustments to the system modeled. (The reports produced by the latest version of the CPA differ in some details.)

> which may be interrelated and others of which may be independent. ▷

➤ **Input to the Concurrent Processing Analyzer consists of a series of run descriptions identifying the processor time,** ➤

CASE
Tesdata Systems Corp.

The principal function of CASE CPA, however, is not to estimate the length of time that programs will run in a multiprogramming system. Its real function is to select, among different methods of organizing a system, the method most likely to yield the results the installation desires. Thus, CASE CPA is really a compararer rather than a predictor. It is, therefore, an iterative tool for management rather than a forecasting tool.

It is important from a user's point of view to bear this discrimination clearly in mind. Systems analysts frequently meet a situation where a job can be organized in a number of different ways, and the selection among the different techniques is rather arbitrary. The usual method of selection is simply to take the most pedestrian and obvious approach. This approach actually has many advantages–it can be fitted into the current system easily, it is easy to explain to other departments, and it is easy to implement.

However, the obvious way may not be the optimum way.

The function of CASE is to support an analyst who wants to do better than just take the obvious route. In the past he has normally been unable, because of the extensive work involved, to redo the system design more than two times at the most. Two times is inadequate to obtain an optimum solution. With CASE available to automate much of the work involved, many more iterations can be made.

The position of CASE CPA relative to CASE IPA is a fairly simple one. The Concurrent Processing Analyzer uses the output of the IPA system and assumes that it is accurate.

Evaluating CASE is difficult. Its success is highly dependent on the "goodness" of the library of hardware and software timing parameters. This is a highly proprietary area, and Tesdata is understandably reluctant to release bulk information about what is in the library. The CASE package will produce reports that list all characterists used in a specific simulation. The company promises to provide "periodic" updatings of the library.

Currently the Library includes almost all of the major computer systems that most users are likely to have or contemplate acquiring–plus a few surprises such as the DEC PDP-8 and PDP-10, the Xerox Sigma 2, 5, 6, 7, and 9, Xerox 930, emulation of the IBM 1400 series on a System/360 Model 30 or 40, and the time-sharing UNIVAC Series 70/46 and IBM System/360 Model 67.

Many computer people who are well aware of the sheer complexity of a computer system and the difficulty of obtaining reliable estimates of processing time, not to mention operating system overhead, may well be skeptical of a program that gives easy, all-inclusive answers.

memory required, etc. These run descriptions are taken from the output of the CASE Independent Processing Analyzer (IPA). As the two CASE phases are usually run together, the input for the IPA section also forms input for the CPA section.

In addition, a series of control cards identifies constraints, priorities, and other factors affecting the multiprogramming operating environment.

The control cards specifically identify:

- **Number of IPA input files and starting time.**
- **Priority scheduling of high-priority jobs.**
- **Priority job execution and successor/predecessor relationships.**
- **Size of memory and number of peripheral units.**
- **Partitioning and job step limitations.**

Other factors, such as limiting utilization of various components, and program options, such as optimizing, are also specified.

PROCESSING: CASE IPA takes the file, run, and configuration specifications and draws on the library to calculate the processing and input/output times.

CASE CPA lays out the time and resources used for each program and real-time (communications) task. Programs are "entered" into the simulated system as soon as there are facilities available and no higher-priority job is outstanding.

The schedule is developed through the use of a constraint analysis routine, using the limitations set up in the input. The program forms a network, identifying the earliest positions at which the programs can be scheduled, the latest positions at which the programs can be scheduled if they are not to cause delays in other areas, and the slack time–very much like a critical path network.

CASE LIBRARY: Included with the master CASE program, the library consists of a set of detailed parameters about hardware and software components, including operating systems and language processors. It is updated periodically to reflect new releases from the computer manufacturers.

The library includes parameters for just about all of the major second- and third-generation computer systems, including the time-sharing IBM System/360 Model 67 and UNIVAC Series 70/46. In addition, some computer systems from Digital Equipment Corporation and Xerox appear in the Library, including the DEC PDP-8 and PDP-10, the Xerox Sigma 2, 5, 6, 7, and 9, and the Xerox 930.

A comprehensive selection of Teletype terminals, as well as Bell System and Western Union modems, are included, along with entries for the Raytheon DIDS-400 display system. Other library entries that are somewhat surprising are the EMR 6130, IBM 650, and UNIVAC 1004/1005. Omissions include the second-generation systems from Honeywell and RCA. Entries for System/360 Model 30 and Model 40 emulation of the IBM 1400 Series computers are also provided.

CASE
Tesdata Systems Corp.

The question then becomes: how well does CASE perform? The company states that overall CASE estimates are typically within 10% of actual timings–a very creditable performance. However, estimates for some individual runs or programs such as compiles may be as much as 50% off. This points up the primary use for CASE–as a comparative tool rather than a predictor.

Several factors tend to minimize the obvious doubts about the validity of the approach used by CASE (and other similar programs) for estimating computer system performance. The processing power of mainframes today in comparison with peripheral capabilities means that the peripheral devices are the limiting factors in most situations (as demonstrated by the current emphasis on multiprogramming). This condition is favorable because estimating peripheral performance is considerably easier than estimating the internal processing load. Second, most operating systems handle most functions in a relatively straightforward way, again simplifying the estimating procedure.

The CASE system requires the use of a skilled analyst and is most often used in large, expensive installations that can economically afford such a person. Today, these large installations are run almost entirely in a Multiprogramming mode, and practically every future installation of this class will be considered a multiprogramming system. As such, the information provided by the CASE IPA is insufficient to provide the necessary system optimization to maximize throughput, even though it may be sufficient to check out procurement details. With the addition of the CASE Concurrent Processing Analyzer, a reasonable simulation and optimization of the overall loading can be handled, and a design can be chosen as a result of more extensive knowledge of the opportunities and the problems.

Tesdata Systems emphasizes the need for trained analysts in order to get reliable results and is willing to provide that training for customer personnel. The company estimates the average training time at five days. Other seminars for executive personnel are also available. Tesdata will also provide simulation services for users not wishing to install CASE.

However, more than systems analysis work is required. Management must be involved also, because what appears on the surface to be a good theoretical solution may have practical difficulties as a result of other departmental problems. To gain the advantages that are available from multiprogramming and resystemization, which can be very large indeed, negotiations and compromises are often involved. With CASE on his side, the systems analyst will be able to document the cost in computer time of some of the various compromises, and therefore take a constructive part in the negotiations. However, providing the time and the necessary strength

In general, software entries for each system cover the available language processors (COBOL, FORTRAN, PL/I. ALGOL, and Assembly), sorts (tape and disc/drum), and operating systems. The coverage sppears to be reasonably complete for the popularly used pieces of software in each system.

OUTPUT: The output from CASE IPA and CPA runs is a series of detail and summary reports that present the picture of the computer workload specified.

The basic report furnished by IPA is the Run Detail Report. This report includes a summary of the system and file characteristics taken from the input. The key factors in this report are the times reported for each file and for the processor. In addition to showing a time for overhead associated with initialization and I/O set-up, two times are shown for each file: unit time and run delay time. The unit time reflects the total time this file (and corresponding I/O unit) are being used. The run delay time shows the portion of the total unit time during which the processor is not active. In this way inefficiencies of use are pointed out. Places where a faster I/O device would be useful are pinpointed.

The three summary reports show a summation for: (1) all runs, (2) all files, and (3) all peripherals. Taking several CASE runs for different configurations, the "optimum" balance of I/O components and mainframe processing power can be achieved. The summary reports allow time-consuming runs and files to be identified.

In addition to the Run Detail Report, IPA produces a host of other reports that detail or summarize the configuration, files, processing tasks, peripheral utilization, input/output channel utilization, and overall performance.

The organization of the reports produced by CASE CPA is similar to that for IPA. A central report establishes the performance characteristics for each processing step in terms if "slices." A slice for CASE is defined as the period between changes in activity: i.e., a slice is completed each time the activity changes. This method leads to reporting on steps while they "are in the middle of being processed." The Concurrent Processing Slice Report, as can be seen in the accompanying illustration, enumerates the steps active in each slice, reports the processing accomplished for each step during the slice, and shows the utilization of the various configuration components in use.

To provide an idea of the effectiveness of the configuration in a multiprogramming mode, a measure called "M.P. Efficiency" is employed in both the Slice Report and in some of the supporting reports. However, different definitions of the term are used. For example, in the Slice Report M.P. Efficiency (now called Slice Effective Progress Rate) measures the extent to which several different tasks can be kept going. There is no direct relationship with any processing steps. In one of the summary reports, which shows overall results for each step, M.P. Efficiency measures the elapsed time to accomplish a step compared to the time it would have taken on the same configuration running in a monoprogramming mode (one job active). In addition, these figures only measure the efficiency of the tasks as defined by the output of IPA. Any one or all might be poorly organized and thus inefficient, no matter how much overlap with other tasks is achieved. The preceding comments are intended not so much to critize the use of M.P. Efficiency as a measurement as to

CASE
Tesdata Systems Corp.

▷ during negotiations so as to insure that the computers are being used at maximum efficiency is not the analyst's job, but management's.

All in all, CASE represents a useful first tool for use in large installations for comparing projected workloads on definite computer system configurations. But the tool is neither cheap nor simple to use. A trained analyst is required to utilize the output of the program, and a significant amount of time is required to set up the input and interpret the results. □

➤ point out the utter impossibility of a single-number measurement for a complex computer system. However, in all fairness, CASE's M.P. Efficiency doesn't do badly, and it is only one of the measurements indicating efficiency.

In addition to the Concurrent Slice Report, numerous detail and summary reports furnish a wealth of information for study. Included are reports showing performance of each job step, component contention with queuing and utilization factors, scheduling constraints, configuration requirements, and overall performance.

For real-time processing tasks (such as data communications), a special series of four reports is generated. The first shows processing factors such as waiting time, response time, completion time, executions per period, and gross time for static conditions (i.e., no dynamic impact from other processing tasks). The static and dynamic memory requirements are shown in a second report. The third report shows waiting, response, and completion times for each activity. Times are in terms of average, maximum, and deviation. This presentation permits a better interpretation of the possible effects of a particularly bad sequence of events, a sequence not anticipated in the original specifications. The degree of overcapacity required to handle such a situation can be judged to some extent by proper interpretation of the times presented. (Some allowance for nonpredicted sequences is necessary because the actual environment is seldom well enough known to permit precise modeling.) The last report presents the total time for each job step, again in terms of average, maximum, and deviation.

The full complement of reports, including multiprogramming and real-time analysis, numbers about 45, three of which may run a dozen or more pages each. Not all reports need be generated for each use of CASE. Problem areas can be identified, modifications made, and new runs made without generation of all reports. This cuts down the paper generated to some extent. The two central reports, however, are the main basis for interpretation of results, and one or the other will be generated for each iteration.

PERFORMANCE: Execution times vary with the hardware configuration used to run CASE. Typical times required for a CASE IPA run are 120 seconds on an IBM System/360 Model 50, 30 seconds on a Control Data 6600,80 seconds on a Honeywell 635, and 50 seconds on a UNIVAC 1108. Sample run times for simulation of a 200-file task with CASE CPA range from 10 to 15 seconds on computer systems such as the Control Data 6600, Honeywell 635, and UNIVAC 1108, and 70 seconds on an IBM System/360 Model 50.

HARDWARE REQUIREMENTS: 256K bytes of main memory plus 1 million bytes of mass storage.

SOFTWARE REQUIREMENTS: CASE will run under almost any operating system. Tesdata will install it on the user's system. Case consists of three overlay modules of 155KB, 175KB, and 313KB each.

PRICING: The two analyzers are licensed as a package for $9,000 on a single-year basis, $12,000 for a 3-year term, or $15,000 for a 5-year term. Monthly updates are provided at $500 per month during the license period. A wide variety of additional license plans is available, depending upon term. Contact Tesdata for details.

INITIAL DELIVERY: July 1969.

CURRENT USERS: More than 50. ■

EDOS
The Computer Company

MANAGEMENT SUMMARY

EDOS (Extended Disk Operating System) may just be one of the best things that has yet come along for IBM System/360 or 370 DOS users. EDOS is a collection of additions, performance enhancements, and an extended spooling facility for DOS Release 26 that can easily improve the thoroughput of a typical DOS installation by as much as 45 percent. Also included are facilities that substantially streamline operations and permit the user to obtain some of the most significant advantages of full OS within the DOS environment. And all of this is available at a cost paralleling that of a typical DOS spooling supplement and for as little as one-fourth to one-third the cost of those EDOS features that are available separately from other companies.

Note that DOS Release 27 – the last "real" DOS version – and Releases 28-plus – DOS/VS – are available for the System/370 only. Release 26 (or lower), however, can be run on the 370 as well as the 360.

Among the more than a dozen major "OS-type" features found in EDOS are the following:

- Full-fledged job accounting statistics, prepared with a report generator included in the system.
- Automatic volume sensing, allowing the operator to mount tape and disk volumes with minimal console intervention.
- User program relocatability.
- Dynamic priority reassignment to balance CPU- and I/O-bound problem-program partitions for increased throughput.
- Source program library facilities, enabling the user to store, modify, and retrieve his programs more efficiently.
- Support for up to six partitions within DOS (one of only two extra-cost options in the system), allowing up to twice the usual number of job streams to be processed in parallel.
- Comprehensive I/O spooling on disk for up to 63 "virtual" devices (the second extra-cost EDOS option).

The extended or six-partition support feature has certainly drawn the most attention for EDOS, but it should be recognized that six-partition support, while significant, is only one aspect of EDOS. And not its most significant aspect at that. Six partitions can give the use more flexibility in loading job streams, and the capability does provide a better opportunity for the load-balancing algorithms to drive the system more efficiently; but an operation with that many mouths needs a lot of hands to feed it. In that sense, six-partition support may turn out a richer meal than many DOS installations can digest–although six partitions are by no means unreasonable when compared to a typical OS environment. In fact,

▷

EDOS is a collection of enhancements for IBM's DOS Release 26 that extend the power of DOS toward that of OS. Significant savings can be realized by putting this heavily-muscled package into DOS, and the useful life of System/360 computers can thereby be prolonged.

CHARACTERISTICS

SUPPLIER: The Computer Company, Seventh and Franklin Building, Richmond, Virginia 23219. Telephone (703)644-1841. EDOS is also marketed by the CIG Computer Products Group, a subsidiary of Computer Investor's Group, 1351 Washington Blvd., Stamford, Connecticut 06902. Telephone (203) 359-2100.

FUNCTION: EDOS is a major supplementary package to IBM's Release 26 of DOS for use on System/360 or 370 computer systems. EDOS provides a number of operational features and capabilities otherwise offered by IBM only for OS. While many of EDOS's capabilities are not unique as DOS supplements–some are currently offered on an individual basis by a host of independent vendors–EDOS has the only approach that incorporates these features (plus others) as an integrated part of the user's operating system.

OPERATION: Starting with an operational version of IBM's DOS Release 26, the user loads the basic EDOS libraries on the DOS System Residence Pack, and generates a new supervisor specifying only one additional macro. This new supervisor is either 2K bytes larger than the unaided DOS, or 4K bytes larger if six-partition support is included, or 2K or 4K bytes larger if spooling is included. (Note that all increases to the DOS supervisor must be on a 2K-byte boundary according to conventions established in the design of DOS by IBM, and that the above additional requirements are cumulative).

Following this Sysgen, the rest of the EDOS modules are link-edited, and an initial program load (IPL) is performed to bring up basic EDOS (or "A" System under EDOS terminology) with three partitions. After initialization of the "A" System, a second "initial" program load can then be performed to set up the "B" System, which supplements EDOS with up to three additional partitions. Each partition is provided with hardware/software storage protection. (Note that although there are only three such protect keys in the System/360 hardware, an EDOS software scheme effectively extends such protection to the "B" system.)

The user at this point can begin to process his workload under EDOS. There are some 14 significant enhancements and/or extensions made to IBM's DOS by EDOS. All of these, except the optional six-partition capability and spooling, are standard and available to the user with the basic 2K-byte increment to DOS.

Essentially, EDOS is run under user control with specific parameters that activate the system components. These parameters can be entered at Sysgen time or in the job stream at run time, depending upon which EDOS feature is to be activated.

With none of the EDOS facilities activated, the user merely runs under DOS Release 26 without enhancement (except

➤

EDOS
The Computer Company

about 80 percent of the current EDOS users are running extended-partition processing (4, 5, or 6 partitions).

EDOS offers a number of other useful facilities, all of which are discussed in detail in the Characteristics section of this report:

- "Intelligent" Procedure Library Support
- Resident Transient Routines
- F-Level Linkage Editor
- Modified Storage Dump routine
- Blocked Fetch
- High-Speed Volume Dump/Restore
- Documentation Aids Program

Planned extensions to EDOS include:

- A Change Priority Command (CHAP), allowing the operator to dynamically alter partition priorities as desired.
- Additional Sysres device support, removing the IBM 2314 Direct Access Storage Facility requirement from EDOS and allowing the system to reside on a 2311 or 3330.
- Extended System/370 support for newer peripheral devices, etc.

Many DOS users quite naturally will wonder just how safe it is to do business with anyone other than IBM for operating system segments that are so intimately concerned with the "heart" of their computer system. And this is a valid question. Of the EDOS users and trial installations contacted by Datapro, two-thirds of them reported that one or more problems had been encountered in DOS Release 26 from IBM since EDOS was installed. In each of these cases, IBM did some rather remarkable foot-dragging for a company with its huge resources before applying a fix, and in one case finally offered to fix the problem at a charge. Turning then to TCC, the EDOS users received program fixes and such other assistance as they needed to correct IBM's DOS errors in what amounted to as good or better response time than could otherwise have been hoped for from IBM.

Indeed, TCC openly advertises that it would like to become known as "the firm that took over 360 DOS maintenance." And the designers of EDOS present a remarkable variety of credentials, largely from prior experience at IBM, where current TCC staff members made significant contributions to the development of IBM's Compatibility Operating System (COS), OS, POWER with Remote Job Entry (RJE), Data Base Support for APL, etc.

It should be noted that with the "stabilization" of System/360 DOS at Release 26 by IBM, Class A (no-charge maintenance) status will be removed as of March 31, 1973. Thereupon, problems with IBM such as those currently encountered by EDOS users as a "penalty" for dealing with a non-IBM software source and/or refusing to meekly cooperate with IBM's "migration" plan will probably be encountered by *all* System/360 DOS users as part for the presence of the unused EDOS modules in the supervisor and on the system residence library).

The capabilities of EDOS are discussed individually in the following paragraphs.

1. Extended Partition Support provides up to six partitions (up to three are provided by DOS plus up to three more by (EDOS). This optional feature (System "B") requires 2K bytes in addition to the basic 2K bytes already added to the size of the supervisor by EDOS. (Other EDOS enhancements and features do not add resident memory requirements to the supervisor.) System "B" requires a 14K-byte partition for a cold-start IPL, or for de-allocation.

2. "Intelligent" Procedure Library Support permits conditional execution of cataloged procedures (JCL streams). IBM's POWER spooling supplement to DOS also provides a basic procedure library facility, but one that contains only a straight listing of the JCL cards. The EDOS procedure library facility goes quite a bit further. Uder EDOS, the cataloged job streams can logically be entered at a number of points during processing to dynamically alter the job execution logic based either upon problem program conditions experienced during processing (completion or ABEND), or upon user selection of system options through SYSPARM'S or UPSI's (User Program Sense Indicators). Parameter substitution enables procedures to be tailored to meet individual program requirements. Up to 15 levels of procedure statement nesting can also be supported.

3. A Relocatable Loader is provided for user program relocatability. Although a number of such routines are available from other sources, this is the only one that is an integral part of the user's operating system. The relocatable loader extends full support to programs in all high-level procedural languages and in BAL, including support for programs written with an overlay structure.

4. A Resident Transient Routines facility extends support to ISAM or BSAM for disk or tape files by providing the transients in relocatable form so that they can be link-edited into the individual problem-program partitions. It should be noted that the DOS transients are the routines that perform I/O commands, and having them in the problem-program partition greatly reduces the fetch time required and enhances the overall performance of the system. In fact, the processing time for most ISAM files can be cut in half, according to actual user experience, by virtue of this EDOS feature.

5. An F-level (44K-byte) Linkage Editor is provided as a replacement to IBM's standard 10K-byte linkage editor. The EDOS linkage editor is about twice as fast as the IBM routine. Because of the comparatively large memory requirement for the F-level EDOS editor, however, the standard IBM editor is automatically invoked by EDOS when less than 44K bytes of core is available.

6. Automatic Volume Sensing (AVS) is provided as an operator aid to assist in mounting tape reels and disk packs. This operation support feature relieves the operator of the need to specifically assign logical tape or disk volumes to physical drives; the EDOS system accepts the logical identifiers and resolves the actual system assignments.

EDOS
The Computer Company

of IBM's campaign to upgrade them to a System/370 and virtual storage.

As for the current users of EDOS, they represent a very educated, sophisticated class of computer user. Virtually all of them have already been weaned of total dependence upon IBM; third-party lease agreements and plug-to-plug peripheral replacements are liberally sprinkled among them. In fact, a rather typical EDOS user with a 512K-byte 360/50 is doing business directly with IBM for his keypunches only. Few others do not have at least some non-IBM tape or disk drives, at least.

Into this cost-conscious market, EDOS has been received with an immediate, enthusiastic response. Datapro's survey of users revealed that TCC has established a top rating for credibility and quality of service. Virtually all claims made by the vendor have been satisfactorily fulfilled, including fast turnaround of maintenance requests. One user reported that the reduction in overtime payments to IBM for his computer system has paid for the cost of EDOS, and he is able to handle a larger workload in addition.

Also of major significance in current users' decisions to use EDOS is the fact that EDOS can defer or indefinitely postpone an upgrade to OS, to a larger System/360 under DOS or OS, or to a System/370 under DOS or OS (or their VS counterparts) on the grounds that the larger system is required to handle an increased workload. For a typical 360/40, an upgrade from DOS to OS is 66K bytes more "expensive" (in terms of software residency requirements) than an upgrade to EDOS; and for a typical 370/145, the DOS to OS upgrade is 106K bytes more "expensive" than the EDOS upgrade. These differences translate into about $1,600 and $1,900 per month, respectively. Thus, a potential DOS upgrade user who does not require all of the fancy aspects of OS or VS (TSO, HASP, etc.), but who needs greater system throughput (typically up to about 30 percent more), is well advised to examine EDOS carefully and give serious consideration to its use. Likewise, many current DOS users who are not imminently ready to upgrade to OS or to a 370, but who are nevertheless paying overtime charges on their present systems, can supplement DOS Release 26 with EDOS and save the monthly cost of the EDOS system or more, while receiving other benefits in terms of more effective, streamlined operations.

One additional factor deserves consideration by users who want to prolong the lives of their System/360 DOS installations. While the problems resulting from IBM's dropping of System/360 DOS support can well be overcome by EDOS (as they indeed already are being overcome), and the raw processing capability of the System/360 hardware may be fully adequate to handle the user's workload for an indefinite period of time, the question of support for language processors and utility routines must be raised. Currently, for example, the Data Base Task Group of the CODASYL Committee is pushing for significant extensions to high-level languages, including "stable" languages such as COBOL. At such time as major enhancements are made to these languages, many System/360 users will naturally want to take advantage of these advances—and it may well be that such progress

7. Load Balancing is provided to dynamically alter partition priorities based upon the frequency of I/O interrupts in the respective partitions. This scheme automatically distributes system resources more efficiently between I/O-bound and CPU-bound programs running simultaneously in different partitions. A number of independent DOS supplements are available that perform this function—among them some of the spoolers—but EDOS once again provides the only load balancer that is an integral part of the user's operating system.

8. Improved Job Accounting facilities include a report generator for output reports. The EDOS accounting routine also includes dynamic disk extent switching to permit the accounting statistics to be dumped for reports without stopping the entire system and thereby actually losing some accounting data. Unaided DOS provides raw accounting data only, although there are a number of independent DOS supplements that do produce formatted output accounting reports. The EDOS Job Accounting Routines require partition sizes of 14K or 24K bytes to dump/swap or summarize the output report(s), respectively.

9. Modified Storage Dump facility permits the printing of supervisor storage to be bypassed. Such printing is normally quite time-consuming, and the time required to produce dumps can be reduced up to 35 to 40 percent without the supervisor dump, depending upon the size of the supervisor and user program(s).

10. Blocked Fetch is provided to shorten the load time required to fetch program modules from the core image library. Blocked Fetch can run up to eight times as fast as the single-program fetch used in unaided DOS, with typical speed-up factors of two to three times reported.

11. An F-level (44K-byte) Volume Dump-Restore is provided as a significant improvement over the standard IBM dump-restore. Where the IBM system takes 27 to 31 minutes to dump the contents of a 2316 Disk Pack to tape (depending upon the speed of the tapes and the availability of main memory for blocking), EDOS performs the same function in 4 or 5 minutes. The performance of the Dump segment is significantly better if a partition size of up to 80K bytes is available.

12. Source Library Support is provided for selective retrieval of any portion of any source statement book regardless of the programming language involved. Standard DOS provides a source library support facility for BAL, and a limited facility for the Procedure Division of COBOL programs only. Once again, independent sources are available for this capability, but there is a great price differential in favor of EDOS—as well as the fact that the EDOS facility is integrated into the operating system. An EDOS utility program, COMPILE, is available to extend source library support to individual language processors and provide optional backup for the user programs. COMPILE requries a partition of 38K or 14K bytes to initialize and control the extension of support or to update and retrieve the user programs, respectively.

13. An F-level (44K-byte) Documentation Aids Program (DOCUMENT) is provided to produce additional sets of updated documentation right on the user's own system. While a certain processor expense is thereby

▷(supported by IBM) can come about only with an upgrade to VS on a System/370. Of course, such a "product gap" would likely create a new market need that might be quickly filled by another independent software firm (or perhaps by TCC itself).

On the other hand, no matter what the future holds for System/360 DOS users, there is always OS, DOS/VS, OS/VS, and the System/370. If EDOS can be used at this time, it will produce significant benefits and savings for as long as it is in operation. These savings will be cumulative until (if ever) an upgrade to the System/370 and VS is made. □

➤ incurred, most users find that the timeliness and ready availability of the documentation is worth the cost. TCC keeps the documentation program updated. Document can run in a partition of only 20K bytes if no cross-reference (index) to the documentation is called for by the user.

14. An optional Extended Spooling Facility (ESF) is available with an ancillary ESF Utility program. TCC's design of ESF places the basic input data reading ability for a program in the supervisor. This basic capability will accept input and place it on a disk queue in an unstructured way. The ESF Utility program operates from a separate partition to form an input queue, handle operator commands, and manage the actual transfer of output data to physical card punch or printer devices. The ESF Utility does not have to be present in main memory in order for problem program processing to take place, but will have to be brought in to unload the output queues. An early output start is also available. Efficient disk I/O buffer area management that shares buffer areas between programs permits reduced disk queue area requirements.

In addition to the above features, all standard DOS services and facilities are available through the basic DOS Release 26 from IBM.

PERFORMANCE: Although some of the modules that comprise EDOS have been commercially available for as long as two years (e.g. the accounting package), full EDOS has been in use for only a comparatively short time. Certain performance improvements, however, have already been quantified by current users in response to a Datapro survey:

- Cataloging procedure time has been cut by 50%, based primarily upon the EDOS F-Level Linkage Editor. (A full Sysgen under EDOS, including re-assembly of the Supervisor and a link-edit of both DOS and EDOS including 6-partition support has taken as little as 70 minutes.)
- Dump-Restore time has been cut by as much as 75% with EDOS.
- ISAM file processing time has been cut about 50%, due primarily to the EDOS Resident Transient routines feature.
- Object programs have been loaded into main memory for excution by the Blocked Fetch Feature in as little as 1/8 the time otherwise required, with typical user savings of 1/2 to 2/3 the time otherwise required.
- In addition to as much as a 25% improvement in throughput available from Basic EDOS plus six-partition support, the Extended Spooling Facility (ESF) can provide an additional 15 to 20% throughput improvement for an overall gain of up to 45%.

HARDWARE/SOFTWARE REQUIREMENTS: An operational version of IBM's DOS Release 26 is required on an IBM/System 360 or 370 computer. EDOS expands the size of the DOS Release 26 supervisor by 2K or 4K bytes, depending upon whether extended (six) partition support is included in the system. The extended spooling facility adds from 2100 to a maximum of about 8100 bytes to the operating system, with 2500 bytes being typical. Even complex spooling requirements, however, seldom exceed about 4K bytes of additional memory requirement. Because the system size must be increased in 2K-byte increments, this means that ESF can typically add 2K to 4K bytes, depending upon how close the supervisor already is to a 2K-byte boundary. The ESF Utility (run in a separate partition to formulate the input queue or handle the actual transfer of output data to physical devices) requires from 16KB to 25KB, with 16KB sufficient for a medium-size environment. (Note that the ESF Utility need not be present during basic input reading or during batch processing of already-queued input data sets.)

PRICING: EDOS is available on a paid-up lease basis for a one-time charge, on a monthly license basis concellable upon a 30-day written notification, or on an annual lease basis. Maintenance is included in all leases at no additional charge, and for at least two years on a "purchased" system at no charge; thereafter a maintenance contract will be negotiated. A pre-installation testing period of 60 days is provided at no charge. Should the user not want EDOS by the end of the 60-day trial period, the system is returned to TCC with a $50 handling charge. Installation assistance is available at a charge of $200 per day plus expenses, although most users will be able to install the system without assistance.

	One-Time Charge*	Monthly Lease**	Annual Lease**
Basic EDOS, Release 5.0 (includes all features except six-partition support and spooling)	$6,750	$225	$2,400
Each additional copy of Basic EDOS	2,250	75	800
Six-partition Support Feature (SPSF)	2,250	75	800
Each additional copy of SPSF	750	25	270
Extended Spooling Facility (ESF)	6,000	200	2,100
Each additional copy of ESF	2,000	70	740

* Includes maintenance for a minimum period of two years.

**Includes maintenance for the life of the contract; 75 percent of lease payments up to $2,000 may be applied toward the one-time charge.

INITIAL DELIVERY: April 1972.

CURRENT USERS: As of November 1972, 25 installations on lease and 20 more on the 60-day test basis. ■

TASK/MASTER
Turnkey Systems, Inc.

MANAGEMENT SUMMARY

Standard operating system facilities provided for the central computer often require extensive interfacing in order to be operated from remote on-line stations. These communications-oriented considerations include system start-up and cycle-down, file and communications access method interfaces, system recovery procedures, security measures, etc.

It is commonly desirable for data communications to be handled as one of a number of jobs under control of the computer's operating system instead of dedicating an entire processor to controlling the multi-station network. As the number of terminals and concurrent tasks in a communications network increase, the requirement for allocating and accounting for the system resources demanded by each user program becomes very complex. Solutions to resource contention depend upon the application of highly sophisticated programming techniques. These include the establishment and maintenance of message queues, assignment of task priorities, program swapping into and out of main memory, sharing of common code through program re-entrancy, etc. Furthermore, changes to the configuration or applications programs must be possible without major reprogramming.

For users of System/360 or 370 computers under DOS, OS, or VS, Turnkey Systems provides answers to all these problems with TASK/MASTER. Basic features common to all TASK/MASTER versions include:

- Password protection for individual programs and the data base(s), as well as restricted network access.
- Start-up and cycle-down facilities to control the data communications network.
- Device independence using a library of Terminal Independent Modules (TIM's) provided to handle virtually any type of terminal.
- Stow/Find facilities to set certain main memory areas aside to facilitate development of reusable and reentrant programs for multi-thread operation. Stow/Find also eases the process of transferring data between segmented programs.
- Simultaneous update protection at the individual record level when a read request with "intent to update" is issued.
- System accounting and system resource utilization statistics.

▷

This high-quality data communications monitor handles network administration functions, thereby isolating user programs from these considerations. TASK/MASTER runs in one partition or region of current IBM System/360 or 370 operating systems, and allows user programs to be written in COBOL, FORTRAN, PL/1, or BAL.

CHARACTERISTICS

SUPPLIER: Turnkey Systems, Inc., 111 East Avenue, Norwalk, Connecticut 06851. Telephone (203) 853-2884.

TASK/MASTER is also available through Periphonics Corporation, Airport International Plaza, Bohemia, New York 11716, telephone (516) 567-1000, in conjunction with its front-end processor and voice response equipment. See Report 70G-420-01.

In Europe, contact Hoskyns Systems, Ltd., 91-93 Farringdon Road, London EC1N3LB, England; telephone 01-242-1951; or TESCI SOFTWARE, 56 Rue la Boetie, Paris 75008, France, telephone 225-8683.

BASIC FUNCTION: TASK/MASTER is a general-purpose data communications monitor that operates in a partition or region of an IBM System/360 or 370 under DOS, DOS/VS, OS/MFT, OS/MVT, OS/VS1, or OS/VS2 to control multiple on-line user terminals and applications. It consolidates all of the communication interfaces and I/O and control functions required in the data communications network and isolates users' application programs from the communications environment. TASK/MASTER also supports COBOL, PL/1, etc., simplifying on-line applications development. Users can modify the operation of TASK/MASTER by altering selected control parameters, and the system provides extensive debug and restart facilities. It also interfaces data base systems.

OPERATION: As the interface between user-written applications programs, the operating system, and communications requirements, TASK/MASTER normally operates in a high-priority partition or region that contains four major functional elements of TASK/MASTER, with standard interface areas for data transfer.

- The Control Program sets up the user job processing queues, including all necessary tables and indicators, Application program service requests are processed by the Control Program, and the data necessary to service them is distributed as required to other functional areas. The Control Program scheduling algorithm handles relative priority assignments specified by the user.
- The Phase Loader interfaces user-program processing requests to the Control Program and passes control to specific user programs in main memory. If a user program that is to have control is not in main memory, the Phase Loader initiates the loading of that program if enough contiguous space is available. Otherwise, the load request is queued in an internal

➤

TASK/MASTER
Turnkey Systems, Inc.

Optional TASK/MASTER facilities include:

- Multi-tasking within the TASK/MASTER partition or region for concurrent processing of multiple applications programs. Control is passed between programs when an I/O request is initiated.
- Support of virtual operation under DOS/VS, OS/VS1, or OS/VS2.
- Message switching or "broadcasting", as well as queues to hold individual terminal message traffic.
- Error recovery and warm-restart facilities that continuously checkpoint all program operations.

TASK/MASTER is a highly flexible system. It can be customized at installation time, and the user himself is free to customize and/or modify nearly all of it to suit his purposes through a GENESIS process using easily altered parameters. TSI provides complete installation and education support at no extra charge.

Moreover, TASK/MASTER is a popular system. TSI estimates that in new system sales, TASK/MASTER commands 25% of the market, against about 65% for IBM's CICS (see Report 70E-491-02) and 10% for all other offerings of this type. TSI also claims significant numbers of "rebound" sales from former CICS users, on the basis of TASK/MASTER's power, flexibility, extreme ease of installation, and low core usage (typically about 20 to 50% less than CICS). What's more, TASK/MASTER can be purchased outright, or leased at a lower cost than CICS, while also reducing hardware costs.

Under TASK/MASTER, individual application programs can be written in high-level languages (e.g., COBOL, FORTRAN, PL/1, and BAL). Use of these languages plus system control features built in to Task/Master reduces the time and expense of developing on-line systems and can make on-line applications programs no more complex than programs for a batch environment.

A standard interface is provided for any language that can be link-edited into the operating system. This promotes the use of high-level languages and system independence for data communications applications by isolating user application programming from the communications control functions. Therefore, conversions to different operating systems or computer systems are much easier than would be the case where the individual applications carry much of the burden of the communications environment.

Turnkey Systems currently supplies TASK/MASTER interfaces for most of the proven data base systems (e.g., TOTAL and DBOMP), has recently announced an interface for IBM's DL/1, and states that it will commit to

table until sufficient room is available. The Phase Loader is also responsible for managing main memory allocation to individual user programs.

- The I/O Monitor and various file access methods handle all file I/O for the user programs, thereby eliminating the need for individual user programs to contain any I/O instructions.
- The telecommunications interface to all standard IBM telecommunications access methods, such as GAM, BTAM, etc., handles all terminal data communications requirements, thus isolating the user programs from the communications environment. Control Program requests for open, close, read, and/or write, as well as completion indicators for these actions, are processed through the Task/Master Network Supervisor. TASK/MASTER will support VTAM when it becomes available.

Also operating in the TASK/MASTER partition or region are the user's programs, written in COBOL, PL/1, FORTRAN, or BAL, as well as a number of transient routines for use during specific operational phases such as system initialization, cycle-down, etc.

Specific services provided by TASK/MASTER include:

- Start-up/cycle-down and sign-on/sign-off. Password protection is provided to control access to the on-line applications. After all terminal activity is stopped and the system files are closed, statistical information can be displayed.
- Task initiation to load and transfer control to user programs based upon terminal messages or internal processing requests.
- I/O queues and an unsolicited message queue to store data transmission and message traffic for the terminals. Main memory and/or direct-access storage devices are used to hold the queues of user-specified length. Point-to-point or broadcast-type message switching facilities are provided. The queues are protected against system failure.
- Imbedded multi-tasking support (contained within TASK/MASTER and independent of the IBM operating system type or version) that transfers control between programs when the currently active program initiates an I/O request. A multitasking supervisor is not required to implement multitasking under TASK/MASTER.
- Multi-threading, supported by the TASK/MASTER storage management routines, in which a single copy of the "pure" or unaltered portions of a re-entrant program is successively used by multiple users. The altered portion of the user program is "stowed" and later "found" when the same user resumes control. This technique can not only conserve main storage requirements, but can also reduce swap-time overhead. Stow/Find uses a collectable pool, thus eliminating core fragmentation.
- Debug facilities, including optional trapping of program checks and program loops, a user-specified n-step trace facility, etc. TASK/MASTER also provides an optional Warm Restart facility. This capability can be activated or suspended by individual jobs or selected terminals. The Warm Restart module, if it is to be

interface with any data base software the user may have, often at no additional charge.

USER REACTION

TASK/MASTER users extoll the comparatively low cost and high performance of the package, and also praise the ease with which it can be used. It is also cited for being easy to install, having reasonable core requirements, and being terminal-independent. The users interviewed were well qualified to be critical, and one in particular showed that he knows how to stretch his EDP dollars; besides subscribing to DATAPRO 70, he has installed independent main memory on his IBM computer and obtained a replacement operating system for IBM's DOS.

Any user seriously considering a data communications monitor should look carefully before he leaps. The in-house solution is one which should not be undertaken lightly; it involves highly sophisticated skills and tends to result in a special-purpose design that will not be readily adaptable to future configuration changes. Most good communications monitor vendors are happy to expose their systems to good prospects, and Turnkey simplifies the user's evaluation through a continuing tour of no-charge seminars in U.S. cities. These seminars provide a good opportunity to view TASK/MASTER at first hand.□

used, must be included at GENESIS time in the user's version of TASK/MASTER.

- Administrative services that include a TASK/MASTER system log, as well as a complete statistical accounting of system use with on-line availability to the terminals.
- Terminal Independent Modules (TIM's) to isolate application programs from the terminal hardware, especially where the terminals require extensive use of control characters imbedded in the messages. For example, TIM's make all device control characters for complex CRT's (such as the IBM 3270) transparent to the application programmer.

TASK/MASTER is normally provided on magnetic tape in source language. A customizing routine, GENESIS, responds to user parameters by providing a JCL stream and a tailored version to fit user requirements, including the I/O environment, optional usage statistics, password initialization, file access methods, message queue lengths, etc. Typically, 1/4 to 1/3 of the approximately 37,000 source statements provided on the master tape are incorporated into the version for a particular installation. A full GENESIS run takes about two hours to complete, with subsequent TASK/MASTER modifications handled in much shorter runs.

Also provided with TASK/MASTER is a stand-alone system response time simulator, written in FORTRAN, and a standalone application test facility.

HARDWARE/SOFTWARE REQUIREMENTS: The stand-alone GENESIS run to produce a customized version of TASK/MASTER requires a 64K-byte IBM System/360 or 370. TASK/MASTER itself runs on an IBM System/360 or 370 under DOS, DOS/VS, OS/MVT, OS/MFT, OS/VS1, or OS/VS2. A typical medium-size TASK/MASTER network, supporting 3 concurrent tasks on 30 terminals of the same type, requires a partition size of about 75K bytes. A very large version of TASK/MASTER, supporting 8 concurrent tasks on 100 terminals of 3 different types, will typically require a partition or region size of about 170K bytes. In all cases, the specific main memory requirements for TASK/MASTER depend upon the options included during GENESIS.

PRICING: The full TASK/MASTER system can be purchased for a single payment of $25,000 or purchased on a 3-year accrued equity plan for $850 per month. Alternatively, various versions can be leased for $200 to $575 per month plus a $4,500 installation fee. The installation fee is reduced by $1,500 for each year of lease. Thus, the installation fee is $3,000 for a 1-year lease or $1,500 for a 2-year lease; no installation fee is charged for leases of 3 years or more, or for purchased systems.

Lease prices include maintenance and updates. For purchased systems, a maintenance charge of $100 per month is imposed after the first 3 years. A purchase option allows 50% of all lease payments to be applied toward purchase.

Purchase entitles the user to one copy of the system, and he may use it as desired, except that it may not be duplicated. Unlimited use is allowed at one location; additional locations, including full support allowances, are charged for at 70% of the first-copy prices.

Installation of the system includes two weeks of on-site support and requires some test time on the computer system. Average installation of a full system takes about eight days, with the rest of the support allowance credited to the user for future use. Additional custom service beyond the standard support period and after the system is installed is available at a per diem rate of $225/day plus travel.

INITIAL DELIVERY: November 1970.

CURRENT USERS: 75 as of September 1973. ■

UCB Payroll System
United California Bank

MANAGEMENT SUMMARY

United California Bank is the second largest processor of customer payrolls in the United States and has been engaged in this activity for about 12 years. To service this activity, the bank has produced and refined a general-purpose payroll package. The bank is marketing this package to other banks, service bureaus, and large companies. To date, the buyers of the package have been other banks.

The features of this payroll system are tailored to the customer service function. The most striking of the features are the method of input for payroll information and the custom output.

Before commenting on these features, clarification of the differences between the customer payroll service the bank offers and the package itself are necessary.

As UCB's payroll service is structured, the bank offers complete payroll preparation and related functions including acceptance of input, keyprocessing of input, generation of checks and reports, delivery of checks and reports, automatic deposit of employee's pay (in UCB, of course), and check reconciliation.

The package itself accepts the input, computes the pay, taking into account taxes and deductions, and outputs the paychecks along with reports.

The payroll is designed around a concept that could be called payroll by exception. Every active employee gets a pay based on information held in the master file unless an exception card is received; exceptions can be pay rate or units, pay structure, or deduction changes, and can be temporary or permanent.

The data for all companies is held on a single master file. Input is batched and accumulated throughout the day. The master file is updated in one pass. The output of this pass is another file containing all data for all reports, including checks. A separate module breaks this file down into individual files for each type of report. Customized forms are used for all reports, and the same report for all companies is printed at the same time.

The Labor Distribution Reporting function and the Employee Quarterly History Report are extra-cost options. All labor distribution reporting is done on the basis of an 18-digit labor code. All pay information is totaled for each employee and listed in data sequence. Both options have extensive flexibility to satisfy the reporting requirements of different customers. ▷

This payroll package, which runs on an IBM System/360 or 370, incorporates elements of other proprietary packages for tax specifications and report generation. It is designed for banks and service bureaus but can also be used by large companies. The package features ease of input for customers and flexible output reports.

CHARACTERISTICS

SUPPLIER: United California Bank, Business Services Division, 400 East Live Oak, Arcadia, California 91006. Telephone (213) 445-4660.

BASIC FUNCTION: The UCB Payroll System is a complete payroll package designed to be used in a service bureau for multiple clients, but it can also be used by a large company in-house. For the service bureau, it features ease of input by customers. Extensive reports can be prepared.

OPERATION: The system consists of five modules: Edit and Capture, File Maintenance/Posting, Checks/Report Generator, Labor Reporting, and Employee Quarterly History Reporting.

All input to the system is made through the Edit and Capture module. The input is checked for validity and completeness. Inputs are accepted in batches, with batch total checking. The inputs are accumulated on tape in input order for later processing.

At the cutoff time, the input tape is fed into the File Maintenance/Posting module, where it is sorted to disc. All records are kept on one master file by company. All updating and file maintenance is processed in one pass of the master file. The outputs of this module are a sorted transaction file on tape for historical purposes, a Labor Transaction File on tape for labor reporting (optional), a Reports File on magnetic tape for all report writing, and an audit listing of all changes.

The Checks/Report Generator includes the option to automatically stack report files on tape and disk, according to the user's processing requirements. Each of the stacked files contains report information for each company represented in the run. Each report is run individually for all companies at the same time to minimize the operator intervention required to mount the special preprinted forms for each different type of report. In this context, the paychecks are one type of report.

The whole system consists of about 30 programs and 4 sorts. Most of the programs are used to prepare the individual reports after the necessary information has been extracted by the Report Separator routine within the Report Generator module. These programs are individually written COBOL routines, except for the check writing function, which is done in assembly language. ►

UCB Payroll System
United California Bank

▷ DATAPRO 70 contacted the users of UCB's Payroll System, and they were all very satisfied with the package. UCB was given a high rating for professionalism, and users reported that the company has stood firmly behind the package with excellent maintenance support. □

➤ Calculation of taxes is performed with the assistance of the ALLTAX package from Management Information Service. Under the terms of an agreement, users of the UCB Payroll System become subscribers to the ALLTAX service and receive periodic updates; the service covers all domestic taxes.

The Labor Distribution module is optional. Typically, the Labor Transaction File is broken down into areas, with a change listing and distribution report for each company for each area. Reporting can be to job, department, or cost center. Period-to-date and year-to-date accumulations can be shown, and budgetary allocations can be shown for comparison purposes. Several levels of summaries based on an 18-digit labor code can be made.

MASTER FILE: The master file contains information on each company and employee. The file contains seven different types of records and is organized by employee number within company. A single 412-character record leads off each series of records for a different company. A 62-character header record is provided for each subcompany. The remaining five types of records are used to describe the employee, his pay rate, deductions, and earnings, as well as accumulated totals.

The five records are Employee (143 bytes), Pay (53 bytes), Deduction (55 bytes), Total (234 bytes), and State Change (74 bytes). Individual Pay and Deduction records (up to 100 of each type) are carried for each different pay rate and deduction.

INPUT: Input to the processing system is through keypunched records. Input to keypunching is through a series of special forms. A normal byproduct of the processing run is a series of pay cards and deduction cards. These cards are slightly larger than a standard punched card for convenience of handling. If any changes have occurred in an employee's pay status (including termination) or deductions, the appropriate card is edited and sent to the computer center. Otherwise, the pay is automatically calculated and a check written as before. New cards are printed out each cycle, reflecting all changes. Another two-part form is available if a daily recording of labor is needed at the customer's location.

OUTPUT: There are four principal types of output: audit trails, checks, management reports, and labor distribution reports.

Listings and registers are provided to show exactly the records input to the system and the disposition of each item.

The employee checks contain extensive summary information on the stub, with year-to-date totals for gross pay, taxes, and deductions.

Management reports can be custom-tailored to fit the needs of the individual customer, but in general they are the same for all companies on the master file.

Labor distribution reporting is highly flexible and can be tailored to fit the needs of individual clients. In general, direct and indirect labor expenses can be allocated to job, department, or cost center.

PERFORMANCE: The performance figures cited by UCB for its internal operation are impressive. The Business Services Division prepares about 7,000 payrolls a month covering some 250,000 employees and involving 1,000,000 records. In 1971, over one billion dollars in payroll checks was issued by the Business Services Division. The main posting run requires about 75 to 90 minutes of computer time per day on a 512K System/360 Model 50. As many as 500 payrolls have been prepared in a single shift. Typically, UCB promises two-day turnaround from the time that input is received from the client until the checks are delivered to the client.

HARDWARE/SOFTWARE REQUIREMENTS: The minimum configuration consists of an IBM System/360 Model 50 with 512K bytes of core storage, four tape drives, one disk drive, one card reader/punch, and one printer, operating under OS/MFT or MVT. Multiple printers or off-line printer stations can be accommodated to speed the printing of reports.

PRICING: The basic Payroll System is sold for $20,000. This price includes a related 3-program Payroll Scheduler System. The Labor Distribution module costs an extra $3,000. The sales agreement includes a six-month warranty along with continual support of all tax specifications (ALLTAX).

For customers who are going into the payroll service business, the publication rights to marketing material developed by United California Bank for the Payroll System are available for a flat fee of $2,000. A customized color movie is also available.

SUPPORT: Three man-weeks of effort for the installation and training are included in the sales agreement. This is normally furnished by three UCB staff members conducting a one-week training course in the specific areas of sales, operations, and systems. UCB will also customize an interface between the Payroll System and a purchasing bank's demand deposit accounting and deposit programs.

INITIAL DELIVERY: The Payroll System has been in use within the UCB Business Services Division since April 1968. The first delivery to an outside company was in November 1969.

CURRENT USERS: There were 4 users of the UCB Payroll System, all banks, as of November 1971. (This does not include the two internal computer centers of UCB.) ■

DOS ASAP
Universal Software, Inc.

MANAGEMENT SUMMARY

ASAP is one of a growing number of IBM System/360/370 DOS spooling supplements that improve the throughput and greatly extend the power of DOS. USI claims a typical throughput improvement of 20 to 40 percent over unaided DOS and 10 to 15 percent over IBM's own POWER spooling supplement. This enhanced DOS operation often results in reduced expenditures for additional equipment of overtime charges, and may permit postponement of a transition from DOS to OS on the grounds that a more powerful operating system is required to handle a heavier work-load.

As is the case with most spooling supplements, ASAP is intended to be installed by the user with minimal assistance required from the software vendor. Toward this end, USI provides a clear, easy-to-understand set of documentation that indicates the correct procedures to follow in installing ASAP.

In the highly competitive spooling marketplace, ASAP finds itself in competition with other spoolers about 75% of the time. Most of these situations involve IBM's no-charge POWER, but still others match ASAP against Software Design's GRASP (Report 70E-760-01), Boothe's SPOOLER (Report 70E-100-01), etc. ASAP's competitive position against POWER is substantially the same as that of most other serious spooling contenders: improved performance with as little as 20 percent of POWER's storage requirements.

Spooling stands for Simultaneous Peripheral Operations On-Line. A fundamental difference between POWER's spooling and that of most its independent competitors, such as DOS ASAP, is that POWER is a synchronous spooler while most of the others are asynchronous spoolers. That is, although a job under POWER "spools" printer output to a disk, the disk-to-printer output of the job doesn't begin until the job has been completely processed by the CPU. (POWER, however, remains the only available spooler with automatic scheduling facilities.)

With an asynchronous spooler, on the other hand, the printer can be fed from the disk as soon as the disk receives the first record. Naturally, the printer will lag somewhat behind the disk, but the significant point is that the printing operation can begin right away. This type of spooling also saves disk space, since the disk doesn't need to hold the entire output file destined for the printer. What's done is to allow less disk space and wraparound queue the output records.

Since the initial Datapro report on DOS ASAP, several enhancements have been made (or at least announced) for the package. These are: remote terminal communications capability (now in final test stages and scheduled for release in the third quarter of 1973); retrieval of cataloged procedures from the system's source program library (also in final test for scheduled fourth quarter 1973 release); simultaneous tape/printer output (tape-only output is ▷

This user-proven spooling supplement for IBM's System/360/370 DOS increases system throughput by making it possible to perform multiple media conversion operations simultaneously with up to three independent user programs. ASAP is available in either an input/output or output-only spooling version.

CHARACTERISTICS

SUPPLIER: Universal Software, Inc., Commerce Park, Danbury, Connecticut 06810. Telephone (203) 792-5100.

FUNCTION: ASAP (Automatic Spooling with Asynchronous Processing) is a spooling extension to IBM's Disk Operating System Supervisor. It can provide three-partition spooling for all 360/370 DOS installations. It can also provide spooling in a single-partition non-multiprogramming DOS system even if the Storage Protection feature is not present.

Concurrent spooling is provided for any number of standard IBM card readers, punches, and printers, as well as for "pseudo" or imaginary devices which need not be either physically attached to the system or available to a specific partition. Input job streams are processed by ASAP and may be placed in intermediate disk buffers prior to loading into main memory for execution. Output files begin direct media conversion to cards or paper, using intermediate disk and main memory buffering for that portion of the output which exceeds the on-line capacity of the card punch(es) or line printer(s).

OPERATION: ASAP requires that 1 macro be incorporated into the DOS Supervisor for a typical total of about 170 to 500 bytes of additional code. The Supervisor is then reassembled with the ASAP macro included and catalogued. Invocation or termination of ASAP operation is done from the operator's console or by control cards in the input job stream. The ASAP Supervisor macro provides linkage for all I/O requests and interrupts to ASAP, which resides in from less than 4K to as much as 12K bytes, either in upper core above the highest allocated partition, or between partitions under certain special circumstances. The latter case arises, for instance, in systems where CS/40 is used, and certain partition boundary alignment requirements cause unused areas in main memory to be set apart which can contain the ASAP System. Using this design, the ASAP supervisor macro does not stretch the size of DOS to a point where major readjustment to established linkages is required, nor does ASAP itself consume a user partition in main memory. Thus, the user can run his programs in 3 partitions.

ASAP is available in either of two versions: the Reader/Writer version to handle both input and output spooling, and the Writer version to handle output only. The available features of ASAP include the following:

- Early printer start, enabling output to be printed while the job creating the output is still running.
- Two-level output buffering to both disk and main memory, enabling the printer(s) or card punch(es) to be driven from memory while additional output is buffered to a disk.
- No changes required to previously set up JCL for input job streams.
- Multiple print and card file support, with more than one printer and/or card punch driven at the same time.
- Support of IBM 1400 emulation.
- Automatic restart.
- Ability to support three user partitions, including teleprocessing in a foreground partition, if desired, as well as running under a non-multiprogramming system. ►

▻ scheduled for the fourth quarter of 1973); user program relocatability (which will work well with USI's RELO-PLUS described in the following report, 70E-879-02); job accounting; and multiple output copies.

USER REACTION

DOS ASAP isn't the leading spooler from an independent vendor; GRASP is. Thus the questions most asked about the USI package are "Does it run as well as GRASP?" and "How's the vendor support?" Responses to Datapro's user survey and interviews with other users provided positive answers to both questions.

Users interested in technical detail should note that both ASAP and GRASP provide full 3-partition spooling, but in different ways. (POWER, by the way, must reside in a DOS partition itself and can only spool the remaining partitions.) ASAP resides in high core (or, optionally in multiprogramming systems, in interpartition core areas) and adds a small linkage module to the IBM DOS supervisor. It can thus provide spooling for all IBM DOS partitions. GRASP, conversely, does not add to the IBM DOS supervisor, but resides in a partition, like POWER. Optionally, GRASP can reside in a specially-created partition, called FO, and thereby provide full 3-partition spooling.

A representative ASAP user interviewed by Datapro has an IBM System/370 Model 135 running three spooled DOS Release 27 partitions. This user's expanded ASAP core requirements are 8K bytes plus 500 additional bytes in the IBM DOS supervisor. This user rejected GRASP as too expensive (extra cost to install each additional "psuedo," or "imaginary" device, and also for FO) and rejected Boothe's SPOOLER because it must occupy a DOS partition, like POWER.

In summary, DOS ASAP's competitive price and wide range of available and planned features make the system an attractive alternative – not only to IBM's POWER, but also to other commercially available spooling supplements. ASAP users contacted by Datapro report that the system performs very well and lives up to the claims made by USI.□

➤
- Support of UCS for printer output.
- Ability to backspace the printer file for recovery.
- Automatic forms alignment routine; support for printer/punch forms changes; and support for special printer carriage tapes.
- Shared spool file ability, allowing output from more than one spooled device to share the same disk extent for more efficient space utilization as well as reducing arm movement.
- Priority Interrupt feature, allowing priority job output to be printed while holding output from prior jobs.
- Partition balancing.
- Imaginary device support.
- Program relocation.
- Multiple copies.
- Job accounting.
- Simultaneous printer/tape output.
- Tape-only output (available 4Q 73).
- RJE capability (available 3Q 73).
- Retrieval or cataloged procedures from the source program library (available 4Q 73).

Of particular interest is the ASAP capability to use a single disk extent for all spooled devices. This disk management concept allows conservation of disk space by not requiring the user to set up a separate disk spool area for each output device. For instance, when card output is expected to account for only a small percentage of the total output from a processing run, space need not be tied up on the disk just for that small output requirement. A common disk spooling area can be used. Another advantage of the common spool area is that disk arm movement will be reduced–although this advantage can be partially offset because a single spindle is accessed, rather than multiple spindles with parallel or overlapped access.

For input spooling, job streams can be loaded from one card reader and processed in any partition. A priority interrupt queue can be used to alter the normal DOS first-in/first-out priority scheme.

About 30 diagnostics are provided for the user by ASAP.

HARDWARE/SOFTWARE REQUIREMENTS: IBM System/360/370 Model 25 or higher operating under DOS with command chain retry; appropriate input/output devices; and a practical minimum of 10 cylinders of 2314 disk storage or 30 cylinders of 2311 disk storage. This disk requirement is for a shared spooling area typically capable of supporting 2 printers and a card punch. Neither a multiprogramming supervisor nor the Storage Protection feature are required. The basic ASAP system can be tailored to operate in about 4K bytes of memory, but practical main memory requirements consist of:

Devices Supported*	ASAP Size (bytes)
1 printer and 1 card punch	6K
1 printer, 1 card reader, and 1 card punch	8K
2 printers, 2 card readers, and 2 card punches	10K
3 printers, 3 card readers, and 3 card punches	14K

* Applies to real or imaginary devices.

PRICING: ASAP is available for purchase only. Maintenance and future enhancements to ASAP are furnished at no additional cost for 1 year, and at $300 per year thereafter. The prices for the 3-year ASAP license are as follows:

Writer Version	$2,900
Reader Support	600
Imaginary Device Support**	475
Additional CPU***	1,000
Job Accounting	250
RELO-PLUS (with ASAP)	950

** For any number of imaginary devices.

***For the 1st additional CPU; further multiple-copy discounts are available.

European customers should contact Westinghouse Management System, SA (Paris), USI's European marketing representative for DOS ASAP.

INITIAL DELIVERY: January 1971.

CURRENT USERS: About 180 as of June 1973.■

RELO-PLUS
Universal Software, Inc.

MANAGEMENT SUMMARY

RELO-PLUS is a System/360 or 370 DOS enhancement that saves program library space and improves operational efficiency by allowing one copy of a program to be cataloged in the core image library and executed in any partition.

RELO-PLUS is among the best of a considerable number of enhancements for IBM's DOS. Several were available prior to the announcement of DOS/VS by IBM in August 1972. With that announcement, much of the steam was taken out of the marketing efforts for DOS enhancements by independent software vendors.

DOS/VS (Release 28 of IBM's DOS) includes a self-relocation capability that may supersede the need for RELO-PLUS. Furthermore, the availability of DOS/VS at no additional charge on System/370 Models 115 through 158 tends to offer an almost irresistible incentive to current System/370 DOS users (who can make excellent use of RELO-PLUS at this time) to upgrade to DOS/VS. Coupled with the end of free support for DOS on System/360 or 370 computers by IBM on March 31, 1973, a truly remarkable front of pressure is being applied to DOS users to give up DOS and/or the System/360 in favor of OS and/or DOS/VS on the System/370. (An analysis of IBM's upgrade strategies can be found in the System/370 report, 70C-491-04).

On the other hand, preliminary reports on initial use of DOS/VS Release 28 reveal some problems, and IBM has already used Release 29 in some European locations. Thus, System/370 DOS Release 27 users and users of System/360 DOS can well profit from a DOS self-relocatability enhancement for some time to come. Subsequent to the free availability of a proven self-relocatability function in DOS/VS, the market for RELO-PLUS may be pretty much confined to a dwindling list of System/360 DOS users—but their ranks currently number some 10,000. For these existing DOS installations, RELO-PLUS can provide a useful additional facility.

Self-relocatability permits savings in the following areas:

- Machine time—RELO-PLUS eliminates the need to catalog the same program in the Core Image Library (CIL) more than once, for use in more than one partition, and similarly eliminates the need to re-catalog the program when the supervisor or partition size changes.
- Programming time—self-relocatable programming is a complex and time-consuming effort if not done automatically.
- Operations—only one set of Job Control Statements per program is required, and job scheduling is greatly simplified since any job can go into any partition (space and peripherals permitting).

▷

CHARACTERISTICS

SUPPLIER: Universal Software, Inc., Commerce Park, Danbury, Connecticut 06810. Telephone (203) 792-5100.

BASIC FUNCTION: RELO-PLUS is an IBM System/360 or 370 DOS enhancement that allows one copy of any program to be cataloged in a core image library and be executable in any partition, thus saving disk space and easing operational constraints. It also speeds program loading by eliminating DOS directory seeks and fetching multiple library blocks in a single disk revolution.

OPERATION: Relocation is accomplished by adding one control card to a program while cataloging it. Upon execution, relocation is transparent to the user and completely automatic. The card is an ACTION RELO card, added as follows:

```
// JOB ANY
// OPTION CATAL
   ACTION RELO
   (and so on).
```

INSTALLATION: Installation is a 4-step process requiring under an hour in most cases: (1) catalog the RELO-PLUS macro in the source statement library, (2) include a card with "RELO" punched in columns 10-13 directly after the PIOCS statement in the supervisor source deck, (3) compile and catalog the supervisor after checking the assembly listing for size and diagnostic messages, and (4) catalog the RELO-LINKEDT phases in the core image library.

The basic RELO macro adds about 250 bytes to the supervisor, and the improved fetching option adds about 400 bytes. The option can be omitted by specifying "CORFCH = NO" starting in column 16 of the card mentioned in step (2) above.

HARDWARE/SOFTWARE REQUIREMENTS: RELO-PLUS can be installed on any System/360 or 370 system using DOS Release 27 or lower.

PRICING: RELO-PLUS is available on perpetual license at a cost of $2,000. Monthly rental for the package is $150. If the user also has USI's DOS ASAP, a perpetual license for RELO-PLUS is available for $950.

INITIAL INSTALLATION: January 1973.

CURRENT USERS: 20 as of June 1973. ■

RELO-PLUS
Universal Software, Inc.

▷ ● Disk storage—RELO-PLUS saves Core Image Library (CIL) space since the program need be cataloged only once.

As a concept, relocatability (whether by "self" or by an operating system agency) offers a great potential for increased system resource utilization. Under unaided DOS, programs are non-relocatable in the core image library; i.e., they have been assigned to specific main memory locations. Thus, if one of the three partitions in a DOS multiprogramming environment is free, but not the one for which programs awaiting execution have been cataloged, that partition (and share of the system resources) must remain idle. Relocatability allows a program to run from any location in main memory, provided that enough space and peripheral devices are available. Larger systems with plenty of peripherals naturally have more to gain from relocatability, since there is more room to maneuver from a scheduling point of view, and also because the potential for wasted resources is greater.

As the most recent relocatability routine, RELO-PLUS has certain advantages over some or all of the programs of this type that were available before the DOS/VS announcement: RELOCATE (Report 70E-094-01), DOSRELO (Report 70E-100-02), and Anyplace II (Report 70E-603-01). Among these are the following:

- RELO-PLUS supports the relocation of IBM-supplied programs, including compilers, utilities, BOMP (Bill of Material Processor), etc.
- The ANS COBOL SORT verb and segmentation facility present no problem—programs using these ANS COBOL features can be relocated. In fact, any single or multiphased program written in any source language (e.g., Assembler, COBOL D or ANS COBOL, PL/1, RPG, or FORTRAN) is automatically relocated.
- No pre- or post-link-edit runs are required, nor must any special link file be built or maintained, and phases can be link-edited directly from the relocatable library.
- RELO-PLUS does not add to the core requirements of, or modify, any user program.

Relocation is accomplished by the simple addition of one control card when a program is cataloged. Upon execution, relocation is completely automatic and transparent to the user.

RELO-PLUS, like some other relocatability routines, provides an improvement to the DOS fetching method that saves time in loading programs from the core image library by eliminating directory seeks and fetching multiple library blocks in a single disk revolution.

As the other Datapro reports on relocatability programs indicate, prices for the relocatability routines are going down; RELO-PLUS is the newest and one of the lowest in price. Additionally, RELO-PLUS comes from a reputable vendor that has had success in the DOS enhancement market with another product, DOS ASAP (Report 70E-879-01).

USER REACTION

RELO-PLUS users contacted by Datapro indicated that the package performs as advertised, can be installed in less than an hour, and is receiving excellent vendor support. All of the users contacted were also DOS ASAP users.

One user with a 256K System/360 Model 50 runs two batch jobs and a teleprocessing job during the day and three batch jobs at night, and reports having made the IBM 44K Assembler relocatable. This user had only one problem, and that has since been fixed. The bug was an unusual one: one self-relocating program issued a fetch to a nonrelocating program, and, at first, addresses weren't resolved properly.

Another user, with a 128K System/360 Model 30, reports that he's using a now-relocatable COBOL-F compiler, and has also used RELO-PLUS on all of his programs except for his private library system. The user likes the way RELO-PLUS resolves load points and address constants at load time, simply renaming some IBM phases. □

UCC TWO
University Computing Company

MANAGEMENT SUMMARY

There's at least one aspect of IBM computer system operation that's disturbingly reminiscent of the jokes that begin "I've got some good news and some bad news for you." In this case, the good news is that you're upgrading to OS. The bad news is the associated conversion process. UCC TWO (formerly DUO 360/370) is a package designed to help take the sting out of the bad news.

Many installations that change from DOS to OS find that some portion of their operational day must still be devoted to running DOS, so that DOS programs which have not been converted can continue to be used. There are a number of obvious faults in this approach, including the need for dual system training for operators and for maintenance of dual operating systems, language processors, data sets, operating system libraries, JCL decks or cataloged procedures, and all the corresponding documentation.

IBM offers no effective transition aids for upgrades to OS within the System/360 family. For the System/370, IBM does offer an OS/DOS Emulator that runs the actual DOS operating system within an OS partition or region. This approach, obviously, consumes a large amount of core storage: a minimum of 40K bytes to emulate DOS in a partition under OS/MFT (23K bytes of emulator plus 16K bytes for DOS), or 44K bytes for an OS/MVT region (29K bytes of emulator plus 16K bytes for DOS). Furthermore, the IBM emulator requires hardware emulator features, which are standard on the 370/135, 145, and 158 but optional at extra cost on the 155. Use of the emulator on the 135 and 145 can require the acquisition of additional control storage, which the user does pay for.

Other aspects of IBM's System/370 OS/DOS Emulator are even more disturbing: (1) both OS and DOS JCL must be included with each DOS program; (2) DOS programs continue for the most part to use DOS facilities and are limited to three concurrent jobs; (3) DOS programs enjoy only limited use of OS facilities; (4) devices can only be shared if the proper OS JCL is added; (5) DOS ISAM can be used, but parallel files must be kept in OS. Using OS ISAM requires adding OS JCL plus from 6K to 8K bytes of additional memory; (6) no storage protection is provided between DOS programs in a multiprogramming environment under OS/DOS emulation; (7) spooled output from DOS programs under the emulator is not released to the OS writer until execution of the entire DOS job stream is completed; (8) a DOS SYSRES disk pack (at least) is required; and (9) the timer is simulated, and therefore may not be accurate.

UCC TWO removes all of these restrictions and is usable on System/360 as well as System/370 computers. Because OS facilities are available to DOS programs under UCC TWO, and the UCC TWO overhead is only about 6K to 16K bytes, UCC TWO can even yield performance improvements for DOS job streams where OS data management facilities such as spooling, device independence allowing tape files to be simulated on disk, etc., are used. ⊳

UCC TWO facilitates DOS-to-OS conversions by permitting DOS programs to run under OS or OS/VS on a System/360 or 370 and access most OS facilities. It is simpler to use, often less costly, and causes less system degradation than IBM's OS/DOS Emulator, yet provides more comprehensive capabilities.

CHARACTERISTICS

SUPPLIER: University Computing Company, P.O. Box 47911, Dallas, Texas 75247. Telephone (214) 637-5010. (UCC TWO was developed and originally marketed as DUO 360/370 by Computer Technology, Inc., formerly a wholly owned subsidiary and now a part of UCC.)

BASIC FUNCTION: UCC TWO is a DOS to OS conversion aid for IBM System/360 and 370 computers that permits DOS object programs to be run within an OS region or partition. DOS problem programs run under OS, and OS facilities are available to these DOS programs. Using DUO, a user can convert his installation to OS according to a deliberate, measured schedule by running most of his existing DOS programs simultaneously with other already converted or newly developed programs under OS.

OPERATION: UCC TWO operates under OS and consists of an addition to the OS nucleus plus a library of some 40 OS assembly-language programs. The addition to OS operates to trap DOS supervisor calls (SVC interrupts), and interfaces these problem program requests to OS facilitates which provide the same net services. The routine which is added to OS is called the DOS SVC FILTER and is link-edited into OS along with a storage area for LUB's and PUB's (logical and physical unit blocks), which must reside below 32K in main memory. Resident requirements for the OS nucleus are increased by about 1200 bytes.

The basic differences between OS and DOS problem programs are in the conventions they use to interface with their respective operating systems. UCC TWO intercepts the operating system SVC's issued by a DOS program and converts these requests to acceptable OS form (and vice versa for operating system communication with the problem program), thus permitting DOS programs to operate under OS without internal changes.

Because full OS facilities are accessible under UCC TWO, some restrictions imposed by DOS are eliminated and a number of additional features are available:

- DOS programs can use the OS data management facilities such as spooling, device allocation, direct access space allocation, etc.
- Up to 15 DOS programs can be run concurrently, instead of 3 as under DOS.
- All jobs can be read through one job stream and processed through one job queue.
- DOS and OS jobs can be run concurrently, and processing modes can be switched between OS and DOS within a single job step.
- DOS or OS sorts can be used by the DOS program.
- Since DOS programs under UCC TWO are handled like OS programs, all DOS programs are relocatable and can be run in any partition or region, with DOS overlay facilities provided by the UCC TWO Linkage Editor. ➤

UCC TWO
University Computing Company

▷ UCC TWO allows the user who is upgrading from a 360 under DOS to a 370 under OS to step up gradually, making the DOS-to-OS transition before actually replacing his System/360 processor with a System/370. Of course, some limitations and restrictions are present in UCC TWO as well as in IBM's DOS Emulator. IBM 1400 emulation and telecommunications are not supported. But, importantly, UCC TWO supports OCR and MICR input.

Because of the addition to the OS nucleus required by UCC TWO, Datapro was initially concerned that some problems might arise with IBM support. But among all the UCC TWO accounts contacted, not a single instance of trouble on this score was reported. The process itself is simple, and a copy of the original OS nucleus is maintained in case system problems are encountered. UCC correctly makes the point that the OS change is via a link-edit, and therefore should be considered by nervous users to be an attachment to OS, rather than a modification. All IBM maintenance is done on the unaltered operating system, and complete support is provided by UCC for subsequent OS additions.

USER REACTION

The last time Datapro looked at UCC TWO, then called DUO 360/370, we contacted four users, each of whom had at least one year's experience with the package. The only trouble found was almost comic; the system's accounting tables wouldn't permit it to function on Leap Year Day, 1972. UCC has guaranteed that the same thing won't happen in 1976.

Since then, UCC TWO has been elected by its users to the 1973 Datapro Software Honor Roll. A glance at the scores reported (under the old DUO 360/370 name) in our User Ratings of Proprietary Software report, 70E-010-40, shows why: UCC TWO's scores were perfect except for a single user's marks. Moreover, Datapro's investigations showed that the lone set of low grades was based upon a temporary misunderstanding that has since been resolved to the user's complete satisfaction.

Contacting the other reporting users, Datapro found only one comment out of the realm of the expected. One user feels the package's worth is of particular significance when the system being converted from DOS to OS is using major amounts of assembly-language coding; otherwise, it's his opinion that COBOL and other high-level language installations should be prepared to bite the bullet and convert in a straightforward manner. But about 95% of UCC TWO users are primarily COBOL, PL/1, or RPG shops, indicating general disagreement with that premise.

UCC TWO can alleviate a task that is otherwise likely to be a real nightmare for a medium-to-large-size installation with a considerable library of programs. Serious consideration should be given to UCC TWO as a conversion tool not only to aid in the upward migration from a System/360 to a System/370, but also as a possibility to defer the hardware conversion by moving to a more powerful operating system on the same equipment. □

➤
- Console services are provided by a special console handler which withholds actual console message traffic from a terminal until a reply is needed, at which time a batched series of messages is transmitted. Predefined DOS console reply messages can be supplied to reduce or eliminate the level of console activity, which is normally higher for DOS operation than for OS operation.
- DOS jobs can be submitted via remote job entry (RJE) or the Time-Sharing Operating System (TSO).
- MICR and OCR input are supported under UCC TWO, but not under the IBM emulator.
- Local (but not remote) BTAM 2260 support is available.
- OS ISAM instead of DOS ISAM data management facilities are used, without requiring any program changes.

Limitations imposed upon UCC TWO use include the following:

- Location-dependent code, other than LUB's, PUB's, and the full DOS COMRG, is not supported. (There is no DOS supervisor.)
- Restrictive considerations are also present for various aspects of multi-tasking, checkpoint/restart, etc.

Two principal areas of conversion must be addressed when using UCC TWO: data set conversion and Job Control Language (JCL) conversion. OS JCL must be used, which means that existing DOS JCL must be replaced. The resulting DOS programs with OS JCL are then link-edited into OS libraries.

Except for DOS ISAM and split cylinder data sets, DOS disk data sets do not require conversion. ISAM files must be dumped by DOS and recreated under UCC TWO using the DOS Restore Program. If the user runs Release 25 or later of DOS, the same dumped file may be used by UCC TWO, OS, and DOS. Although no file conversion problems exist for unit-record and tape data sets, special consideration must be made for tape label formats when more than one standard-label tape is required, each of which contains the same volume serial number, or if the data set names begin with a blank character.

HARDWARE/SOFTWARE REQUIREMENTS: UCC TWO runs under OS and OS/VS (all versions) on an IBM System/360 or 370 and consists of up to 40 routines which total up to 25K bytes in length. Not all of these routines will be active simultaneously, and many of them are re-entrant. Resident memory requirements for typical installations range from 6K to 16K bytes per UCC TWO partition or region. The OS nucleus is increased in size by about 1200 bytes by the addition of a segment of UCC TWO coding.

PRICING: UCC TWO is available only on lease, with maintenance included. UCC recognizes two types of customer: those who use UCC TWO only for in-house work, and those who use it for external third-party applications. The latter are considered "service" users, and pay lease rates 25% higher. Multiple CPU leases are available at discounted rates.

Minimum Term (months)	Monthly Charge (internal use)	Monthly Charge (external use)
6	$1,300	$1,625
12	$1,000	$1,250
24	$ 800	$1,000

INITIAL DELIVERY: August 1970.

CURRENT USERS: About 200 as of mid-August 1973. ■

UCC ONE
University Computing Company

MANAGEMENT SUMMARY

UCC ONE, an IBM System/360 and 370 OS tape management system, protects data, reduces operator activity, provides real-time information, extends OS, VS1, or VS2 capabilities, and can increase throughput and reduce costs. Its tape record-keeping facility can help to eliminate errors by librarians.

Let's suppose that you're the manager of a data processing system employing a magnetic tape library of over 20,000 reels. How do you feel when you hear that an operator has clobbered an important master file by mounting it as a scratch tape? If your answer is unprintable, you may need a tape library system.

University Computing Company's UCC ONE has been in commercial use since September 1971. It was originally called the UCC Tape Management Software, or simply TMS. Its function is basically to provide tape library services that reduce or eliminate human errors. It will normally prove to be cost-effective in libraries with about 1500 or more volumes. Additional benefits of the package include the potential to improve the operational efficiency of a system by reducing required manual library functions, providing immediate and audit information, extending some OS facilities, and protecting data.

UCC ONE eliminates use of external tape reel labels and generates automatic picking lists for scratch tapes. It streamlines operations during job preparation by doing away with labor steps in creating data set labels and placing them on the reels. While a job is running, operator decisions as how to respond to OS messages giving him responsibility for overriding non-expired tapes are eliminated, as is the need to handle file protect rings or verify the correctness of mounting no-label tapes.

Once a data set has been created on tape, the UCC ONE user's operator has no need to worry whether he's placed the proper label on the new reel, because labels are not used. When the tape is returned to the library, UCC ONE automatically takes care of recording and retaining data set names, volume serial numbers, retention periods, and so on, without any human need to copy or keypunch these items or provide an inventory or scratch list.

To do all this, UCC ONE receives control at Open, End-Of-Volume (EOV), and Close, and records all necessary information about the tape volume in a disk data set called the Tape Management Catalog (TMC). For tapes used as input, UCC ONE checks the JCL statement against the TMC entry for that volume to make sure that the data set names match. Full OS checking is provided for tapes with standard labels, non-standard labels, and no labels. Then, facts such as the job name, date, and address of the tape drive used are stored, along with an updated use count for the reel, in the TMC record.

If the tape is used for output, the program checks the TMC record and determines whether the tape is eligible for output use. Then, if the expiration date is the current date or less and the tape is not still otherwise protected, UCC ONE records the data about the data set being created (or extended to additional volumes) on the TMC.

▷

CHARACTERISTICS

SUPPLIER: University Computing Company, 700 Stemmons Freeway, P. O. Box 47911, Dallas, Texas 75247. Telephone: (214) 637-5010.

UCC also has offices in New York City, Atlanta GA, Troy MI, Vienna VA, Chicago, and El Segundo CA. In the United Kingdom, contact UCC Computer Utility Centre, 143 Bromsgove St., Birmingham B5 6RH; telephone: (021) 692-1041. In Australia, contact UCCOMPUNET, 23 Cleg St., Sydney.

BASIC FUNCTION: UCC ONE (formerly called Tape Management Software, or TMS) is a tape library system for IBM System/360 and 370 OS systems. Its main purpose is the elimination of manual Tape Librarian functions (e.g., creation and placement of physical labels on reels, ledger- and record-keeping, etc.), but audit, backup, and reporting and inquiry facilities are also provided, as is the capability to use tapes from outside a particular installation.

OPERATION: Several major areas of responsibility rest upon the systems programming staff of a UCC ONE user. Four of these functions are performed only once for each release of the operating system. These are establishing the UCC ONE console, installing the UCC ONE program modules, establishing the UCC ONE data sets, and establishing modifications to OS Data Management. UCC ONE can be initialized after IPL or will initialize itself at the first OPEN. The Tape Management Catalog is updated automatically, or by on-line or batch utilities if desired. The TMC backup services must be performed on a periodic basis. These functions can be performed under the direction of the systems programming group or the tape librarian.

The UCC ONE console can be used for on-line inquiry and/or update, and can be an OS or ASP console. Program modules to be installed include volume serial number editing programs, SVC library modules, optional OS linkage library modules, UCC ONE private library modules, OS PARMLIB modifications, and, under ASP, DSP (dynamic support program) modules. The package gives the user the practical capability to run complicated tape jobs from an RJE terminal for the first time.

Two programs, USEREDIT and EDITBACK, respectively, convert volume serial numbers that are not 6-digit numerics into that format for UCC ONE, and do the reverse. These programs must be user-written. Volume serial numbers must be unique and must occur in contiguous ranges. For example, an installation could have 30,000 volumes with serial numbers in the ranges 000001-010000, 500001-510000, and 720001-730000, but the same 30,000 volumes scattered between 000000 and 999999 could not be handled with a TMC of reasonable size.

There are at least 17 UCC ONE private library modules. Each has "TMS" in the first three letters of its name, a carryover from the days when the system was known as UCC Tape Management Software. Modules can, for example, produce reports on an expired catalog list, scratch and clean list, produce the volume serial list, produce the high-retention list, test OPEN and EOV interfaces, and update any field in a TMC record.

➤

UCC ONE
University Computing Company

For complete TMC backup, an Audit Data Set record is written each time a TMC record is updated. If the tape is from an outside installation, it can still be handled by UCC ONE. But instead of TMC and Audit Data Set records, an Exception Data Set record is maintained for these tapes.

Finally, UCC ONE has a spectrum of accounting, reporting, and inquiry aids. A special program allows on-line inquiry and authorized updates to the TMC; this facility exists for OS and its VS counterparts, HASP, or ASP, and can also be run under batch utilities. UCC ONE has user hooks that allow use of a 41-byte reserved field in the TMC for any purpose. A systems utility is provided to compare entries in the OS System Catalog against those of the TMC and report on any differences. Another available utility provides TMC records in a sequential manner and suitable for input to a batch utility or report generator program. Additional reports are available through utilities provided as part of the UCC ONE system.

USER REACTION

All users that Datapro contacted, whether we learned of them through the vendor's auspices or from the recent Datapro survey of proprietary software users, indicated that the UCC ONE system performs as advertised in a fast and efficient manner. Sample quotes are: "It (UCC ONE) provides the best and most accurate tape record keeping (this installation) has ever had", "UCC ONE has eliminated human errors (in the tape library)", and "Our librarians say they couldn't live without it."

Nevertheless, one survey response indicated only fair quality of documentation and vendor service, and Datapro dutifully checked it out. In so doing, we brought to light facts that should have a constructive effect upon the vendor's service in this particular aspect in the future and should thus benefit the user community. The complaint was about long lead times between OS releases from IBM and release from UCC of necessary user hooks for the Open, Close, and End-Of-Volume modules in the operating system. The user reported that it took two months after his installation of OS Release 21.0 and again after his installation of OS Release 21.6 for UCC to supply the hooks. Also, the user stated, a large PTF (modification) on OS Release 21.6, which is considered a prerequisite by IBM and which impacts End-Of-Volume modules, destroyed the UCC ONE hooks for this user's unusually large number of multivolume unlabeled tapes. This problem existed for four months before the UCC-issued correction arrived.

Commenting on this particular problem, UCC reports that Release 21 of OS involved a major rewrite of OS OPEN/CLOSE/EOV, and that locating the UCC ONE "hooks" required microfiche and/or module dumps and documentation from IBM, the late delivery of which accounted for the delay. UCC claims that lead times for the "hooks" has been reduced in later releases, but concedes that customers sometimes find the "hooks" for themselves first if they have earlier access to the required information from IBM.

Two recently added modules are Vault Management and Auxiliary Disposition. The former controls tapes routed in and out of the library via batch utility programs, and can be applied to controlling tapes that were sent to a vault or series of vaults and then must be retrieved or exchanged, or to mailing tapes out of a library. Routing can be by Julian date, cycles, or other factors. The latter module enables users to print any user-defined text on the console when a tape is unloaded; it can direct an operator to hold a tape for a succeeding job or send the tape to an off-line printer, for example.

Prior to beginning UCC ONE operation, the TMC, an Audit Data Set, an Exception Data Set, and a Backup Data Set must be established. The first two of these should be isolated from each other for additional security from catastrophe.

A difference in tape operations messages (there are several such differences between UCC ONE and OS) is of prime importance: Under OS, the operator is asked to decide if an unexpired data set can be written over. Under UCC ONE the data management interface module makes the decisions as to the expiration of volumes, and the operator has no choice. Special messages also facilitate handling unlabeled tape data sets and tapes from outside libraries.

The tape librarian has four typical areas of responsibility when UCC ONE is used; obtaining reports, inquiries into the TMC, updating TMC entries, and volume management. To aid him, UCC ONE provides 11 reports for his use, including a volume serial list, job name list, scratch forecast list (list in volume serial number sequence of all tapes due to be scratched within 7 days of the current date), etc.

PERFORMANCE: Benefits of UCC ONE use will naturally vary, and should be evaluated by individual users. Users can expect improvements in the following areas: (1) increased system throughput due to operational efficiencies inherent in the reduced paperwork duties for tape operators; (2) a possible reduction in the number of person employed; (3) data protection; (4) reduced programmer paperwork, possibly yielding increased programmer productivity; (5) increased capabilities (e.g., ability to run business jobs from RJE terminals); (6) increased stability of operation, potentially improving employee productivity and morale. While many of these performance factors are intangibles, the package bears credible user references and definitely performs as advertised.

HARDWARE/SOFTWARE REQUIREMENTS: UCC ONE requires an OS System/360 or an OS, OS/VS1, or OS/VS2 System/370 with sufficient disk space for the TMC and tape units for other required data sets.

PRICING: UCC ONE can be licensed for 12 months at $475 per month, for 24 months at $400 per month, for 36 months at $350 per month, or for unlimited use at a one-time charge of $10,000. License charges are discontinued after 3 years. Maintenance after the initial year of unlimited use or after 36 months of monthly use is available for $600 per year. The program material is distributed on one 9-track tape reel (800 or 1600 bpi), plus a small card deck.

INITIAL DELIVERY: UCC ONE was first delivered as UCC Tape Management Software (TMS) in October 1971.

CURRENT USERS: 51 as of mid-July 1973. ■

Even the user with this complaint, though, found the package completely satisfying in all other respects. Based on the responses from experiened OS users with literally hundreds of thousands of tape reels in their combined libraries, Datapro feels that UCC ONE has much to offer users with library problems. □

UCC TEN
University Computing Company

MANAGEMENT SUMMARY

UCC TEN is designed to provide essential data definition and control capabilities for IMS users who function in either a batch or on-line environment. As a data dictionary, it manages and controls data definitions, maintaining all information in a central file that can easily be accessed via cross-reference facilities. As a data manager, UCC TEN simplifies daily usage of IMS by providing capabilities for automatically generating control statements, enforcing standards, creating test and production definitions, and assisting with data base design.

The potential advantages of data bases are well known. But consider for a moment the typical managment and control problems associated with data base processing. A data base definition can fail to consider the impact on all programs. With as few as 10 data base programs, often over 3000 different definitional items could be involved. How, then, can each programmer determine what the others are doing to shared data, and what the resulting impact upon his programs will be? How does a manager eliminate costly data name duplications, data redundancies, and needless programming?

With UCC TEN, all control statements for IMS production must come only from the UCC TEN central information file. This file is updated by only a single entry, and any change in one definition is automatically reflected in all control statements generated thereafter, thus assuring management that there will be reliability and consistency among users of the same definitions.

UCC TEN is a central repository for definitions of data bases, data set groups, programs, and communications classifications, both logical and physical. Each element can have up to 99 attributes defined relating to it, and text can be associated with each element. Elements that have been entered into UCC TEN can still be recalled and modified as necessary. UCC TEN can enforce standard naming conventions through a user-written edit routine. Used in on an on-line mode, UCC TEN can capture all data-describing fields, segments, and so on, and collect these elements for use in structuring a desired data base description.

UCC TEN can be used in an on-line mode or a batch mode, or transactions can be entered via an on-line terminal for later batch processing. It can generate the on-line nucleus of a data communications system and provide terminal security input for the security maintenance portion of IMS.

UCC TEN was originally developed for IMS in April 1970, and has been in use at UCC's 23,000-transaction-per-day ▷

UCC TEN, a data dictionary/manager, is a tool that makes IBM's IMS or any OS data base application simpler to use, easier to control, and less subject to errors. It runs on IBM System/360 and 370 computers under OS or OS/VS.

CHARACTERISTICS

SUPPLIER: University Computing Company, 7200 Stemmons Freeway, P.O. Box 47911, Dallas, Texas 75247. Telephone (214) 637-5010.

UCC regional offices are: Eastern–New York, NY and Atlanta, GA; Midwestern–Troy, MI and Chicago, IL; Southwestern–Dallas and Houston, TX; and Western–El Segundo and Palo Alto, CA. International operations are handled through the Dallas office. Offices are also located in most major U.S. cities.

BASIC FUNCTION: To make IMS simpler to use and easier to control by (1) managing and controlling data definitions from a central file, and (2) generating control statements, enforcing standards, creating test and production definitions, and assisting in data base design. UCC TEN can also be used with any OS data base application, as it supports all OS file organizations. UCC TEN can be used as an IMS application on-line, or as a DL/1 batch program. No modifications to either IMS or OS are involved. It is predicted that UCC TEN will also run with any IMS virtual storage system.

OPERATION: All control statements neccessary for IMS production must come from the UCC TEN central information file, which is updated by a single entry that is thenceforth reflected in all automatically generated control statements. The program is transaction-driven, and uses function codes to identify each operation to be performed.

UCC TEN helps data base administrators by clarifying data structures and avoiding data redundancies through the central information file of data bases, programs, communications, and hierarchical structures. It generates accurate, dependable data definitions through automatic control-statement generation. It accesses impacts of changes by cross-referencing elements, providing program profiles, and supplying transaction terminal security. Finally, for the data base administrator, it avoids duplications and, through a user-written module, enforces standards such as naming conventions and editing for IMS requirements.

UCC TEN helps operations personnel to avoid production failures because the package largely eliminates errors due to missing data bases, programs, transactions, etc., from IMS generations and allows generation of test data definitions independently of production. It also helps to solve operational problems by answering inquiries about data base and program relationships.

Using UCC TEN, analysts and programmers can avoid needless programming, since it provides a centralized source of all program and data definitions. They can also reduce ➤

UCC TEN
University Computing Company

IMS environment in Dallas. It was enhanced in April 1972 to incorporate IMS on-line facilities.

In use, UCC TEN tracks 12 definitional elements, most of which can cross-reference one another in various directions. The elements are: data base, data set group, segment, field, program, transaction, line group, line, physical terminal, logical terminal, control unit, and pool/subpool. The types of questions UCC TEN can provide answers to, for example, are: What programs access a segment? What data bases can be accessed by this program? What hierarchial structure is used by this data base? What transactions are permitted from this terminal? What attributes does this segment have? What logical terminals are associated with this physical terminal? What transactions are processed by this program? UCC TEN can answer these and many more questions, including some of a more technical nature.

USER REACTION

UCC TEN users contacted by Datapro were unanimous in their opinions that the package performs as advertised. What's more, there was nary a whimper about tradeoffs, meaning that the package's services are regarded as so beneficial that its core requirements and run times are completely acceptable. And Datapro talked to some very sophisticated users; one, for example, has two 370/155's and a 360/65MP joined in a multiprocessing system, plus an additional 370/145. Another has a 370/165 and a 370/155, both running IMS-2.

UCC TEN has been called "the heart of the data base administration area" by one user. It serves its users by providing a dictionary to control use of the IMS environment, by generating IMS macros, by furnishing a central file of IMS information, and by ensuring consistency and security in IMS use.

UCC's technical support was always rated as satisfactory, and usually excellent. Some users reserved comment on vendor support because they'd never had trouble with UCC TEN. Another user pointed out that UCC has been most cooperative in adopting suggestions for improvements. Complaints were picayune; one user would like to see a reference card for the 90 or so key words in UCC TEN, and another wants the package overview in the user's manual improved.

In summary, UCC TEN should be considered as an advisable acquisition to assist in the all-important data base administration function in an IMS system. □

development time because of the single-entry updating and automatic generation of the Data Base Definition (DBD), Program Specification Block (PSB), Program Control Block (PCB), Segment Search Arguments (SSA's), and I/O statements. The package further provides these personnel with test and production definitions, on-line data base design facilities, and automatic documentation.

Systems software personnel are assisted in maintaining production definition integrity because of the ability of UCC TEN to provide independent test definitions. Also, generations are reliable under UCC TEN, which automatically generates Stage-1 Sysgen statements, selects from either production or development, and compares the current to the previous one.

And finally, UCC TEN does a few things for management. Its design and coding aids speed new application development. Its elimination of error-prone procedures and automation of manual tasks can greatly reduce errors. By providing key data to the proper individuals, it can greatly improve control.

PERFORMANCE: UCC TEN is one of those products that performs a vital service at an acceptable tradeoff in overhead. User consensus indicates that the package performs as advertised. Please refer to the "User Reaction" paragraph of the Management Summary for particulars.

HARDWARE/SOFTWARE REQUIREMENTS: UCC TEN requires an IBM System/360 or 370 under OS or OS/VS. It consists of 125 modules, of which about 90% are coded in ANS COBOL and the rest are coded in Assembly language. Batch-mode operation requires at most 100K bytes of memory in addition to DL/1 requirements. On-line operation requires a 56K-byte IMS message processing region. Defined within UCC TEN are seven HISAM data bases. They are Data Base/Data Set, Segment, Field, Program, Communications, Text, and Queue. (These support the 12 UCC TEN definitional elements that operation relies upon.)

PRICING: UCC TEN is distributed as unloaded load-module data sets on magnetic tape, with three copies of a user's guide, installation material, and two man-days of on-site training and installation assistance when requested. UCC grants a 30-day acceptance period. Maintenance is included in monthly rentals for up to 36 months and for the initial year of an unlimited license; thereafter it costs $1,200 per year. Additional copies of UCC TEN are discounted at 25% of the charge for each successive copy.

Minimum License Period	Charge
36 months	$550/month
24 months	$625/month
12 months	$700/month
Unlimited	$15,000

INITIAL DELIVERY: March 1973 (operational within UCC since 1970).

CURRENT USERS: 15 as of September 30, 1973. ■

UCC FIFTEEN
University Computing Company

MANAGEMENT SUMMARY

Most OS computer shops have problems setting a job up for restart or rerun after it has failed. IBM's OS has four types of restarts: deferred, checkpoint, automatic, and step. OS handles most restarts adequately. In a deferred step restart, however, all OS does is to skip to the proper step; but a great deal more needs to be done.

Since it's impossible to write JCL so that the OS catalog will always be correct for a restart, the OS user is faced with a dilemma: If all JCL is processed through the job stream, card handling and storage space are often excessive. On the other hand, if the JCL is placed on SYS1.PROCLIB, it is difficult to begin execution at any step other than the first.

Here are the manual tasks that must usually be undertaken to restart a job: (1) start execution at the proper job step, (2) correct the OS catalog for a subsequent execution, (3) scratch unwanted data sets that will be re-created, and (4) adjust generation data group bias numbers. Along with studying the system message presented when a job fails, changing the JCL and/or catalog, and so on, this all gets a bit sticky and can cause missed production schedules. It can also cause those 3 a.m. phone calls to programmers' homes and 4-hour scrambles to recover from failures. Then there's those reruns of reruns . . .

To understand the value of the UCC FIFTEEN restart management system, it is instructive to consider briefly the rerun problem. Take the following simple example:

You have a two-step production job. In step 1, data set A on disk is an activity set that is input and processed against input master file tape B, which is cataloged as a generation data group. Step 1 input is generation n, and its output is generation n+1. Also output by this step is a work data set, data set C, on a tape. Step 2 accepts generation n+1 of data set B (let's just call it B+1) and data set C as input, and produces tape data set D (the new master) and disk data set E as output.

Now, what is needed to rerun the job, under just one type of failure condition? The answer: Data sets B+1, C, D, and E must be uncataloged. The generation data group bias number for data set B must be readjusted back to n, from n+1. Disk data set E must be scratched. You have to skip to the proper job step. And JCL must be fixed.

UCC FIFTEEN handles all of this for you. Additionally, if you're using UCC ONE (a tape library management system, see Report 70E-885-02), and a tape is uncataloged during restart, UCC FIFTEEN will reduce its expiration date. All of this is done without changes to production ▷

This restart management system can eliminate major headaches that often arise when there is need to restart or rerun production jobs on an IBM System/360 or 370 computer under OS or OS/VS.

CHARACTERISTICS

SUPPLIER: University Computing Company, 7200 Stemmons Freeway, P.O. Box 47911, Dallas, Texas 75247. Telephone (214) 637-5010.

UCC regional offices are: Eastern–New York, NY and Atlanta, GA; Midwestern–Troy, MI and Chicago, IL; Southwestern–Dallas and Houston, TX; and Western–El Segundo and Palo Alto, CA. International operations are handled through the Dallas office. Offices are also located in most major U.S. cities.

BASIC FUNCTION: Restart management in IBM System/360 and 370 OS and OS/VS systems. UCC FIFTEEN prepares jobs for restart by: (1) starting execution at the proper job step, (2) correcting the OS catalog for restart, (3) scratching unwanted direct-access data sets for re-creation, (4) correcting the generation data group bias for restart, and (5) if UCC ONE is in use and a tape is uncataloged during restart, reducing the tape's expiration date.

OPERATION: To perform its basic functions, UCC FIFTEEN uses a disk data set called the Catalog Management Table (CMT). The CMT contains information for each job step regarding data sets which are to be cataloged. Based on the information that was automatically recorded in the CMT during the prior normal production run, tapes are scratched, bias numbers are reset, etc., automatically before the rerunning of a production job.

UCC FIFTEEN is run as the initial step of a job, with a parameter specifying whether the job is a production run or a rerun, and, if a rerun, what range of steps is to be executed.

In use, the UCC FIFTEEN step is added as the first step of the production job; this can be in PROC or as card JCL. For a rerun, only one JCL card need be changed. That card is the parameter card, which specifies: (1) first time–build a CMT entry; (2) this is a production run; (3) restart, with start-step and/or end-step optionally specified; (4) special case, deferred scratching of disk data sets; or (5) bypass the UCC FIFTEEN function.

The only impact upon JCL standards is that jobs must have unique job names that do not change from run to run, and step names must be unique within jobs. For best operation, it is recommended, but not required, that refer-backs and condition parameters referring to previous steps be minimized.

Installation of UCC FIFTEEN requires a link-edit of the package and allocation of a partitioned data set for the ➤

UCC FIFTEEN
University Computing Company

▷ JCL other than one change to the parameter card of the UCC FIFTEEN restart program.

USER REACTION

UCC FIFTEEN is credited by large, sophisticated users in banking, government, and industry as having given personnel on second and third computer-room shifts (these are personnel not normally as "sharp" on JCL or as likely to have been associated with an application's structure as prime-shift personnel) the ability to restart complicated jobs without the usual attendant confusion. In one user's opinion, UCC FIFTEEN fills in a gap in catalog management left in the OS software by IBM.

Users say that UCC FIFTEEN installs easily, and one points out that UCC is the only vendor, in his experience, whose PTF's (program temporary fixes) work the first time. The users rate UCC as very responsive as well. UCC states that the program will be supported through all OS releases, and user experience has shown it to work properly the first time upon transfer from OS/MVT to OS/VS2.

In summary, UCC FIFTEEN is a program that can remove the trauma and panic from restarts. It should be closely examined by users with OS applications having complicated job steps, especially when their operations are around-the-clock. □

CMT, which will be built automatically the first time the job runs under UCC FIFTEEN. The size of the CMT is determined by the number of entries, each entry being an 80-byte record describing the data set to be handled by the restart management program.

PERFORMANCE: UCC FIFTEEN runs in about 20K bytes in a few seconds, with the exact time depending upon the application, configuration, and location of the data sets involved. Please refer to the testimonials in the "User Reaction" section of the Management Summary.

PRICING: UCC FIFTEEN is delivered on a tape that contains the program and source code. Also included is a sample set of the JCL necessary to install it into the operating system, implementation instructions, and three copies of the user's guide. UCC grants a 30-day acceptance period. Maintenance, including fixes and all updates for new releases of OS and program enhancements, is included in the monthly rental for up to 36 months and for the initial year of an unlimited license. Thereafter, the maintenance charge is $420 per year. Multiple copies are discounted at 25% of the price for each successive copy.

Minimum License Period	Charge
36 months	$185/month
24 months	$200/month
12 months	$240/month
Unlimited	$5,000

INITIAL DELIVERY: December 1972.

CURRENT USERS: 14 as of September 30, 1973. ■

Assembler G
University of Waterloo

MANAGEMENT SUMMARY

Assembler G is a program that will be of significant interest to confident, knowledgeable users of large System/360 and 370 computers because of its potential in terms of cost savings and extended features. Users should note, however, that the program originates from and is supplied by an academic center, and not by a commercial software vendor.

The University of Waterloo, in Ontario, developed Assembler G to meet its performance needs (that's a nice way of saying "to overcome IBM's Assembler F inadequacies"), and first installed the system in March 1968. There are currently 108 educational and 177 commercial users of Assembler G. The program can be obtained for just $300, a charge that merely defrays the costs of documentation, maintenance, and distribution.

The University of Waterloo will undertake to provide certain services, but users must be aware that they cannot count on all the services normally expected from a commercial vendor. The University will: (1) mail a copy of the source and object programs and documentation, with sample jobs for installation and testing, on a tape supplied by the user; (2) place the user on a mailing list to receive modifications automatically until January 1975; and (3) provide advice by mail or telephone. Conversely, the University will accept no obligation to adapt the assembler to the needs of any individual installation.

Semiannual maintenance can be expected to be the norm for as long as the University of Waterloo itself uses the program, and it plans to do so for the foreseeable future. In fact, another version of Assembler G can be expected, because it is currently based on the about-to-be discontinued IBM Assembler F 21.7, and the next version will run under the Cambridge Monitor System (CMS) under the IBM VM/370 facility. The current Assembler G supports IBM System/370 instructions.

So much for the disclaimers and extenuations. Now let's look at what Assembler G (ASMG) can do for you. It can:

- Dramtically outperform Assembler F, in direct proportion to the number of macros assembled. It does so in an optimum region of about 150K bytes (versus about 80K for ASMF), but is nearly as fast as Assembler H, which optimally uses about 250K.
- Maintain its high performance when the link pack area is in IBM 2361 or equivalent Large Core Storage (LCS), popular on the 360/65 and up, whereas ASMF suffers badly.
- Provide some of the language features of IBM's ASMH (though the 2-pass design of ASMH is different) and Extended ASMF (ASMXF) in a version of ASMG that will be available early in 1974. (Certain experts feel that ASMXF does no more than provide ASMH-type features at about an ASMF performance level.)

Assembler G is an inexpensive, extended-feature assembler for IBM System/360 and 370 computers under OS, OS/VS, and CMS. Especially on a 360/65 or larger, it assembles macro-laden programs much faster than IBM's Assembler F, and nearly as fast as IBM's Assembler H, but in much less memory.

CHARACTERISTICS

SUPPLIER: Supervisor of Technical Products and Program Distribution, Computing Centre, Math & Computer Building, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada. Telephone (519) 885-1211, extension 3268.

BASIC FUNCTION: To improve the assembly (especially macro assembly) of programs on large-scale IBM System/360 and 370 computers under OS/PCP, OS/MFT, OS/MVT, OS/VS1, OS/VS2, or the Cambridge Monitor System (CMS). Assembler G can be used on a system as small as a 360/40 with floating-point hardware, but optimal performance is obtained on a 360/65 or larger due to double-word instruction and data fetches. The assembler can operate in virtual mode but is not currently optimized for such operation. Assembler G also offers language extensions and features not found in IBM's Assembler F. (Strict conformity with ASMF can be requested via the EXTEN parameter) It also requires much less main storage than IBM's Assembler H. It is faster than, almost as powerful as, and about the same size as IBM's Assembler F Extended.

OPERATION: Compared to IBM's Assembler F, Assembler G has 3 minor incompatibilities and 13 major extensions. These are listed below.

Incompatibilities: (1) the default options are LOAD, NODECK instead of DECK,NOLOAD: (2) the default instruction set includes the branch conditional register mnemonics, so error messages will result for programs that contain definitions for these mnemonics as macros; and (3) SYSLIN is the name of the DD card preferred for writing the object deck under the LOAD option; if SYSLIN is missing and SYSGO is present, the latter will be used.

Extensions: (1) abbreviations are accepted for PARM options, and many new PARM's exist to control other extensions; (2) printing of the External Statement Dictionary (ESD) and Relocatable Dictionary (RLD) is normally suppressed, but optionally allowed; (3) the XREF is normally printed in a paper-saving, time-saving "squished" format; it can be optionally printed in full format, in 2-column format, squished or full, or suppressed; (4) a literal cross-reference (LREF) printout is optional; it provides the same information about literals as XREF does about symbols, and it is listed in EBCDIC collating sequence with format allowance for extreme and variable literal string lengths; (5) a BATCH option allows multiple source decks to be assembled in one step, eliminating multiple job steps; (6) an EXECUTE option permits "load and go" operation immediately after assembly; (7) an INSTSET= option allows assembling programs for the 360/20, 360/44, or 360/67, DOS Assembler F and OS Assembler F compatibility, extended branch registers, and System/370; (8) since the size of the unsubsetted local dictionary can exceed 64K, programs larger than those allowed under Assembler F can be assembled; (9) data sets with unlike characteristics or on unlike devices can be concatenated on SYSIN or the optional SYSUP; (10) an EXTEN option controls eight language extensions: PRINT statements allowed in macros, attributes of symbols defined in macros, &SYSDATE, &SYSTIME, &SYSTYP, and &SYSPARM system variable symbols, support of named

Assember G
University of Waterloo

- Handle larger programs than ASMF can.
- Assemble OS or DOS programs for System/360 Models 20, 44, and 67, and for System/370 computers.

There can be no doubt that IBM software is designed, at least in part, to sell IBM hardware. One way to exclude IBM's "silent salesmen" from your computer site is thus to develop or obtain systems software elsewhere. In-house development can be costly and often amounts to "re-inventing the wheel." Competent, self-reliant installations should heed the advice derived from the following summary of user experience and seriously consider software facilities such as the University of Waterloo Assembler G.

USER REACTION

Assembler G users with considerable expertise and experience (using systems involving both a 370/165 and a 370/168, and even a 360/75 and 360/91 ASP system, for example) don't excite easily, but they nonetheless often find Assembler G's speed amazing. In fact, some users have standardized on Assembler G, saying there's just no comparison with the IBM Assembler F. One user mentioned that reductions in assembly time from a half hour to 6 minutes have been typical in using ASMG instead of ASMF.

The users interviewed by Datapro find that the main storage required for ASMG (130K for up to 6,000-line programs and 300K for programs twice that size, or sometimes just a standard 240K) is more than justified by the speed and availability of some of their favorite ASMG features: batched assemblies (one user reported 150 assemblies in 1 pass in 6 wall-clock minutes), the update facility (even used to update ASP modules), and language extensions (such as character string variables, an IBM ASMH feature).

Even though Assembler G's vendor is not of the commercial type, users find no problems with "vendor service." The program is said to have a better track record for freedom from bugs than IBM's ASMF. In fact, the only user interviewed who had ever found a bug in ASMF found only one in two years, and that one was non-repeatable and corrected in the subsequent level. □

COMMON, SETC variables with lengths less or greater than 8 bytes, the K' (count) operator used on the SETC variable, SETC variables containing C, X, or B-type self-defining terms used in SETA expressions, and macro definitions included in source code as programmer macros using COPY, providing COPY doesn't exist in these macros; (11) a FULLLIST option prints library macros as they're edited so that syntax errors in them can be pinpointed; (12) by reassembling three assembler modules, some 360/67 RPQ instructions can be supported; and (13) an UPDATE facility is available to allow simultaneously reading an IEBUPDTE format update deck on SYSUP and an old master data set on SYSIN, doing the assembly on the resulting (nonexisting) new master.

Assembler G's excellence is in its real-time performance, which is derived from its speed in macro assembly. This, in turn, is achieved through use of an in-core buffer and in-core storage management. Assembler G doesn't spill anything to disk until main storage is full, whereas Assembler F does BSAM I/O directly. The originator of this innovation was Mr. Rennie Petersen, formerly of the University of Waterloo.

PERFORMANCE: The figures in the following table are for two jobs benchmarked in standalone mode on a 1-million-byte 360/75 at Waterloo. They are not as spectacular as some reports in the EDP press, but are more representative of what is commonly achievable. Space does not permit full definitions and documentation of the test in this report, but Waterloo's Computing Centre can provide full details. The table shows the real (wall-clock) time and non-HASP EXECP's (execute channel programs, or supervisor calls) required to assemble two programs (called ASMGF2 and LANDR) by means of IBM's Assembler F, Waterloo's Assembler G, and IBM's Assembler H in OS/MVT regions of from 64K to 500K bytes.

Region Size	Real Time, minutes			non-HASP EXECP's		
ASMGF2	F	G	H	F	G	H
64K	1.44	–	–	2900	–	–
80K	1.21	–	–	1781	–	–
88K	1.23	1.13	–	1801	2515	–
100K	1.19	1.10	–	1716	1401	–
150K	1.14	0.82	–	1552	996	–
176K	1.14	0.75	0.55	1552	776	398
200K	1.15	0.74	0.55	1552	708	391
250K	1.14	0.76	0.54	1552	698	389
300K	1.13	0.77	0.54	1552	688	388
400K	1.14	0.80	0.54	1552	667	388
500K	1.14	0.81	0.54	1552	628	360
LANDR	F	G	H	F	G	H
64K	8.72	–	–	18801	–	–
80K	8.26	–	–	12825	–	–
100K	7.81	–	–	11508	–	–
122K	7.79	1.66	–	11393	2496	–
150K	7.75	1.46	–	11258	1589	–
206K	7.77	1.32	–	11306	969	–
212K	7.78	1.32	1.65	11318	957	1152
250K	7.65	1.34	1.32	10944	950	800
300K	7.66	1.35	1.29	10944	938	769
400K	7.64	1.38	1.28	10944	817	754
500K	7.65	1.40	1.28	10945	892	747

In CPU time, ASMG required about 20% more than ASMH but about 20% less than ASMF throughout the benchmark tests. Using Large Core Storage (LCS), ASMG and ASMH values remained about the same, but ASMF's went up about 15%.

HARDWARE/SOFTWARE REQUIREMENTS: An IBM System/360 or 370 computer under OS, OS/VS, or CMS is required. The best performance will be achieved on a 360/65 or 370/155 or larger, using a partition or region of about 150K bytes. A 9-track 800- or 1600-bpi tape unit and either a 1403 Printer with a 62-character type-bar, or their equivalents, are required. The CPU must have the Universal (Commercial plus Scientific) Instruction Set. Under OS/PCP, OS/MFT, or OS/VS1, at least an 84K partition is required. Add about 15K more for the region size requirements for OS/MVT or OS/VS2. An 8000-line program under OS/MVT or OS/VS2 requires a region size of 116K, for example. (Of course, in VS2, memory only comes in 64K increments.)

PRICING: $300, plus a tape reel at least 1200 feet long.

INITIAL DELIVERY: March 1968.

CURRENT USERS: 285, including 108 in education and 177 commercial users. ■

WATBOL
University of Waterloo

WATBOL is an ANSI COBOL compiler for OS System/360 Model 40, System/370 Model 135, and larger IBM systems that features comprehensive diagnostic support and high-speed compilation. This in-core compiler can compile and execute successive programs in batch mode, with simplified job-to-job transition.

MANAGEMENT SUMMARY

WATBOL is upward compatible to Version 2, 3, or 4 ANSI COBOL as implemented by IBM. It was developed and is supplied by the University of Waterloo, of Ontario, Canada. WATBOL operates under OS in a minimum 130K-byte partition or region. Its supreme design objective was to be useful in debugging programs, with speed of compilation a closely-ranked secondary objective.

Although only one of the 75 current WATBOL users is a commercial installation (the rest are universities), the program has merits that should enable it to catch on among business users. These potential users should note, however, that WATBOL originates from and is supplied by an academic center, and not a commercial software vendor.

The University of Waterloo, consequently, will only undertake to provide the following services: (1) mail the program, on distribution media, plus documentation to the user, (2) maintain the user on a mailing list for notification about modifications, etc., and (3) provide advice by mail or telephone, without obligation to adapt the program to fit the needs of any individual.

WATBOL was designed to fit the needs of a university in teaching the concepts of solving problems using a computer. Its design features permit large numbers of novice programmers to use it, while it provides high-speed compilation, clear and comprehensive diagnostic and error messages, and simple job-to-job transaction (the new programmers need not know the OS Job Control Language). Provisions exist, however, for an experienced programmer to submit his own JCL and use his own files. Read-only files can also be created.

The implications behind this in-core compiler (which can be used in a batch mode) should be clear: While it is designed for instruction in an academic environment, it can be of considerable value in similar industrial and business environments, and to general users as an economical debugging and testing tool.

The compiler itself resides entirely in main storage. After generating the object code for aprogram, WATBOL loads the necessary run-time routines and initiates execution of the program. Test runs on a 360/75 indicate compilation rates in excess of 7000 source cards per CPU minute.

WATBOL has a number of useful user exit routines. It also has a few technical restrictions in terms of currently unsupported features of ANS COBOL. The unsupported features are enumerated in the Characteristics section.

CHARACTERISTICS

SUPPLIER: Supervisor of Technical Products and Program Distribution, Computing Centre, Math & Computer Building, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada, Telephone (519) 885-1211, Extension 3268.

BASIC FUNCTION: Provides fast COBOL compilation, in compile-and-run mode, with clear and extensive diagnostics and error messages. WATBOL can handle batched source programs, with fast, easy job-to-job transition. It generates reasonably efficient object code.

OPERATION: WATBOL is entirely core-resident. After it generates the object code for a program, it loads into main storage whatever run-time routines are needed for the program's execution. When used in batch mode, job-to-job transition is via simple control cards, relieving the novice programmer of the burden of learning OS JCL. An experienced programmer, however, can submit his own JCL and use his own files. Files can be made read-only.

The compiler was written in a special language called Z-1, and its source code is unavailable. However, source code is available for an accounting module written in BAL. The module also contains exit routines which are called by the compiler; it can also make available the compiler's current option table and statistics table, allowing the user to: (1) alter the control card format, (2) alter the format of statistics printed out at the end of a job, (3) produce new header pages, and (4) place the collected statistics of the compiler into an accounting record.

WATBOL is supplied on a University of Waterloo tape, using the IBM IEHMOVE utility, and includes the load module, run-time routines, a procedure library, an accounting interface source module, and a set of WATBOL test programs. Manuals are included to explain how to implement a WATBOL compiler at the particular installation using the supplied data sets. The manual also describes some WATBOL operating characteristics and contains some user-oriented materials for the WATBOL programmer.

The highlights of WATBOL are its clear diagnostic and error messages, its speed of compilation, and its ease of use. It also produces fairly efficient object code.

There are 16 features of IBM's ANSI COBOL that are currently unsupported in WATBOL. They are: Report Writer, REPLACING option of the COPY feature, special registers (CURRENT-DATE, TIME-OF-DAY, SORT REGISTERS), sequence number checking, insertion of the current date by the DATE COMPILED statement, RENAMES clause in structures, object deck punching, I/O control statements (RERUN and SAME-AREA, not implemented because of

WATBOL
University of Waterloo

USER REACTION

WATBOL users like the package's compilation speed and diagnostics. They especially like the fact that diagnostic messages appear in-place in the source code listing. Some users are running WATBOL in only 120K bytes of memory, which is less than the recommended minimum, with no apparent problems.

WATBOL was said by one OS/VS2 user to compile programs in as little as one-fifth the time required by the IBM compiler, due to the long I/O times under the OS/VS2 operating system.

On the other hand, there were some who expressed a wish that WATBOL had ISAM support, and that it were "as accurate a subset of COBOL as WATFIV, in effect, is of FORTRAN G."

Still, Datapro feels that WATBOL has excellent potential as a checkout compiler, a programmer training tool, or a one-shot COBOL job compiler for commercial use. It is a bargin at its price. □

different buffer management), ISAM, declarative section, tape label exit handlers, segmentation (which is not likely to be implemented in future versions), OCCURS DEPENDING ON, some debugging features (READY, TRACE, EXHIBIT, etc.), listing control features (EJECT, SPACE, etc.), and spanned records on files (RECORDING MODE S).

PERFORMANCE: WATBOL typically compiles about 7000 source cards per CPU minute on a System/370 Model 75.

The WATBOL messages are unusually clear. For example, indefinite execution time produces "Error 302, Execute-time exceeded. Program was executing line (n) in routine (name) when termination occurred." Similarly, an attempt to read after end-of-file has occurred results in "Error 312, attempt to read after end of file. Program was executing line (n) of routine (name) when termination occurred." Similar message types abound; e.g., "variable used before open," "direct access file is full," "file block size exceeds unit maximum," "insufficient memory for file buffers," and so on. The line being executed at termination and the program name are always given. Also presented are: compile and execution time, on both the CPU and wall-clock; cards read, lines printed, and pages printed during compile and execute; and the main memory available at entry to execution.

HARDWARE/SOFTWARE REQUIREMENTS: An OS System/370 or 370 (Model 40 minimum) with 256K bytes of main storage and at least a 130K partition in which to run WATBOL, a direct-access storage device, and a tape drive to restore the distributed tape to direct-access storage. The OS system includes the supervisor and data management macros and facilities, and the ability to handle the distribution tape formatted by the OS utility IEHMOVE. It is necessary to apply some "Superzaps" to the distributed WATBOL so that it will run under OS/VS1 or OS/VS2.

PRICING: WATBOL is available at two prices, depending on the class of user: academic or commercial. The term of use is a yearly lease costing $600 annually to the former and $1,200 annually to the latter. Third-party usage contracts are negotiated on an individual basis.

INITIAL DELIVERY: Test versions were in the field on September 1971. The first official release was in September 1972.

CURRENT USERS: 75 as of October 1973 (74 academic and 1 commercial). ■

WATFIV
University of Waterloo

MANAGEMENT SUMMARY

WATFIV is a fast, in-core FORTRAN compiler developed by the University of Waterloo as a successor to WATFOR. Its design objectives are aimed at filling the needs of personnel who regard the computer as a tool for solving their problems, but who have little programming experience. That is, programs must be compiled quickly, error diagnostics at compile and run time must be clear, and the mode of operation must be simple "load-and-go" (i.e., compile a program and execute it immediately). There must be no complex JCL between job steps; instead, batch-mode operation without knowledge of JCL must be possible. "Production" programs are often one-time solutions in such environments. Other WATFIV design objectives included generation of reasonably efficient object code, implementation of as much of full System/360 FORTRAN IV as feasible, efficiency of coding in "crucial" areas, and a load module of minimum size.

While these objectives stem from the needs of a university's students and faculty, they can easily apply to a business firm's engineering staff as well. Users should note, however, that the program comes from an academic center, not a commercial proprietary software vendor. The Unviersity of Waterloo undertakes only to: (1) mail the program on a user-supplied tape, along with documentation, (2) maintain a user mailing list for notifications, (3) provide advice about maintenance and debugging by mail or telephone, and (4) periodically distribute updates.

WATFOR was developed at the University of Waterloo in 1965 for its IBM 7040 computer. A version for the System/360 under OS was released in April 1967. WATFIV was released in March 1969, and in 1973 was extended to DOS and interactive CMS (Cambridge Monitor System) users. There are currently about 200 users, and about seven-eighths of these are in educational fields.

Popular usage of WATFIV among commercial users has been as a debugging tool, with debugged source code passed on to IBM's FORTRAN IV compilers for creation of production programs. The newly available interactive debugging capability under the Cambridge Monitor System should further popularize its use as a program testing tool. In addition, the "free I/O" concepts in WATFIV make it easy for clerical personnel to use it. Also, WATFIV has full facilities for experienced programmers, who can use their own JCL, access their own files, and compile their large jobs one at a time.

The power of the predecessor WATFOR compiler as a debugging tool was illustrated in an article by Dr. Stanley Siegal in the November 15, 1971 issue of *Datamation*. The

WATFIV, successor to the famous WATFOR, is an in-core, compile-and-run FORTRAN compiler for System/360 and 370 computers under DOS, OS, and their virtual storage counterparts, and for interactive use under CMS. It features fast compilation, comprehensive error diagnostics, and reasonably efficient object code.

CHARACTERISTICS

SUPPLIER: Supervisor of Technical Products and Program Distribution, Computing Centre, Math & Computer Building, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada. Telephone (519) 885-1211, Extension 3268.

BASIC FUNCTION: Fast FORTRAN compilation, with comprehensive compile and run-time diagnostics, on IBM System/360 and 370 computers under all operating systems. Implementation of System/360 FORTRAN IV, generation of reasonably efficient object code, special code efficiency in crucial areas, and load modules as small as possible are other WATFIV objectives. An additional main function is compile-and-run operation, with the ability to batch compilations and transit from job to job without use of JCL.

OPERATIONS: The compiler functions in batch mode, with job-to-job transition by means of simple control cards based on IBM 7040 IBSYS; for example, $JOB, $ENTRY. The experienced programmer, however, can use his own data sets and run one-program batches. WATFIV is a "load-and-go" compiler; no link-edit phase is involved, and execution of object code commences immediately upon the completion of compilation. The job-to-job transitions in a batch are also immediate, with the compiler returning to the operating system only at the end of the batch run.

WATFIV is implemented using the IBM IEHMOVE utility from the copy placed on the tape supplied by the user. Source decks, a macro library, object decks, a load module, a set of FORTRAN test decks, a listing of WATFIV error messages, and a procedure library that includes procedures for the assembly of source decks, link-edit of object decks, and other functions are supplied, together with an implementation guide.

The user has a choice of coded or full error messages. The latter requires more main storage. No object deck is produced by WATFIV.

Important FORTRAN extensions in WATFIV allow use of CHARACTER variables in logical expressions, multiple statements per card (using ALGOL-like colon and semicolon delimiters), ability to load object modules from disk or card, and the ability to call BAL programs. WATFIV isolates "system dependent" functions to one routine, and consolidates accounting facilities and $JOB card analysis to one source deck for easy modification. It improves upon

WATFIV
University of Waterloo

article gives examples of FORTRAN programs run on System/360 under four different levels of IBM FORTRAN (G, H0, H1, and H2) and WATFOR, showing how programs with various errors (e.g., undefined variables, variable subscript values that become improper, illegal branches, etc.) either compiled and ran (with varying results) or failed to compile (for unclear reasons) under the IBM compilers, but were "caught" and clearly diagnosed by WATFOR.

USER REACTION

WATFIV users praise the package's diagnostic capabilities, speed of compilation, and bug-free operation. Users state that WATFIV cuts debugging down by "about an order of magnitude." One called WATFIV "an indispensable debugging tool in our shop."

Speed is important to some, and users report that WATFIV compiles programs about 8 to 10 times faster than IBM's FORTRAN G and about 10 to 30 times faster than IBM's FORTRAN H. Execution under WATFIV is a bit slower, though. One user's analysis of WATFIV versus FORTRAN G with Symbolic Debug, and the tradeoffs, shows that WATFIV is a winner due to its debugging power. (Even though each run involves a compilation, this is not necessarily an unreasonable tradeoff on large programs, since programs that large take longer to debug, too. Besides, any program has to be recompiled if changes are made in it.) In this user's analysis, program size and compile time were irrelevant factors, and the degree to which a program is compute-bound negatively impacts WATFIV execution time.

WATFIV has also been available to clients of National CSS time-sharing services since February 1973. It has become a great favorite there, both with subscribers and within National CSS.

In summary, computer users with needs that match the problems addressed in this report can do themselves a favor by looking into WATFIV. □

the FORTRAN compiler in the search technique for tables and in the coding of DO and certain other statements. Patch areas have been inserted into WATFIV's communication areas, and certain modules reorganized, so that the times required for certain modules of the WATFIV compiler, as compared to WATFOR, can be reduced.

PERFORMANCE: On an IBM 360/75, source statements blocked at 10 cards per record and read from a 2314 disk were compiled at the rate of about 17,000 statements per minute in a benchmark test.

HARDWARE/SOFTWARE REQUIREMENTS: An IBM System/360 or 370 with at least 128K bytes of main storage, the Universal Instruction Set (i.e., floating-point and decimal arithmetic), one tape drive (to handle the distributed product), and at least two disks (2311's will suffice). A minimum of 74K and a maximum of 105K is required for WATFIV itself. The minimum compiler would have coded error messages, and would load library functions and I/O routines into the work area. The maximum compiler would have English error messages, and all routines included in the load module would be placed in main storage. Disk transient error messages are also available.

WATFIV was developed under OS, but it has already been modified to run under DOS, OS/VS1, OS/VS2, and VM/370, and is currently being implemented under DOS/VS. It also runs under CMS (Cambridge Monitor System) with execution-time monitoring facilities and under TSO. Machine requirements and functional characteristics for WATFIV are identical under OS and DOS. BAL source code is provided.

PRICING: WATFIV cost educational users only a $500 one-time charge. For commercial users, it must be leased yearly at $1,200 per year, and third-party use is negotiable on an individual installation basis.

INITIAL DELIVERY: WATFOR was released for the IBM 7040 in 1965 and for the System/360 in April 1967. WATFIV was released in March 1969, and was extended to DOS use and provided with CMS interactive execution-time debugging in 1973.

CURRENT USERS: 198 as of October 1973; 173 educational and 25 commercial. ■

Computer Scheduling and Control System Value Computing, Inc.

MANAGEMENT SUMMARY

The Computer Scheduling and Control System–$(CS)^2$–is actually a rather wide-ranging family of administrative computer control programs that aim at getting more value out of a computer system. The $(CS)^2$ product actually began with a single-machine scheduler–System III–that was initially installed in November 1969. At that time, Value Computing's scheduling system–an outgrowth of OPS (On-Line Performance Scheduler)–was one of the very few commercially available schedulers.

Although the necessary inputs to System III originally had to be manually prepared, they were almost completely derivable from the time accounting data captured by standard operating system logging routines. The next logical development for $(CS)^2$, therefore, was automation of the input phase for the Scheduler, and that was accomplished by System I, first delivered late in 1970.

System I builds a data base that includes, as one of its most important elements, a historical summary of each individual job's computer resource requirements. These job "profiles" are updated periodically to reflect recent activity and form an excellent basis for characterizing a job for scheduling purposes. Another important function of System I is to produce a series of utilization reports that summarize the system's throughput activity.

Subsequent development work on $(CS)^2$ included the Billing module–System II–and the Multiple Machine Scheduler; both were delivered during 1972. The most recent addition to $(CS)^2$ is the Tape Library Monitor, released in January 1973.

Each of these $(CS)^2$ modules can be considered on its own merits, but the most distinctive aspect of $(CS)^2$ is its scheduling capability. Scheduling yields maximum improvements in effectiveness for large computer installations where a greater quantity of resources offers a greater potential to improve the operating efficiency. This is particularly true in heavily production-oriented environments with numerous, stringent time-in/time-out demands. In these cases, the fundamental concept employed by the Value Computing scheduler is that of using the computer to drive the computer. Complex algorithms using a series of mathematical models are run to form a coordinated schedule that matches an appropriate mix of I/O-bound and CPU-bound programs in a multiprogramming system to maximize throughput. Of course, the system must also preserve predecessor/successor relationships and take account of I/O device assignment requirements when building the schedule. In fact, it is the exhaustive number of alternative schedule permutations that dictates the need for a computer to ▷

$(CS)^2$ is a useful collection of batch-oriented administrative control programs to handle job accounting, billing, and scheduling, as well as a tape library monitor system for control of physical tape reels. An important attribute of the system is its ability to handle multiple-machine scheduling.

CHARACTERISTICS

SUPPLIER: Value Computing, Inc., 496 North Kings Highway, Cherry Hill, New Jersey 08034. Telephone (609) 667-8770.

BASIC FUNCTION: The Computer Scheduling and Control System–or $(CS)^2$ as it is called by Value Computing–is a comprehensive set of five batch-oriented job accounting, billing, single/multiple-machine scheduling, and tape library monitor programs. The programs are written in COBOL for operation under IBM System/360 or 370 DOS or OS (or their VS counterparts) or UNIVAC Series 70 TDOS or DOS. About 40 pre-formatted reports are prepared through these modules, including all of the standard reports ordinarily used to manage a typical commercial computer installation.

OPERATION: The complete $(CS)^2$ system consists of five distinct programs, two of which are available separately (System I and the Tape Library Monitor). Four of these programs are generally complementary to one another in a sort of ascending hierarchy (Systems I, II, III and the Multiple Machine Scheduler), and the Tape Library system is basically independent of the other $(CS)^2$ programs. A combined version of System I with an enhanced System II is called Comput-A-Charge. All systems except the Tape Library program use the same fundamental accounting data base developed by System I, and System I is a prerequisite for installation of Systems II, III, and the Multiple Machine Scheduler.

SYSTEM I: As a cornerstone for most of the $(CS)^2$ programs, System I is a job accounting system that interfaces with the time accounting or log data captured by IBM's System Monitor Facility (SMF) for OS or VS; or the Job Accounting Interface (JAI) in DOS or DOS/VS. System I ordinarily takes about five minutes to run as a batch program and is typically run once a day to capture the basic computer time accounting data.

The data base established and maintained by System I is called the Workload Control File, a disk-resident file that contains information valuable for scheduling, including descriptions of the computer configuration(s), applications, and run models. This information is based upon historical run profiles, and can be altered directly by the user to reflect non-standard workload projections, etc.

Output from the System I run includes the following:

- Updated Workload Control File
- Daily Performance Reports showing shift summaries, multiprogramming graphs, detailed run reports, idle time report, and an equipment malfunction report. ►

schedule itself, using effective algorithms and the job characteristics tables.

The result of the Value Computing schedule run is a printout that identifies which jobs should be run in which partitions/regions and at what time. In order for an effective throughput improvement to be experienced, however, the operator must adhere to the schedules, and the schedule itself must be valid. (It is no great trick to set up an "impossible" schedule using bad input in terms of job requirements, etc., or to apply poor algorithms that do not yield effective results.)

To this end, users of (CS)^2 report that scheduling is handled most admirably by System III or its multiple-machine version, and that the entire (CS)^2 system has been remarkably free of bugs. One user reports that the acquisition of an additional multiprogramming processor was staved off indefinitely through the use of (CS)^2 scheduling, since throughput on the existing system was improved to the point that an expanded workload could be accommodated with what previously had been a fully loaded computer system.

Users of other (CS)^2 modules also report significant benefits; several System I users, for example, have been able to identify excess peripheral capacity and have compensated by returning a tape drive to the vendor.

Overall, (CS)^2 is a legitimate system well worth investigation by medium-to-large computer installations, which are likely to find one or more of its modules to be of considerable interest. □

- Monthly/Quarterly/Year-to-Date Reports, with At-A-Glance Device Usage and Malfunction Report, Device Exception Report, Rerun Summary, Major and Minor Accounting, Major Application Graph, and System Usage Report.

SYSTEM II: This is the job costing system and will ordinarily be run at the same time as the end-of-month System I job. Billing can be produced for a job using any or all of the following elements of resource utilization: CPU time, main memory used, cards read, lines printed, and EXCP's by class of device (I/O device utilization), including page in/out activity for virtual storage. Output from System II includes the following:

- Run Utilization Report presenting a cumulative picture of job activity during the preceding month broken out by job ID, resource utilization, and job charges.
- Application Utilization Report giving up to three levels of accounting for each major application system by detailed job, minor application subsystem, and overall application system.
- Cost Feedback Report providing the operations management with a breakdown of charges distributed to individual resources. This report can be used to adjust billing rates and/or the fundamental structure of the charging algorithm.
- Major Application Cost Graph illustrating cost distribution to major users.

An enhanced version of System II is also available at an additional charge for use in conjunction with System I. This combination is called Comput-A-Charge. Specific enhancements to System II consist of the following:

- Daily costing in addition to month-to-date and year-to-date summaries.
- Provision for different shift/priority billing.
- More detailed I/O device billing, including partial disk pack utilization, etc.
- Time Sharing Option (TSO) accounting, including connect time, line in/out counts, and concurrent user graphs.
- Paging usage graph showing the paging rate as a function of time.
- Program frequency counts for up to 50 program names.
- Historical profiles showing system utilization data by application over a 12-month period.

SYSTEM III: The System III Job Scheduler uses the Workload Control File produced by System I to develop a "best possible fit" of a projected input job stream against a set of available computer resources. A complex algorithm matches priorities and job profiles (maintained as historical data in the Workload Control File) to forecast an optimal schedule in a multiprogramming environment. System III can be run daily or at the beginning of each shift to produce a new schedule as often as required. After each run of System I, the Workload Control File job profiles are updated, and a subsequent System III run can take advantage of this revised job characteristics information to schedule more effectively.

There are three major subsystems in the System III Scheduler:

- The SCHED Program accepts user-specified scheduling parameters, activates the scheduling algorithm, and produces the schedule with beginning and end times for each job plus individual-program main memory locations.
- The MAINT Program allows the user to add, modify, or delete program profiles in the Workload Control File. This maintenance capability permits the user to update run models to "tune" the SCHED program output.
- A long-range scheduling program–SCHED 31– can forecast up to a month's upcoming workload, although not to the same level of detail as the daily (or shift) scheduling run.

Output from System III is also distinguishable by subsystem. The SCHED program produces three reports:

Computer Scheduling and Control System
Value Computing, Inc.

- Load List with all runs identified.
- Status Report that lists all jobs with run model data contained in the Workload Control File. These values can be temporarily changed for the current scheduling run.
- The Schedule itself, which shows just what jobs should be run in what partition/region and at what time. Excess resources are identified on a time basis to assist in running "walk-in" or unscheduled jobs as required.

The MAINT program produces a confirmation list of changes required in the data base by the user, and an Actual Versus Scheduled Report to measure the accuracy of the values used in the scheduling algorithm.

The Long-Range Scheduler produces a summary report showing projected total run time, average main memory usage, total number of runs, and time and percentage of utilization for each type of peripheral device.

MULTIPLE MACHINE SCHEDULER: This program can schedule a composite workload for as many as 15 computer systems. Each job is defined to have a "primary" or preferred machine, and can be run on up to four "secondary" or alternate systems. The net effect of using the Multiple Machine Schedule is to produce the most efficient possible schedule given a homogenous input stream and a distributed source of resources. Obviously, the scheduling algorithm is considerably more complex than that in a single-machine scheduler, but considerably greater overall system efficiency can be achieved with the greater degree of flexibility and the potential to fill schedule "gaps" more effectively.

TAPE LIBRARY MONITOR: The final component of $(CS)^2$ is the Tape Library Monitor System. This independent system handles the administrative functions of maintaining the library, including inventory reports and other operational record-keeping. Input can be either the Workload Control File produced by System I, or SMF accounting information plus a limited amount of manual manipulation. Four program modules comprise the Tape Library Monitor System:

- Tape Library Maintenance Program: Creates, modifies, or deletes tape library information in the Workload Control File.
- Daily Tape Feedback Program: Accepts current status information on newly created or deleted tape volumes by serial number.
- Daily Tape Procedure Program: Interfaces with System III (if in use) and produces a setup list in the proper time sequence for the scheduled runs. If System III is not used, this list will appear in a job-name sequence.
- Tape Report Program: Produces a number of administrative reports for management purposes upon demand.

Among the reports produced by the Tape Library Monitor System are a Daily Tape Retention Report, Daily Scratch Tape Listing, Tape Pulling List (including a check-digit auditing capability), Label List, Tape Inventory Report, Tape Retention Report, Off-Site Storage Report, Test Tape Report, and Data Set Audit.

HARDWARE/SOFTWARE REQUIREMENTS: All of the programs in the $(CS)^2$ system are designed to operate in batch mode on an IBM System/360 or 370 under DOS or OS (or their VS counterparts). The $(CS)^2$ system can also be run on UNIVAC Series 70 equipment under TDOS or DOS. Because the system is batch-oriented, it imposes no resident main memory requirement during normal production-mode operation. The individual modules of $(CS)^2$ can periodically be loaded for execution as required; the size of the largest $(CS)^2$ module–System III–is 90K bytes. In addition to main memory, the $(CS)^2$ system requires one magnetic tape drive, about 50 cylinders of 2319-type disk storage (on-line only during execution of $(CS)^2$), and access to a card reader and line printer (or other spool in/out devices).

PRICING: All elements of the $(CS)^2$ system are available on a perpetual license basis, and Systems I and II are also available on a lease basis. All modules except System III and the Multiple Machine Scheduler offer a 25% reduction in license cost for additional physical installations, and there is a 50% reduction for additional copies of System III at a single physical location. All additional-site discounts apply for 1 year only from the date of initial installation. Quantity discounts are negotiable for the Multiple Machine Scheduler. An upgrade for $(CS)^2$ module from DOS to OS can be obtained for the difference in license charges; for System I, an upgrade from DOS to OS costs $1,000.

	Perpetual License	Annual Maintenance/ Enhancement*	Support (man/ days)
System I – Job Accounting	$4,000	$480	2
System II – Billing	2,000	240	2
Enhanced System II – Billing	3,500	420	2
Comput-A-Charge – Job Accounting plus Billing	7,500	900	4
System III – Single-Machine Scheduling:			
DOS (or DOS/VS)	7,000	840	10
OS/MFT (or VS/1)	10,000	1,200	10
OS/MVT (or VS/2)	12,000	1,444	10
Multiple Machine Scheduler:			
DOS (or DOS/VS)	20,000	2,000	20
OS (or VS)	25,000	2,500	20
Tape Library Monitor:			
DOS (or DOS/VS)	4,000	480	5
OS (or VS)	6,000	720	5

*Maintenance during the first year is provided at no additional charge.

FIRST INSTALLATION: System I – November 1970; System II – September 1971; System III – November 1969; Multiple Machine Scheduler – February 1972; Tape Library Monitor System – January 1973.

CURRENT USERS: System I – about 70; System II – about 50; System III – about 40; Multiple Machine Scheduler – 5. ■

DOS Dump/Restore/Copy
Westinghouse Tele-Computer Systems Corporation

MANAGEMENT SUMMARY

As every IBM DOS user knows, a major part of the work of running his System/360 or 370 includes the use of a variety of disk utility programs. For instance, when system or private libraries have to be purged of obsolete files, or when backup of important files or volumes is required, or when the scheme of file residence on disk has to be reallocated for any reason, a utility operation involving tape-to-disk, disk-to-tape, or disk-to-disk data transfer has to be performed. These nonproductive tasks can consume a significant portion of overall system resources: typically from 10 to 20 percent for small to large-size installations, respectively, while on-line systems may spend up to 35 percent of their resources upon such operations.

IBM provides numerous utility routines to perform these backup-related functions, all for no additional charge. One group consists of IBM Type III Prior Use programs that have no free maintenance. Another category includes standalone utilities that require the system to be shut down for their use and require a re-IPL to resume DOS operations. The third and most competitive group of utilities consists of programs that perform many of the basic functions of the Westinghouse package and are standard system components.

These IBM utilities, however, are rather slow, inflexible, and use large quantities of magnetic tape. The Westinghouse package was originally written to replace these IBM utilities in 25 Westinghouse divisions operating under DOS. After two years of internal use, the package was made available commercially and has enjoyed considerable success, as evidenced by more than 850 installations to date. Typical performance improvements offered by the Westinghouse programs over their IBM counterparts are throughput increases ranging from 200 to nearly 500 percent.

Of particular note is the Westinghouse capability to relocate files during a tape-to-disk or disk-to-disk copy. This very helpful feature is not found in the standard IBM utility packages.

Users contacted by Datapro were very favorably impressed with the package, and many rated it as their best software purchase. One complication was mentioned, however, in conjunction with the use of the package: the lack of a standalone tape-to-disk program. Since tape backup prepared by the Westinghouse program can be restored to disk only by the appropriate Westinghouse tape-to-disk program, when a disk failure is experienced that takes down the system packs, the system cannot be backed up from tape (since the system itself is no longer functioning). Thus, system packs still have to be backed ▷

DOS Dump/Restore/Copy is one of the most successful software packages, with more than 850 current installations. As a major improvement over IBM's no-charge DOS utilities for disk-to-tape, tape-to-disk, or disk-to-disk data transfers, the system promises significant advantages to DOS or DOS/VS installations.

CHARACTERISTICS

SUPPLIER: Westinghouse Tele-Computer Systems Corporation (a division of Westinghouse Electric Corporation), 2040 Ardmore Boulevard, Pittsburgh, Pennsylvania 15221. Telephone (412) 256-5584.

BASIC FUNCTION: The Dump/Restore/Copy program is a disk utility system designed to provide IBM System/360 or 370 DOS users with a fast, easy-to-use method of backing up disk-resident files on tape or disk.

OPERATION: The Dump/Restore/Copy utility consists of three self-relocatable programs, which provide the following basic capabilities:

- Disk to Tape (DT) Dump allows multiple sequential (BSAM), indexed sequential (ISAM), or direct access (BDAM) files or entire disk volumes to be dumped to tape. Either multi-file volumes or multi-volume files can be handled. ISAM file reorganization can be specified to take place during the dump process. All disk areas that are not to be reorganized are dumped in track-physical format, including the physical address of each track, the data portion of record zero, and the count-key-data areas of all records on the track placed in contiguous locations. Any type of track data configuration including erased tracks is allowed. ISAM files that are to be reorganized are dumped in logical record sequence by ascending key. Only the key and data areas of all the records on the file are dumped. Count areas, record zeros, and index areas are excluded. Twenty-five clearly identified diagnostics are provided.

- Tape to Disk (TD) Restore can selectively restore any file or volume dumped to tape, with both parity and wrong-length-record checking, As a default, the TD program restores all the files and volumes contained on the tape, while optional parameter cards can specify individual files, for copying. Reorganized ISAM files can be relocated anywhere on disk, but all other restored files must be returned to their original relative locations. Reorganized ISAM files are reloaded to disk with the Disk Verify option off to speed the process, and two disk I/O areas will be used if sufficient core is available. In this instance, file verification is performed by scanning the restored area only after the file has been completely reloaded. Forty-five clearly identified diagnostics are provided.

- Disk to Disk (DD) Copy allows complete volumes and/or files to be copied from disk to disk with relocation, including the copying of disk areas by cylinder and head addresses. Any combination of BSAM, BDAM, or ISAM files or entire disk volumes can be copied to another disk or—for portions of a disk volume—to another location on the same volume. The ►

DOS Dump/Restore/Copy
Westinghouse Tele-Computer Systems Corporation

▷ up on disk. Westinghouse is aware of this limitation and has announced a stand-alone tape-to-disk utility, for availability by early 1973, which will be provided at no additional charge to all current users of the utility package.

With the recent announcement of DOS/VS, certain minor modifications will have to be made to the Westinghouse programs to enable them to run in virtual mode on the System/370. These modifications (which will limit the length of the chained CCW strings) are planned, and the system will be available for use with DOS/VS. DOS (or planned DOS/VS) users are well advised to benchmark the Westinghouse programs and convince themselves that the system has the potential to improve their operations while paying for itself. □

➤ copied disk areas are completely verified with both a wrong-length-record check and a parity check following the copy. Any number of volumes and/or files can be handled during one run. Thirty-three clearly identified diagnostic messages are provided.

In addition to the diagnostic messages provided for each different program, thirty-two more diagnostic messages that can apply to any of the Dump/Restore /Copy programs are included in the package.

A number of restrictions or limitations are placed upon use of the DOS Dump/Restore/Copy utility, including the following:

- Overflow records built using the Record Overflow hardware feature are not supported. This prevents the WTSC programs from handling full volumes built with that feature to write data blocks larger than the physical track size. (This capability is not supported under standard DOS, and the user can only encounter this situation when working with disk packs created under OS.)
- The Tape to Disk (TD) program will accept any tapes produced by the Disk to Tape (DT) program; and the tapes produced by the DT program can be restored only by the TD program.
- ISAM files cannot be reorganized in track-physical mode by the Disk to Disk program. (ISAM file reorganization can be accomplished by dumping the file to tape and reorganizing it when restoring it to disk.)
- Standard labels only are used by the DT and TD programs.
- While sequential files can have any number of extents, ISAM and direct files must have 10 or fewer extents.
- The DOS System Residence File cannot be relocated during a disk-to-disk operation.

Each of the utility programs automatically adjusts itself to the system environment before execution. The initialization process for each program provides for the recognition and adaptation to the type(s) of disk devices available, as well as self-relocation by each program. Any parameter cards that must be provided to control the operation of the system are read by device-independent logical IOCS, thus allowing SYSRDR or SYSIN to be assigned to a card reader, disk, or magnetic tape.

PERFORMANCE: Typical time requirements are summarized below. In general, the Westinghouse utilities provide a throughput improvement of about 300 percent over the comparable IBM-supplied utilities, with Disk to Tape performance ranging up to nearly five times that of the IBM program.

	Typical Time
Disk to Tape Dump (DT) (entire 2314 volume to a 60KB, 9-track tape with tape and disk on separate channels)	10 minutes
Tape to Disk Restore (TD) (entire 2314 volume from a 60KB, 9-track tape with tape and disk on separate channels)	10 minutes
Disk to Disk Copy (DD):	
Full 2311 volume	5 minutes
Full 2314 volume	10 minutes

HARDWARE/SOFTWARE REQUIREMENTS: The Dump/Restore/Copy programs all run under IBM System/360 or 370 DOS and operate with the IBM 2311, 2314 or 3330 disk drives or equivalent devices. Minimum partition sizes are listed below, although larger areas are used effectively when available.

	2311	2314	3330
Disk to Tape	14K	16K	22K
Tape to Disk	12K	16K	22K
Disk to Disk	12K	14K	20K

PRICING: The full set of programs is available only as a group, for a one-time charge of $850. A multiple-site pricing plan gives unlimited, worldwide access to the package for $3,250. The perpetual license includes a three-year maintenance warranty and a copy of the source program at no additional charge.

FIRST INSTALLATION: Internally at Westinghouse in 1968; outside Westinghouse on a commercial basis in December 1970.

CURRENT INSTALLATIONS: About 850 worldwide. ■

Teleprocessing Interface System
Westinghouse Electric Corporation

MANAGEMENT SUMMARY

Westinghouse Electric Corporation, whose DOS Dump/Restore/Copy package is one of the best-liked software products in the EDP industry, markets several other packages designed to aid the DOS user. The Westinghouse Teleprocessing Interface System is presently a low-cost package for local IBM 2260 or 3270 display users with DOS System/360 or 370 computers. It will be available with remote display support in the first quarter of 1974, and versions of the local and remote support packages optimized for virtual storage operation will be available shortly thereafter.

Considering the rave reviews that users are giving the Teleprocessing Interface System and the reputation that Westinghouse has built with its DOS Dump/Restore/Copy and DOS Job Monitor, it seems reasonable to assume that later versions of the package will also bear significant user interest.

The package is not brand new. The 2260 version, marketed as the Westinghouse CRT Interface, was first installed at a customer's site in April 1971. Only the 3270 version is a latecomer; it was initially installed in February 1973. The root of the package is LOTUS (Local Terminal Users System), a no-charge IBM prior-use program. LOTUS was co-authored by Steve O'Donnell, who subsequently made the package available in an improved version through Westinghouse. The users, as we shall see, have some nice things to say about Westinghouse in general and O'Donnell in particular.

The Teleprocessing Interface System functions as a software interface between the terminals, the operating system, and user-written application programs. It provides functions required by an on-line display system that would otherwise have to be expensively designed, coded, and debugged. It consists of a monitor generated from supplied macros, external macros used in writing application programs, and an off-line program used to load predefined screen files on disk.

This 3K to 12K monitor can control up to 255 local terminal devices and application programs, performing all terminal input/output requirements. It can operate one terminal at a time (single-thread) or many terminals simultaneously (multi-thread). In multi-thread operation, displays are serviced on a priority basis, with only the one being serviced using memory (others are held in a disk queue). This technique is not as wasteful of memory as is multi-thread operation in IBM's DOS-Standard level of CICS, but probably not as fast. The DOS-Entry level version of CICS is a single-thread program. CICS is described in Report 70E-491-02.

Another core-saving feature of the Westinghouse program is the Roll-In-Out (RIO) feature. This allows multiple application programs to share a common execution area. Users can designate whether RIO is to be used or not. Additionally, a program can be designated as "read-

This display monitor facilitates the installation of CRT terminal-based systems. It is currently limited to local displays operating under IBM System/360 or 370 DOS, but remote and virtual storage versions will be available shortly.

CHARACTERISTICS

SUPPLIER: Westinghouse Electric Corporation, 2040 Ardmore Boulevard, Pittsburgh, Pennsylvania 15221. Telephone: (412) 256-5583.

Westinghouse is a world-wide corporation.

BASIC FUNCTION: To serve as an interface between terminal hardware and operating system on the one hand, and user application programs and the operating system on the other, in a local-display IBM System/360 or 370 DOS system using IBM 2260 or 3270 display terminals or equivalent units. In this capacity, the package supports all terminal features, facilitates ease of use, provides some data management functions, and minimizes memory dedication. It also provides high-level language interfaces.

OPERATION: A monitor program is generated on-site, using supplied macros for selected options plus external macros used in writing application programs. An off-line program supplied with the package is used to maintain a disk file of predefined display screen formats.

Roll-In–Out (RIO) allows many applications programs to share a common execution area of memory. The feature can be selected or bypassed by the user and imposes no need for changes in application programs.

To minimize the time required to accomplish a Roll-In-Out cycle, disk storage addressing is direct and without indices, full track blocking is used, channel command words for disk reads and writes are chained, the user can waive write verification on roll-out, and programs designated "read-only" are not rolled out.

The Screens File facility allows hundreds of display formats to be placed on call by means of relative record number identification. Records on the file can be easily updated.

The Diskwork Facility provides the capability to manipulate created data strings with tape-like commands (e.g., move forward and backward through data strings), and enables the system to be used for data entry. The facility has dynamic disk allocation and deallocation and return. Warm-start initialization is also featured.

A Wait On User-Defined Event (WOUDE) function permits application programs to communicate with asynchronous subtasks while making wait time available to the monitor. If all file I/O requests are under control of the terminal monitor, this time can be utilized for system purposes, thus increasing the system's usage efficiency. To use WOUDE, the user writes file handling and data base subroutines and attaches these as subtasks with a lower priority than the monitor. The net effect is that file handling and data base requests are completely overlapped with other processing by the monitor.

The system is modular, and terminals and applications can be added without reprogramming all previous work. The new terminals can access current applications, and new applications can access all terminals. Selected terminals and/or operators can be locked out from the system.

Teleprocessing Interface System
Westinghouse Electric Corporation

▷ only," in which case it never has to be rolled out, thus saving execution time and memory cycles.

The off-line Screens File program places all required display formats on disk, where they can be called from the terminals. Application programs can access variable-length display formats by supplying a simple relative record number identification.

The data management scheme of the Teleprocessing Interface System is suitable for data entry. Westinghouse calls this the Diskwork Facility. It allows an application program to create a string of data records of user-specified length and attach these to a terminal. The string is controlled by tape-like commands; the program can move forward and backward through the string. Disk tracks are dynamically assigned, and are returned to the system for re-use as available. A warm-start facility allows the user to add records to an existing string without having to create a new string.

USER REACTION

Datapro contacted serveral users of both the 2260 and the 3270 versions. In these contacts, only one complaint was heard: one user of the 2260 version said that the documentation was not crystal-clear the first time through.

Users praised the ease of installation and the installation assistance they received from Westinghouse, especially the help from the system's author. One user took only 2½ days to generate the monitor, code two application programs, and generate all display formats after a brief contracted class and workshop. Within three weeks, before arrival of the terminal equipment, three additional application programs were coded and ready to run. At this writing, all five applications are running on a system on which the user says IBM hasn't even got the diagnostics working yet.

Rating the performance as excellent, one user gives this example: Using three 3270 terminals with 1920-byte screens, with the entire system shoe-horned into 15K bytes of memory and operating all displays simultaneously, the user's order entry application goes to a disk file, picks up the screen format, and writes on the 3270; the operator keys into blanks on-screen, and when the ENTER key is depressed, editing and writing is done onto a 2311 disk. The response time for all that is less than two seconds in the worst case.

Another user cites an important capability not found within CICS: the capability to add or delete displays while the system is running. Also, the user likes the choice of core-resident or RIO applications.

➤ Application programs can be written in COBOL (ANS and Level D), Assembler, or both combined. BOMP, DBOMP, ISAM, or DAM can be used. A design objective of the package was ease of use and installation requiring two days or less.

PERFORMANCE: The monitor's use of IBM's PIOCS (Physical I/O Control System) achieves performance that has earned user praise. Response time in various applications is generally less than two seconds, according to user interviews. When memory is dedicated (i.e., RIO not used), response is generally reported as being too quick to measure.

HARDWARE/SOFTWARE REQUIREMENTS: A System/360 or 370 local-display DOS system is required. Displays must be IBM 2260 or 3270 types, or their equivalents. The generated monitor requires from 3K to 12K bytes of memory.

PRICING: The package is vended at a one-time charge that covers a 3-year agreement. After the 3-year period, the package becomes the property of the user, so, in effect, the one-time charge is the purchase price. Source code and first-year maintenance are included. Training can be contracted for. Prices for various versions of the Teleprocessing Interface System are:

3270/2260, local support, DOS only	**$ 3,000**
3270/2260, local support, DOS/VS	**$ 4,000**
3270, remote support, DOS only	**$ 8,000**
3270, remote support, DOS/VS	**$10,000**

INITIAL DELIVERY: The 2260 version was first delivered in April 1971, and the 3270/2260 version in February 1973. The remote 3270 version will be available in the first quarter of 1974. Delivery dates for the virtual storage versions have not yet been announced, but are expected to be fairly soon.

CURRENT USERS: Approximately 35 installations outside of Westinghouse are using currently available versions. ■

The users Datapro interviewed had various experiences with other packages. CICS was rejected for price and core requirements, or for lack of facilities in the DOS-Entry level version. The Westinghouse package seems to be faster in response than the free versions of LOTUS and DUCS (Display Unit Control System). It is considerably less expensive than the local version of Minicomm (but remote versions will cost about the same).

According to one knowledgeable user, the first remote display version of the Teleprocessing Interface System will not be overly impressive, since it will depend on BTAM, but future releases will use the Physical I/O Control System (PIOCS) and be very impressive. This user says the vendor's, and particularly the author's, knowledge of PIOCS is the main reason for the excellent performance of the present package. □

DOS Job Monitor
Westinghouse Electric Corporation

MANAGEMENT SUMMARY

Not until DOS Release 25 did a procedure exist whereby the manager of a DOS installation could monitor the daily utilization of his System/360 or 370 computer. DOS release 25 provided a Job Accounting option in the supervisor which collected all job accounting information and placed it in a core area called the Job Accounting Table. But in order to use the information in the table, the user was forced to write a program called $JOBACCT to transfer the accounting data from core to a secondary device for subsequent processing. The Westinghouse DOS Job Monitor, a self-relocating program written using the IBM PIOCS (Physical I/O Control System) and capable of executing in one core image library block for maximum efficiency, performs this function. Moreover, the package produces reports designed to depict, on a meaningful basis, exactly what the computer is doing.

All that is required is a System/360 or 370 DOS or DOS/VS system with the Job Accounting macro, plus adequate disk space in which to place the collected accounting information. The disk space provided can be as small as one cylinder, but larger space permits collection of data for a longer, more meaningful period. In any event, the package can warn the operator that the job accounting file is a specified number of records from an end-of-file condition, so that the file can be scheduled for printing or backed up before it is overwritten and lost.

In addition to savings resulting from not having to write a job monitor in-house, the Westinghouse package presents the user with usage reports by partition and chronologically, charts memory usage by partition, presents a summarized and sorted report by phase name for each program showing the average and total time used and the frequency of program use, and compiles a 30-day summary report with daily data.

Westinghouse DOS software is becoming well-known throughout the EDP user community as a standard of excellence. In addition to the DOS Job Monitor, the firm offers two other important packages: DOS Dump/Restore/Copy (Report 70E-916-01) and DOS Teleprocessing Interface System (Report 70E-916-02).

USER REACTION

Datapro contacted large, multiple-computer System/370 multiprogramming installations and small 360/30 nonmultiprogramming users in its investigation into the worth of the DOS Job Monitor. It was determined that the package performs as advertised, installs easily, and is provided with outstanding vendor support. ▷

This useful routine handles the collection and presentation of job accounting information for IBM System/360 or 370 computers operating under DOS (Release 25 or later) or DOS/VS.

CHARACTERISTICS

SUPPLIER: Westinghouse Electric Corporation, 2040 Ardmore Boulevard, Pittsburgh, Pennsylvania 15221. Telephone: (412) 256-5583.

Westinghouse is a world-wide corporation.

BASIC FUNCTION: Collection, storage, and presentation of DOS job accounting information in chart, report, and summary format on System/360 or 370 computers using DOS, Release 25 or later, or DOS/VS.

OPERATION: The Westinghouse DOS Job Monitor provides the user with the $JOBACCT program called for by Job Control to gather data from the Job Accounting Table in core and place it on disk. Additionally, it subsequently retrieves the information from disk and generates output reports.

The user must specify disk space on the system residence pack, which can be as little as one cylinder, but will practically be two or three cylinders on most systems.

All accounting information related to a job (e.g., job name, phase name, start and stop time, elapsed time, DOS overhead, Start I/O commands issued, job termination codes, and idle time) is gathered by the supervisor and placed in the Job Accounting Table. When the job step is completed, Job Control issues a call for $JOBACCT, which places the information on the DOS system residence pack before returning control to Job Control to begin the next job step. The system can warn the operator when the space allocated on disk is about to be filled up (i.e., a specified number of records from the end of the file). The information must then be printed or moved to backup before the file is filled and begins to be overwritten.

PERFORMANCE: Users indicate (see the User Reaction section in the Management Summary) that the package performs as advertised, but that its output does not contain the totals usually requested by management. To produce these, most users write their own programs to process the package's output.

Power accounting can also present problems, but installations using standard job names can combine the information from Power's accounting feature with that from the DOS Job Monitor into a single comprehensive report.

HARDWARE/SOFTWARE REQUIREMENTS: A DOS system, Release 25 or above, or a DOS/VS system is required. The JA or Job Accounting macro must be specified (JA = YES). When MPS = BJF, 1090 bytes are added to the supervisor. When MPS = YES, 634 bytes are added to the supervisor. When MPS = NO, 771 bytes are added to the supervisor. At least one cylinder, but for practical purposes several, must be provided on the DOS system residence disk pack for accounting information to ►

DOS Job Monitor
Westinghouse Electric Corporation

be stored when retrieved from the Job Accounting Table in core.

PRICING: The DOS version costs $750 and the DOS/VS version costs $950 on a 3-year license agreement. After 3 years, the package effectively becomes the user's property. The price includes first-year maintenance. Multiple-copy discounts are available.

FIRST INSTALLATION: May 1972.

CURRENT USERS: 104 as of August 1973. ■

User interviews indicate that the package's output is seldom used directly for information presented to management. The feeling seems to be that the charts and reports it produces are best suited for the use of operations managers, with totals generally lacking. Users tend to run the output data through their own program to generate totals for management.

Datapro's queries about problems with the package turned up only DOS problems. Most notably, DOS provides interval timer information in places where wall-clock information is needed. Additionally, accounting information on Power, the IBM spooler, isn't handled properly unless Power is batched from disk JCL. Still, users of 3-partition machines with either Power or two teleprocessing programs, as well as non-multiprogramming system users, praised the package's output, the vendor support, and the documentation. (Westinghouse is working on a version that will interface Power.) One user was especially pleased that his program to work with the output data was made easier to write by the fact that the method of specifying the files in both COBOL and Assembler language was documented. □

SYNCSORT III
Whitlow Computer Systems, Inc.

MANAGEMENT SUMMARY

SYNCSORT III is the latest in a series of sorting product releases by Whitlow Computer Systems for the IBM System/360 and 370 computers. While retaining JCL and Sort Control Card compatibility with the IBM sorting products, SYNCSORT releases have introduced many innovations in sorting technology. Datapro has examined benchmark reports which show SYNCSORT's superior sorting performance increasing computer throughput in a congested sorting environment by up to 40 percent. Extra care has been taken so that the major features of SYNCSORT III can be implemented without any change in the user's current procedures. Performance improvements and features embedded in SYNCSORT III are applicable to all System/360 and 370 hardware, for sorting both fixed- and variable-length records in all OS, VS1, and VS2 environments with or without exit processing.

SYNCSORT III supports the COBOL and PL/1 sort verbs and can be linked to/from an Assembler program. It supports all random-access and tape devices as both input/output and working areas.

Whitlow Computer Systems employs a systematized approach to both the marketing and support of its SYNC-SORT products. A six-step sorting survey is offered to prospective users, beginning with a one-and-a-half hour technical presentation at the prospect's location. Second, Whitlow performs an analysis of the prospect's sorting load through a special SMF analysis program. Based on the user's SMF data, it shows the relative importance of sorting in his environment and classifies both the resources used and the type and size of the sorts being executed. It reports on such items as mode (invoked vs. JCL), I/O device mix, use of exits, work device type, distribution of sortworks assigned versus number of spindles used, record length distribution, block sizes, number of steps, etc. Third, based on the data obtained in step two, a test outline is described to reflect the specific requirements of the prospect's sorting environment. Fourth, an on-site demonstration is performed, and fifth, Whitlow tabulates the results and presents them to the prospect. The sixth step is a 30-day trial period. The result of this approach is a clear assessment of the value of SYNCSORT III in the user's environment. There is no charge for this service.

Whitlow guarantees that SYNCSORT III will outperform all competitive sorts. In most OS and OS/VS installations, the performance and operational features of SYNCSORT III should enable the package to cost-justify itself many times over. What's more, since SYNCSORT III uses significantly fewer system resources than most competitive sorts, the performance of other jobs which are executed simultaneously with sorting should also improve. ▷

SYNCSORT III is an IBM System/360 and 370 OS and OS/VS disk and tape sorting package that is compatible with the IBM sorting products. It is guaranteed to outperform all competitive sorts in either OS or OS/VS environments.

CHARACTERISTICS

SUPPLIER: Whitlow Computer Systems, Inc., 222 South Marginal Road, Fort Lee, New Jersey 07024. Telephone (201) 947-8500. In Europe, contact Pan-Data, 5JC Van Markenlaan, P.O. Box 166, Rijswijk ZH, The Netherlands, Whitlow's marketing agent. Marketing representatives are also located in South America, Japan, and Australia.

BASIC FUNCTION: Disk and tape sorting program for IBM System/360 and 370 computers. SYNCSORT III is believed to be the fastest available sort for either OS or OS/VS environments and is designed to cost-justify itself from both an operational and a performance standpoint. It is a compatible replacement for the IBM sorting products.

OPERATION: Although SYNCSORT III employs a new internal high-performance sorting technique, it retains all of the distinctive features and properties of previous SYNC-SORT packages and adds a number of additional features for both OS and OS/VS.

The package can be installed within 15 minutes, with no SYSGEN required. It has the same user exits and links as the IBM sorts. It is release-independent for OS/MFT, MVT, PCP, and all OS/VS systems; user interviews verify its successful use under MFT, MVT, VS1, and VS2.

SYNCSORT III's new high performance applies equally to all record types: fixed, variable, and spanned. It supports all IBM 2311, 2314, 3330, and 3340 and equivalent disk drives and all IBM and equivalent tape drives as input, output, and working storage. In addition, it can operate in a mixed random-access environment, supporting combinations of 2311, 2314, 3330 Mod 1, 3330 Mod 11, and 3340 disk drives during a single sort step. SYNCSORT III also provides COBOL and PL/1 sort verb compatibility.

Furthermore, SYNCSORT III contains the following sorting features:

- Selectable optimization modes—system performance can be optimized in terms of maximum multiprogramming throughput (M), minimized channel time (I), minimized EXCP's (E), or least elapsed time in a dedicated environment (D).
- Support of secondary allocation without inclusion in JCL—avoids "sort capacity exceeded" problems when sort space is underallocated.
- Release of excess disk space without inclusion in JCL—automatically returns overallocated disk space.
- Use of noncontiguous disk space—sorting functions can begin sooner.
- Reduced initial disk space requirement—can sort up to 100 percent more records.

➤

USER REACTION

Datapro conducted telephone interviews with 10 users of SYNCSORT III, equally divided between OS and OS/VS environments. There was no significant variation in the user responses based on their operating systems or machine configurations.

The results were as follows:

	Excellent	Good	Fair	Poor	WA*
Overall satisfaction	7	3	0	0	3.7
Throughput/efficiency	9	0	0	0	4.0
Ease of installation	6	4	0	0	3.6
Documentation	7	3	0	0	3.7
Vendor support	9	1	0	0	3.9
Training	2	7	0	0	3.2

*Weighted Average on a scale of 4.0 for Excellent.

These users represented the following IBM computer and operating system configurations: 370/158–3, 370/165–1, 370/168–3, 360/50–1, and 360/65–2; operating under OS–2, OS/MVT–2, OS/HASP–1, OS/VS1–2, and OS/VS2–3.

The system performed as advertised immediately for all 10 users. Eight of the 10 users reported a 30 to 40 percent reduction in their sorting times, especially for long sorts. Three of the users made modifications to the system, and four had the vendor make them. These were minor changes, such as fixes in their re-link and edit routines. One user reported that his tape sorting required the loading of a different program, but "...other than this, SYNCSORT III is an excellent package."

Overall, the users were enthusiastic about SYNCSORT III and the vendor's support. They mentioned such features as the package's support of secondary allocation without inclusion in the JCL. "In a production environment I have yet to encounter a 'sort space exceeded' condition," stated one user.

For any installation doing a great deal of sorting under OS or OS/VS, SYNCSORT III is one software package that virtually demands examination. □

- SYNCSIM–a program that can predict the resources utilized for a given sort run.
- MULTSIM–a repeatable multiprogramming simulator.
- HISTOGRM–a program that analyzes the characteristics of variable-length files.
- In-core and turn-around sort.
- Page-fix option–VS only.
- EXCP-VR option–VS only.
- Direct communication with invoked sorts.
- Record sizes up to 32K bytes in length.
- Compare Option (CLC or CPD)–use of the CPD option will detect erroneously specified data descriptions.
- Bias detection and utilization.
- Sort Efficiency Analysis Program.
- SMF-type statistical data record.
- "Debug" capability.
- Expanded message options.
- Separate core size options for JCL and invoked sorts.
- ABEND or Return Code 16.
- Merge (JCL and invoked).
- Variable-length record key check.
- Alternate dynamic device selection.
- Re-entrant code.

PERFORMANCE: Datapro has in its possession several copies of benchmarks from actual installed systems, and all show significant performance gains over the IBM sorts. However, since SYNCSORT III's performance can only be measured against selective objectives (usually user-defined), and since sort characteristics and system configurations can vary greatly, the best indicator for performance is user testimonials. Please refer to the User Reaction section of this report.

HARDWARE/SOFTWARE REQUIREMENTS: SYNCSORT III can handle both disk and tape sorts on any IBM computer system operating under OS or OS/VS. Scheduled for delivery in July 1975 is a DOS and DOS/VS version.

PRICING: SYNCSORT, complete with its ancillary programs, is available on a one-year or three-year license. The single-computer one-year price is $3,000 and the three-year price is $6,200. Additional licenses cost $2,700 for one year and $5,500 for three years. All prices include maintenance, updates, and new releases.

INITIAL DELIVERY: SYNCSORT I–January 1972; SYNCSORT II–January 1973; SYNCSORT III–September 1974; SYNCSORT III½–February 1975.

CURRENT USERS: Over 350 copies of SYNCSORT are currently installed worldwide. About 25 percent of these users are in Europe. The vendor claims an acceptance rate of about 90 percent among prospects testing the package. ■