IBM Application Program

Y20-0091-0

1130 Project Control System (1130-CP-05X)

System Manual

This manual describes the routines and subroutines that make up the IBM 1130 Project Control System. It is divided into the seven logical phases of the system and is intended primarily for the programmer who wishes to gain an understanding of the programming design of the 1130 PCS. It will provide him with the necessary information for maintaining and modifying the system if he so desires.

First Edition

Significant changes or additions to the specifications contained in this publication will be reported in subsequent revisions or Technical Newsletters.

Address comments concerning the contents of this publication to
IBM Technical Publications Department, 112 East Post Road, White Plains, N.Y. 10601

Introduction

This manual describes the routines that comprise the IBM 1130 Project Control System. It is intended primarily for the programmer who wishes to gain an understanding of the programming design of 1130 PCS. It will provide him with the necessary information for maintaining and modifying the system if he so desires.

Prior to reading this manual, the programmer should be thoroughly familiar with the information provided in the 1130 Project Control System Application Description Manual (Form H20-0211), and the 1130 PCS User Manual (Form H20-0342). In addition, he should be familiar with the basic language of the system which is 1130 FORTRAN, described in IBM 1130 FORTRAN Language (Form C26-5933).

The organization of this manual is basically modular. After a brief description of the program flow, the routines of the system are described within particular procedural or functional groups. Each routine described is presented as follows:

1. Routine name - that actually used to store the routine on disk.
2. The flowchart identification code for those charts where the logical flow for the routine is available.
3. The purpose of the routine.

4. The calling sequence. A root routine uses a CALL LINK sequence; a subroutine uses a CALL (parameters) sequence.
5. The input in the form of key data files or elements used by the routine. The format of the data files is described in the Disk File Data section of this manual.
6. A processing section which complements the flow charts.
7. The output produced by the routine described in the same form as used to describe the input.
8. A diagnostics section indicating the types of errors which the routine is capable of detecting.
9. The audit trail, if any, printed by the routine.
10. The subroutines used by the routine.

IBM 1130 PROJECT CONTROL SYSTEM

TABLE OF CONTENTS

Program Flow

The System flow of 1130 PCS is shown on page 10. The flow is governed by a control program, not shown in the figure.

1. Initialization consists of:

   a. reading the processing control and network card and setting up in the computer memory those cells needed to control later processing.

   b. calendar generation

   c. generation of resource code description and resource grouping tables.

   All tables are stored on disc. The control cells and calendar are kept in memory for the Input Processor.

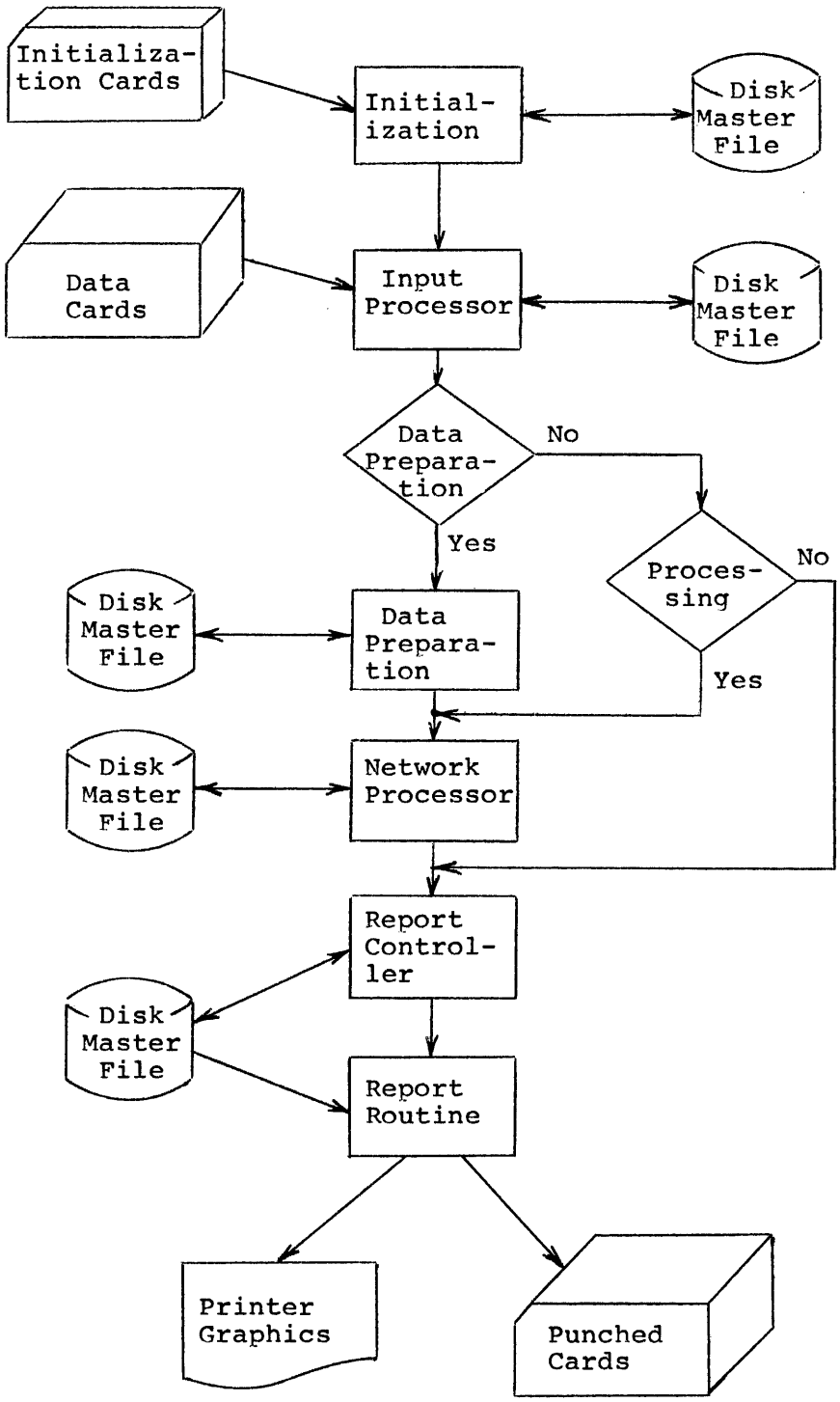
2. The Input Processor reads data cards, checks them for errors and then merges them with the disk master file for use by the Main Processor. Input is assumed to be in order by card control type. A special group header card signals the end of one data block, and tells the program what block of input to expect next.

   The Work Item card is the basic card for the entire system. If any data is given for a work item before its work item card is given, this data is rejected.

   The input processor calculates the current estimated duration for all in-progress work items, converts this to project days and computes all dates as elapsed times from the project base date.

IBM 1130 Project Control System

System Flow



3. Data Preparation incorporates both an Encoder (to translate PERT/CPM networks into Precedence networks), and a topological renumbering routine.

4. The Main Processor performs the network calculations.

   The logical modules included in the Main Processor are:

   a. Calculation of earliest start and finish dates.

   b. Calculation of latest start and finish dates.

   c. Calculation of float.

      The date and float calculations require that the calendar be in core memory so that work is not scheduled on Saturdays and Sundays or other defined weekends, holidays and special non-work days.

5. The Report Controller interprets the output request cards. Each report program called by the Report Controller collects its own data based on a file sorted by the Report Controller, prepares the report and exits back to the Report Controller. When all reports have been prepared, the Report Controller transfers control back to a control program which automatically reads the next network, or terminates the run, as directed by an end-of-file card.

   It is possible to update a master file and go directly to the output report controller, bypassing the network processor, as long as the logic of the network is not changed.

1130 PCS

CONTROL PROGRAM

PCS Bootstrap Routine Chart AA

Purpose

To initialize certain keys, or programmed switches, in the Communications Region and to call the Control Program. This routine serves as the bridge between the 1130 Monitor System and the IBM 1130 Project Control System.

Calling Sequence

This routine (PCS) is called through use of the 1130 Monitor System execute (XEQ) control card.

cc. 1-2: / /

cc. 4-6: XEQ

cc. 8-10: PCS

cc. 17: N (where N is the number of *FILES and *LOCAL cards required for the IBM 1130 Project Control System).

For more details on calling PCS, refer to the IBM 1130 Project Control System Operator's Guide.

Input

The Calendar disk file--this file is read in order to initialize the holiday and special non-work day list located in the Communications Region.

Processing

Refer to the flow chart

Output

None

Diagnostics

None

Audit Trail

None

Control Program Charts AB, AC, AD

Purpose

To call the 1130 PCS procedures required by any given run. This is accomplished by reading and interpreting the Process Control Card and setting the appropriate indicators in the Communications Region (COMMON). The Control Program also controls the calls for the various Initialization Procedure program modules that read and process card types B through F.

Calling Sequence

CALL LINK (CPGM)

Input

Card Type A--Processing Control Card

Card Type B--Network I. D. Card

Card Type C--Calendar Cards

Card Type D--Resource Description Cards

Card Type E--Resource Grouping Cards

Card Type F--Output Request Cards

All card types except A are optional dependant upon the type of run and the extent of processing desired.

Network I. D. disk file--this file is read only to initialize the Communications Region with network data that may or may not be used later in the processing.

Processing

The Control Program uses two keys (IDXCP, IX) to control the basic process flow. These keys are initialized by the bootstrap program (PCS) which sets them both equal to 1. The first phase of processing is the Initialization Phase which calls the program modules needed to read and process card types B through F. The second phase controls the calling sequence of the remaining procedures. These calls are controlled by Run Type, Network Type, and any processing or error flags set in the Communications Region by the various program modules prior to their return to the Control Program. If a fatal error (one that precludes the continuation of processing) has occurred, the Control Program will clear the card reader of any remaining input for this run, check to see if another PCS run follows, and if not exit to the 1130 Monitor. Refer to the flow charts for more details.

Output

For Network Generation Run Only:

Calendar File--Number of holidays and non workdays set to 0.

Resource Description File--Number of resources set to 0.

Resource Grouping File--Number of resource groups set to 0.

Directory File--First record set to 0.

This initializing process occurs prior to the first phase of processing.

Diagnostics

Error 12110: The first card of the input deck is not a Processing Control card. The error message "ERROR 12110

TYPE F" is printed.

Error 12111: There is no valid run type (either G, U, R, or Z) in the Processing Control card. The error message "ERROR 12111 TYPE F" is printed.

Error 12112: There is no valid network type (either CPM or blank) specified in the Processing Control card. The error message "ERROR 12112 TYPE F" is printed.

STOP ON LOOP ERROR This message is printed when a loop has been detected during Data Preparation. This is a fatal (TYPE F) error.

STOP ON NO START/FINISH WORK ITEM

This message is printed when no identifiable start or finish work item(s) has been detected during Data Preparation. This is a fatal error.

Audit Trail

END OF JOB - PCS is printed at end of job and before returning control to the 1130 Monitor.

Subroutines Used

None

1130 PCS

INITIALIZATION PROCEDURE

Initialization Procedure

Purpose

The Initialization Procedure is composed of the following routines:

NET - Network Card Routine

CAL - Calendar Card Routine

RESD - Resource Description Card Routine

RESG - Resource Grouping Card Routine

OR - Output Request Card Routine

This procedure reads and processes those cards which establish the network environment and which are not directly related to any of the specific activities that make up the network. The Initialization Procedure is not called as such but is rather a series of control tests and calls for the various routines that make it up. These tests and calls are integrated into the logic and coding of the Control Program and serve to control the linkage between the routines.

Calling Sequence

Refer to the Control Program (CPGM).

Input

Any of the Initialization data cards (card types A-F). These are:

Processing Control Card (card type A)

Network Title Card (card type B)

Calendar Card (card type C)

Resource Description Card (card type D)

Resource Grouping Card (card type E)

Output Request Card (card type F)

Each of these card types, except for the Processing Control Card (type A), are read as a group. Each group must be preceded by a group type header card which consists of the card type code in cc. 1, and an asterisk in cc. 6.

Processing

The linkage control for the various routines making up the Initialization Procedure is embodied within the Control Program. Refer to the Processing section of CPGM for details. Also refer to the Processing sections of the various routines that make up the Initialization Procedure.

Output

Updated disk files as follows:

Initialization file

Calendar file

Resource Description file

Resource Grouping file

Output Request file.

Updated elements within the Communications Region, most significantly--Project Base Date, Data Date, Network I. D. and processing control flags.

Diagnostics

Refer to the individual routines.

Audit Trail

Refer to the individual routines.

Subroutines Used

Refer to the individual routines.

NET — Chart BD, BC

Purpose

To read, process, and store the information contained within the Network Title Card (card type B) into the Initialization file.

Calling Sequence

CALL LINK(NET)

Input

Initialization data card type-B (Network Title Card). This card is required by every type of run except the Report run where it is optional. It must be preceded by a Group Type Header card with a B in cc. 1, and an asterisk in cc. 6. Only one type-B card is required per run.

Processing

After reading and optionally printing the B-type card, NET check for a "B" in cc. 1 of the card. By calling subroutine DTT, the Project Base Date (PBD), Data Date (DD) and Run Date (RD) are converted to elapsed days from 01 MAR 64. If the run is a Network Generation run, the modification code must be blank (cc. 6 of the B-type card). For all other runs, the modification code must be "C". In these runs, the Network I. D. from the card must equal that already on the initialization file, and the Data Date on the card must be $\geq$ than the Data Date on the file. Before returning control to CPGM, NET updates the Initialization file. All errors detected are fatal errors.

Output

An updated Initialization file.

Diagnostics

12002 - No network I. D. on input card (type-B). Error Type F.

12003 - Data Date is invalid. It is either missing from the B-type card, less than 01MAR64 or, if the modification code is "C", the card Data Date is Data Date on file. Error Type F.

12004 - The Project Base Date on the B-type card is either missing or less than 01MAR64. Error Type F.

12005 - The PBD is greater than the Data Date in a Maintenance (Run Type Z) run. Error Type F.

12006 - Invalid modification code on the B-type card. Must be either blank or "C". Error Type F.

12007 - The modification code is not blank for a Network Generation Run. Error type F.

12008 - With a modification code of "C", either the card network I.D. does not match that already on file or, the Data Date on the card is less than the Project Base Date on file. Error type F.

Audit Trail

"EXECUTE NET" is printed when entering this routine.

Subroutines Used

DTT

CAL Chart BE, BF

Purpose

To read and process C-type input cards (calendar cards). CAL updates the Calendar file.

Calling Sequence

CALL LINK (CAL)

Input

Cards of type C (Calendar Cards). A 1 punched in cc. 8 denotes a card containing holidays. A 2 punched in cc. 8 denotes a card containing special non-work days. The dates entered on the card(s) need not be in any order. A maximum of 480 dates of either holiday or special non-work day type can be entered.

Processing

After reading and optionally printing a C-type card, CAL tests for valid control information. The card is then scanned for dates and when found, subroutine DTT is called to convert the data to a day number equivalent to elapsed days from 01MAR64. Holidays are stored in the calendar as positive values and special non-work days as negative values. These day values are then sorted in ascending sequence. If the run is not a Network Generation run, a test is made for a match on network I. D. The calendar is stored both in COMMON and in the Calendar disk file. CAL returns control to either RESD, RESG, OR or CPGM routines.

Output

A new or an updated calendar both in the communications region (COMMON) and on the Calendar disk file.

Diagnostics

12010 - Modification code on the type C card is not either blank or "D". Error Type I. The next card is read.

12011 - Calendar card type code (cc. 8 of type C card) is not either 1 or 2. Error Type I. The next card is read.

12012 - The date punched into the card is either invalid or less than 01 MAR 64. Error Type I. The next date on the card is considered.

12013 - The Network I. D. code on the card does not match that on file for runs other than Network Generation. Error Type F.

12015 - An attempt was made to delete dates on a Network Generation run. Error Type I. The next card is read.

12016 - An attempt was made to either add dates already in the calendar, or delete dates not in the calendar. Error Type I. The next date on the card is considered.

12041 - The card read was not of type C and was not a group type header card. Error Type I. The next card is read.

Audit Trail

"EXECUTE CAL" is printed when entering this routine.

Subroutines Used

DTT

Note:

If the calendar has not been established during a Network Generation run, it can not be created during subsequent runs. To preclude this, a calendar card without any dates should be entered during the Network Generation run.

RESD — Chart BG, BH

Purpose

To read and process D-type input cards (resource description cards).

RESD updates the Resource Description file.

Calling Sequence

CALL LINK(RESD)

Input

Cards of type D (Resource Description Cards). Each card may contain up to three resource codes and descriptions. A maximum of 100 resources may be entered using a maximum of 100 D-type cards.

Processing

After reading and optionally printing a D-type card, RESD tests for either a blank or "D" modification code. RESD checks to see that resources to be added do not already exist on file, and that resources to be deleted are on the file. There is no check on network I. D. code. A test is also made to see that the number of resources does not exceed 100. RESD returns control to CPGM.

Otuput

An updated Resource Description file.

Diagnostics

12017 - The Modification code on the type D card is not either blank or "D".

Error Type I. The next card is read.

12018 - An attempt was made to add a resource to the Resource Description File when the number of resources is already 100. Error Type I. The next card is read.

12019 - An attempt was made to delete a resource that does not exist on the Resource Description file. Error Type I. The next resource on the card is considered.

12020 - An attempt was made to add a resource that already exists on the Resource Description file. Error Type I. The next resource on the card is considered.

12041 - The card read was not of type D and was not a group type header card. Error Type I. The next card is read.

Audit Trail

"EXECUTE RESD" is printed when entering this routine.

Subroutine Used

None.

RESG Chart BI, BJ

Purpose

To read and process E-type input cards (resource grouping cards). RESG updates the Resource Grouping file.

Calling Sequence

CALL LINK (RESG)

Input

Cards of Type E (Resource Grouping Cards). Each card may contain up to 10 resource codes, each of which must previously have been defined by use of card type D. A maximum of 19 resource groups may be created, each group having from 1 to 20 resources.

Processing

After reading and optionally printing an E-type card, RESG checks for the number of grouping codes already on file. In a Network Generation run, this number would be 0, and all E-type cards must have a blank modification code. RESG will then add new groupings to the file, checking each resource code for a match against the Resource Description file. For an Update run, if the modification code is "D", the grouping code indicated will be deleted from the Resource Grouping file only if no resource codes are punched into the E-type card. Otherwise, RESG will remove the indicated resource codes from the group.

RESG returns control to CPGM.

Output

An updated Resource Grouping file.

Diagnostics

12021 - An attempt was made to add more than 20 resources to a resource group. Error Type I. The next card is read.

12022 - An attempt was made to either add a resource code already in the indicated group, or to add a resource which is not in the Resource Description file. Error Type I. The next resource code in the card is considered.

12023 - An attempt was made to delete a resource code not in the indicated resource group. Error Type I. The next resource code in the card is considered.

12024 - An attempt was made to delete a Resource Group not in the Resource Grouping file. Error Type I. The next card is read.

12025 - The modification code in the E-type card is not either blank or "D". Error type I. The next card is read.

12026 - An attempt was made to add more than 19 resource groups. Error Type I. The next card is read.

12041 - The card read was not of type E and was not a group type header card. Error Type I. The next card is read.

Audit Trail

"EXECUTE RESG" is printed when entering this routine.

Subroutine Used

None

OR — Chart BK, BL

Purpose

To read and store F-type input cards (output request cards). OR stores the output requests on the Output Request file.

Calling Sequence

CALL LINK(OR)

Input

Cards of Type F (output Request Cards). A maximum of 16 output request cards may be used.

Processing

After reading and optionally printing on F-type card, OR checks to see that the network I. D. on the card matches that of the network in process. The following other validity checks are made: 1. Valid major output sort key (cc. 27); 2. Valid resource code (cc. 47-50); 3. Valid resource grouping code (cc. 44-45); 4. Valid span dates (cc. 11-25). OR checks for valid resource and resource grouping codes by seeking a match on the appropriate disk file. The span dates are checked by first converting them to elapsed days from 01 MAR 64 and comparing against the Project Base Date. A check is also made to see that the upper span date is greater than the lower. OR returns control to CPGM. For any type of run, the Output Request file is zeroed out so that new output requests must be entered for any run where output reports are desired.

Output

An Output Request file containing up to 16 output requests for the run in process.

Diagnostics

12027 - The network I. D. on the card does not match that of the network in process. Error Type I. The next card is read.

12028 - Invalid major output sort key in the F-type card (cc. 27). This key must be a numeric between 1 and 4. Error Type R. The key is set to 0.

12029 - Either the indicated resource code or resource grouping code on the F-type card does not match one on the resource files. Error Type R. The invalid code is set to 0.

12030 - An attempt was made to add more than 16 output requests per run. Error Type I. The next card is read.

12115 - The lower span date in cc. 11-17 is either invalid or less than Project Base Date. Error Type R. The lower span date is set equal to the Project Base Date.

12116 - An output request (cc. 53-80) does not contain either blank or a numeric. Error Type I. The invalid request code is set to 0

12117 - The upper span date is either invalid or equal to or less than the lower span date. Error Type R. The upper span date is set equal to 0, and the lower span date is set equal to the Project Base date.

12041 - The card read was not of type F and was not a group type header card. Error Type I. The next card is read.

Audit Trail

"EXECUTE OR" is printed when entering this routine.

Subroutines Used

DTT

1130 PCS

INPUT PROCESSING PROCEDURE

Input Processor

Purpose

The principal function of this procedure is to read and process all Input Data cards (types G through L) and establish or update the appropriate disk data files.

Calling Sequence

Refer to the calling sequence of the various modules making up the Input Processor.

Input

Card type G: Work Item Card

Card type H: Preceding Work Item Card

Card type I: Schedule Card

Card type J: Progress Reporting Card

Card type K: Resource Card

Card type L: Milestone Card

The input data must be grouped by card type, each card type being preceded by a Card Type Header Card (the card type code in cc. 1, and an asterisk in cc. 6). Type G cards must be the first group read. The remaining groups may be in any order. The last card of the input deck must be an end-of-file card which is blank except for an asterisk in cc. 6.

Processing

The Input Processor is composed of the following main routines:

1. Input Processor Control (INP)
2. Work Item Card Routines (WI & FWI)
3. Preceding Work Item Card Routines (PWI & FPWI)
4. Schedule Card Routine (SCHD)
5. Progress Reporting Card Routine (PRPT)
6. Resource Card Routine (RES)
7. Milestone Card Routine (MILE)

The Input Processor is called by the Control Program following the Initialization Procedure for either a Network Generation Run, an Updating Run or a Progress Reporting (Maintenance) Run. The routine actually called by the Control Program is INP which maintains control over the processing of all the input data card types before returning control to the Control Program. Processing details are contained within the descriptions of the various routines making up the Input Processor.

Output

If the input deck contains all possible card types, all the data files listed below will be established and made ready for processing by either Data Preparation or the Network Processor.

Work Item disk file

Preceding Work Item disk file

Processing disk file

Description disk file

Resource disk file

Milestone disk file

Directory disk file.

Diagnostics

Various flags are set in the Communications Region to indicate errors detected by the routines of the Input Processor.

Audit Trail

See each Routine

Subroutines Used

See each Routine

INP — Chart CA, CB, CC, CD, CE

Purpose

To control the processing of the input data cards through calls for the appropriate card routines.

Calling Sequence

```
CALL LINK(INP)
```

Input

There is no physical input to this routine. INP controls the processing of the cards through recognition of a card code transmitted to it by the Central Program or by the Input Processor's card routines. This code is the one contained in the Card Type Header Card.

Processing

When called by the Control Program, INP checks the run type and if a Network Generation Run is specified, the Directory disk file's first record is set to 0. The Control Program will only call INP if any of the card type codes processable by the Input Processor had been recognized during Initialization. Several restrictions on the combination of run type and card code exist:

1. Network Generation - Card type G must be present and must be the first card group read. Card type J is not valid for this type of run.
2. Updating Run - No restrictions.
3. Progress Reporting (Manintenance) Run - Card type J must be present and must be the only card group read. Other card types that follow the J group will be bypassed.

Upon recognition of card type, INP calls the appropriate routine which will process that card group. When the card routine recognizes another Card Type header, control is returned to INP along with the new card type code. If an end-of-file card is recognized, INP will return control to the Control Program.

Output

None

Diagnostics

12032 = First card code is not G in a Network Generation Run.
File organization is not possible and the run is aborted. Error type F. However, the card processing is continued in order to test for possible errors in the proposed deck.

12033 = Invalid card code either read by INP routine, or transmitted by any Input card routine (not G, H, J, K, L or 0 for the "end data card"). Error type I. The next card is read by the INP routine until a valid code is recognized.

12034 = Card code is not J in a Progress Reporting (Z) run (only progress reporting cards are valid).
Error type I. The next card is read by the INP routine until the J code is recognized.

12035 = Card code J in a network generation run (progress reporting cards are not valid). Error type I. The next card is read by the INP routine until a valid code is recognized.

12036 = Run Type code is not blank, U or Z.
Error type F. Control is returned to the Control Program.

12037 = Card code H in an IJ network (precedence cards are not valid).

Error type I. The next card is read by the INP routine until a valid code is recognized.

The following error messages do not relate to INP specifically but pertain to all the following routines:

WI, FWI, PWI, FPWI, SCHD, PRPT, RES, MILE.

They are located here for convenience only.

12041 = Card code change with asterisk missing. (Invalid Card Type header).

a) Card code blank: This card is not the "end-of-file card" and could be an unpunched card. Error type I. This card is bypassed and the next card is read by the current routine.

b) Any card code: This card is not a card group header card.

Error type I. Control is returned to the INP routine.

12042 = Invalid network identification.

a) No identification in the card: 1130 PCS assumes card is valid.

Error type R. The card processing is continued.

b) Any identification on the card.

It could be another network's card.

Error type I. The next card is ready by the current routine.

12043 = Invalid modification code on the card.

a) Not blank, C or D for WI or Milestone card;

b) Not blank or C for Schedule, Progress Reporting and Resource card;

c) Not blank or D for PWI card.

Error type I. The next card is ready by the current routine.

12044 = No WI identification on the card.

Error type I. The next card is read by the current routine.

12045 = Same identification on the card for I and J in a ij network.

Error type I. The next card is ready by the current routine.

12046 = Characters 5 and 6 of the WI identification on the card for an IJ network are not blank.

These characters are not considered by the Encoder.

Error type R. The card processing is continued.

12047 = WI missing on the master file.

Error type I. The next card is read by the current routine.

Audit Trail

"EXECUTE INP" is printed when entering this routine from the Control Program.

Subroutine Used

None.

WI (Work Item Card Routine) Chart CF, CG, CH

Purpose

To read and process G type input cards (work item cards). WI checks each card for possible errors and stores the data in a temporary disk work file for further processing by FWI.

Calling Sequence

CALL LINK(WI)

Input

Card Type G - Work Item Card. A maximum of 2000 work items are processable. A header card for the next card type group is read by the WI.

Processing

After reading and optionally printing a G-type card, the subroutine CNTRL checks the card code, the network I. D., the modification code and the work item identification for validity. WI also checks for valid data fields. Error messages are printed when applicable.

If a change in the external work item identification is called for, an indicator is set.

Original durations are converted into days by subroutine CNVDT.

All work item information is stored in a core buffer equivalent in size to one sector of disk work file (8 cards). When the buffer is full, it is written on disk. When all the cards have been read, the total number of records is set in the first record of the Directory file and the SORP1 routine is called. WI sets a flag in COMMON so that the FWI routine is called following the sort (by external WI code).

Output

Temporary work file for use by the sort routines.

Diagnostics

12050 = Same revised identification on the card for I and J in an IJ network.

The card processing is continued.

12051 = Invalid number of work days in the week.

This number will be assumed to be 5. Error type R. The card processing is continued.

12052 = Invalid WI calendar code.

This code will be assumed to be 0. Error type R. The card processing is continued.

12053 = Invalid conversion factor code.

This code will be assumed to be 0. Error type R. The card processing is continued.

12054 = Revised WI identification field completed without modification code C.

Error type I. The card processing is continued.

12055 = More than 2008 WI cards.

Error type I. Card reading is stopped, and the FWI routine is called.

Additional error messages possibly generated by WI can be found in the Diagnostics section of INP.

Audit Trail

"EXECUTE WI" is printed when entering this routine.

Subroutines Used

CNTRL, CNVDT

FWI Chart CI, CJ, CK

Purpose

To store the information contained in the work item cards (type G) in the appropriate disk master files. In the case of a work item deletion, FWI will restructure all necessary disk master files.

Calling Sequence

CALL LINK (FWI)

Input

The sorted disk work file created by WI. Also, the first record of the Directory File.

Processing

After reading the Directory File to ascertain the number of records resident in the work file, PWI reads and processes the temporary file sector by sector. After processing is complete, control is returned to INP.

The processing is as follows:

1. In a Generation Run, an internal work item number, starting at 1 with a maximum of 2000, is sequentially assigned to each unique work item. Duplicates are bypassed--see error 12057. In an update run, subroutine EWIF is called to check whether the work item is already on file. The work item data is then allocated to the various disk files in sequence by internal WI number. Thus, the internal work item number is equivalent to the record number within a file. In the case of a change or delete code, EWIF retrieves the

previously assigned internal work item number for the work item. Refer to EWIF description for more details.

2- When a work item is added, one record is created for it in the Directory, Processing, Work Item, Description and Resource Files. All unused areas in these records have been set to 0 or blank by CPGM during a Generation Run.

3- When a work item is to be deleted, all information pertaining to this work item is deleted from the Work Item, Preceding WI, Description, Resource, Milestone, Directory and Processing files. These files are compressed after each deletion. The following subroutines are used during deletion:

MPWIF	-	for PWI file
DMF	-	for Milestone file
DWIF	-	for WI file
DPF	-	for Processing file
DDESF	-	for Description file
DRF	-	for Resource file
DDF	-	for Directory file

4- When a change in duration, calendar code or any time is to be made, changes are made to the Processing, PWI and Description files. If the Estimated Duration of an in-progress

work item is changed, Percent Complete to Date is recalculated using the new duration and the amount of time already expended:

$$\%TD = \frac{O.D. - R.D.}{N.O.D.}$$

where time already expended = O. D. - R. D.

O. D. = Original Duration

R. P. = Remaining Duration

A check is made to ensure that $0.0000 \leq \%TD \leq 1.0000$

Furthermore, using the LPWIF subroutine, current lag values are modified if the relationship is in percentage mode and if:

a) The WI has a PWI with an end-to-end relationship

b) The WI is a predecessor with a start-to-start relationship

A modification code of C means that all existing information is replaced by that on the card. To retain information on the files, it must reappear on a card with a C modification code.

Output

Updated disk files: PWI file, Milestone file, WI file, Processing file, Description file, Resource file, Directory file.

If a work item is either added or deleted, FWI sets a flag (IP2) in COMMON equal to 1 to inform the Control Program that a change in the Network logic has occurred.

Diagnostics

12056 = Revised WI identification on the master file.

This revision is not assumed. Error type I.

The card processing is continued.

12057= WI already present on the master file

Error type I. The next record is processed.

12058 = More than 2000 WI records on the master file.

Error type I. The next card is read, and so on until another card code is recognized.

12059 = WI appears to be completed after original duration change.

a) Percent completion to date is set to 1;

b) Remaining duration is set to 0;

c) Current duration is set equal to the original duration.

Error type R. The card processing is continued, but all time information set on the master file needs to be updated by means of a progress reporting card.

Additional error messages possibly generated by FWI can be found in the Diagnostics section of INP.

Audit Trail

"EXECUTE FWI" is printed when entering this routine.

Subroutines Used

CNVDT, EWIF, DMF, LPWIF, MPWIF, DPF, DWIF, DDESF, DRF, DDF.

Note: A *LOCAL control card must be used if FWI is to be executed under the 1130 Monitor System.

PWI (Precedence Card Routine) Chart CL, CM, CN

Purpose

To read and process H-type input cards (precedence work item cards). PWI checks each card for possible errors and stores the data in a temporary disk work file for further processing by FPWI.

Calling Sequence

CALL LINK (PWI)

Input

Cards of type H (Precedence Work Item Card). A maximum of 2240 PWI cards can be read in one run (35 cylinders); however, the maximum number of relationships is 4500 for a network.

No special sequence is required, but substantial time savings can be made if the H-cards are in the same sequence as the WI (C-type) cards. This will permit:

1. No shifting of the PWI file (see APWIF subroutine)
2. A reduction in Directory file modifications (see MDF subroutine)

Processing

After reading and optionally printing an H-type card, the subroutine CNTRL checks the card code, the network I. D., the modification code and the work item identification for validity. The card is then scanned for a precedence work item and when found, an indicator is set to identify the type of relationship between this PWI and the work item.

a) If an End-to-Start relationship (code blank in the card), the lag value is always expressed in time units. This value is converted into days by the FPWI routine.

b) If a Start-to-Start relationship (code S in the card), the lag value is either a percentage of the PWI duration or in time units. This value is converted into days by the FPWI routine and stored as Lag Current Value.

c) If an End-to-End relationship (code F in the card), the lag value is either a percentage of the WI duration or in time units. This value is converted into days by the FPWI routine and stored as Lag Current Value.

Each work item may have none, one, or several preceding work items. There is no limit on the number of PWI's a WI may have. However between two work items, there can only be one relationship of any one type.

The start work item has no PWI and therefore has no record on the PWI file.

All PWI and information is stored in a core buffer equivalent to 1 sector of disk work file (8 cards). When the buffer is full, it is written on disk. When all cards have been processed, the total number of records is written in the first record of the Directory file and the ELOOK routine

is called.

Output

Temporary work file for use by FPWI, and ELOOK.

Descriptions

12060 = First digit of the 5 digit lag area is not 0 for lags of other than form P.

This digit is set to 0. Error type R.

The card processing is continued.

12061 = More than 2008 PWI cards

Error type I. Card reading is stopped, and the ELOOK routine is called.

12062 = Invalid relationship code.

This code is assumed to be blank. Error type R.

The card processing is continued.

12063 = Invalid form of lag code.

This code is assumed to be blank. Error type R.

The card processing is continued.

12064 = Form of lag code =P with a blank relationship code.

The first code is assumed to be blank. Error type R.

The card processing is continued.

12065 = Same identification on the card for WI and PWI.

Error type I. The next PWI on the card is processed.

Additional error messages possibly generated by PWI can be found in the Diagnostics section of INP.

Audit Trail

"EXECUTE PWI" is printed when entering this routine.

Subroutine used

CNTRL

ELOOK (External Work Item Number Replacement) Chart DA

Purpose

To replace external work item numbers on the work file generated by FWI with internal work item numbers and set the proper flags for FPWI to function properly.

Calling Sequence

This routine is a main line routine calledby FWI with a CALL LINK (ELOOK).

Input

The first two records of the directory file are used to get the number of work items in file 205 and the number of records on the work file. Work file 210 is used as input to this routine.

Processing

A portion of the Work Item file is read into core. One sector of the work file consisting of 8 39-word records is read into core. It is assumed that words 7, 18 and 29 of the 39-word record have been preset in the following manner. Word 7 is a special case since it contains the flag for both the work item and the first preceding work item. This word is preset to a 4 since the external work item is present, if there is a PWI in the first area 1 is added and the word contains 5. If there is no PWI specified in the first PWI field 3 is added and the word contains 7. Words 18 and 29 only function as flags for PWI fields 2 and 3 and either contain a 1 if a PWI is present or

3 if the PWI field is blank.

A search is made depending on the flags. If the external work item number is found in the table corresponding to File 205, the sixth word of the record is set to the internal work item number and 4 is subtracted from word 7. For the PWI fields if a 1 or 5 is in the flag a search will be made for the PWI in the table corresponding to File 205. If the PWI is found the internal WI number is inserted in word 12, 23 or 34 and 1 is subtracted from the flag in words 7, 18 or 29. After all records in File 210 and 205 have been matched, control passes to FPWI.

Output

The records in File 210 are changed as indicated in Processing.

Diagnostics

None

Audit Trail

ENTER ELOOK is printed on entering this routine.

Subroutines Used

None

FPWI Chart CO, CP, CQ

Purpose

To organize and store data concerning precedence relationships on the PWI disk file. In the case of a PWI deletion, the DPWIF subroutine is called to update the PWI and Directory files accordingly.

Calling Sequence

CALL LINK (FPWI)

Input

The temporary disk work file created by PWI. Also, the first record of the Directory file.

Processing

After reading the Directory file to ascertain the number of records resident in the work file, FPWI reads and processes the temporary file sector by sector. After processing is complete, control is returned to INP. Processing is as follows:

1. The subroutine EWIF is called to determine the internal work item numbers previously assigned to the work item and its PWI(s) that appeared on the original H-type card
2. If the modification code is a delete code (D), the DPWIF subroutine is called to update the PWI and Directory files.
3. If the modification code is not a D, the addition of a preceding work item (PWI) is called for. The lag is then

examined for the type of relationship and subroutine CNVDT is called to calculate the Lag Current Value. Subroutine APWIF is then called to add the PWI to the PWI file. The doublet WI-PWI and its associated data is inserted into the PWI file such that the file is always in PWI within WI internal number sequence.

In all cases, a flag (IP2) in COMMON is set equal to 1 to inform the Control Program that Topological ordering will be necessary.

Output

An updated PWI file in PWI within WI internal number sequence. The Directory file is updated with appropriate PWI file pointers.

Diagnostics

FPWI

12067 = PWI is not a WI on the master file.

Error type I. The next PWI in the record is processed.

12068 = More than 4500 PWI records on the master file.

a) In precedence network: Error type I. The next card is read, and so on until another card code is recognized.

b) In IJ network: Error type F. Control is returned to the Control Program (in this case, the last PWI record on the master file is printed following the error message).

12069 = PWI to be added already present on the master file.

Error type I. The next PWI in the record is processed.

12070 = PWI to delete missing on the master file.

Error type I. The next PWI in the record is processed.

Additional error messages possibly generated by FPWI can be found in the Diagnostics section of INP.

Audit Trail

"EXECUTE FPWI" is printed when entering this routine.

Subroutine Used

EWIF, CNVDT, DPWIF, APWIF

SCHD (Schedule Card Routine) Chart CR, CS

Purpose

To read and process I-type input cards (schedule cards). SCHD stores schedule date information on the Processing file and organization code information on the Resource file.

Calling Sequence

CALL LINK (SCHD)

Input

Cards of type I (Schedule Card). A maximum of one schedule card per work item is allowed. No special sequence is necessary within the I-type card group.

Processing

After reading and optionally printing an I-type card, the subroutine CNTRL checks the card code, the network I. D., the modification code and the work item identification for validity. The card is then scanned for start and finish schedule dates. If present, these dates are converted into elapsed times from OIMAR64 with the DTT subroutine. Schedule finish date must be $\geq$ schedule start date for the same work item. Error messages are printed where applicable.

The EWIF subroutine is called to test whether the schedule information may be assigned to an already existing work item. After the check, the Processing file is updated. The organization codes are then

read and the Resource file is updated.

Since only a blank or change modification code is allowed for an I-type card, any update run will replace all existing information with that from the card. To delete only a single field then, all other fields to be retained must be repunched.

After all I-type cards have been processed (a new card type code has been recognized), control is returned to INP.

Output

Updated Processing and Resource disk files.

Diagnostics

12071 = Invalid type of schedule start date

This type is assumed to be 1. Error type R.

The card processing is continued.

12072 = Invalid schedule start date.

This date is not assumed. Error type I .

The card processing is continued.

12073 = Invalid type of schedule finish date.

This type is assumed to be 1. Error type R.

The card processing is continued.

12074 = Invalid schedule finish date.

This date is not assumed. Error type I.

The card processing is continued.

12075 = Schedule finish date is not equal to nor greater than schedule start date.

No diagnostic possible. Error type R.

The card processing is continued.

Additional error messages possibly generated by SCHD can be found in the Diagnostics section of INP.

Audit Trail

"EXECUTE SCHD" is printed when entering this routine.

Subroutines Used

CNTRL, DTT, EWIF

PRPT (Program Reporting Card Routine)

Chart CT, CU, CV, CW

Purpose

To read and process J-type input cards (progress reporting cards). PRPT updates the Processing and Description files.

Calling Sequence

CALL LINK (PRPT)

Input

Cards of type J (Progress Reporting Cards). These cards may only be validly used in an Updating or Progress Reporting Run. They are only required for those work items actually in progress. They may contain either time or cost information or both

Processing

After readingand optionally printing a J-type card, the subroutine CNTRL checks the card code, the network I. D., the modification code and the work item identification for validity. All data on the card is analyzed and processed as detailed below. When a new card type code is recognized, control is returned to INP.

1. PRPT checks for run type. If not an Updating or Progress Reporting Run, an error message is printed and the card is bypassed..

2. If progress is being reported only in terms of time information, the time/cost code in cc. 6 must be blank. For time reporting,

the following combinations of data are possible:

a) Actual Start (AS) and either Remaining Duration (RD) or percent complete to date (%D) or percent complete this period (%P).

b) AS and Actual Finish (AF). Other information is ignored.

c) AF only.

d) Either RD or %D or %P.

Note: The DTT subroutine is called to convert actual dates into elapsed times, and CNVDT subroutine is called to convert estimated durations into days. The Data Date (DD) and Project Base Date (PBD) must reside in the communications region (COMMON) in terms of project days of the form XXXX.X.

3. Following is a description of the calculations performed when progress is reported on either %D, %P, or RD.

a) %D - This value is stored in the Description file. The corresponding remaining duration (RD) is calculated and stored in the Description file:

RD = (1.0-%D) OD

where OD = original duration

The current duration (CD) is calculated and stored in the Processing file:

CD = DD-AS+ 1 + RD

Note: if %D>1.0000 on the card, %D is set equal to 1.0000

b) %P - This value is stored in the Description file. This value is then added to the value of %D in the file and if %D > 1.0000, a message is printed and %D is set equal to 1.0000. The corresponding remaining duration (RD) is calculated and stored. The current duration is calculated and stored. (See %D calculations).

c) RD - This value and its conversion factor are stored in the Description file. The corresponding %D is calculated as follows:

$$\%D = \frac{OD-RD}{OD} \quad \text{if } RD < OD$$

$\%D = 0.0000$ if $RD \geq OD$ (a message is printed)

The current duration is calculated and stored in the Processing File.

d) Note: only one of the above values is needed. Their priority is indicated below:

I) If %D and %P are given, only %D is accepted and %P is ignored.

II) If RD and %P are given, only RD is accepted and %P is ignored.

III) If %D and RD are given, only RD is accepted and %D is ignored.

IV) If all three are given, only RD is accepted.

4. If AS and either RD or %D or %P are given, the value of RD is calculated (see 3 above) if necessary and the case is considered to be: AS and RD are given.

   - AS is stored unless AS > DD and then it is ignored and a message printed.
   - The value of current duration (CD) is calculated and stored in the Processing file. CD will not be calculated if RD=000.0 and %D=1.0000, and will retain its previous value.

     CD = DD - AS + 1.0 +RD

5. If AS and AF are given:

   - AS and AF are stored in the Processing file.
   - RD is set equal to 000.0
   - %D is set equal to 1.0000
   - CD is set to the value of O.D.

AS must be $\leq$ AF which must be $\leq$ DD. Otherwise, a message is printed and the values are ignored. All other time information on the card is ignored in this case.

6. If AF is given, it must be $\leq$ DD. Otherwise it is ignored and a message printed.

   - AF is stored on the Processing file
   - %D is set equal to 1.0000

- RD is set equal to 000.0

- CD is calculated as follows;

  a) If AS was previously given, then

     CD = AF - AS + 1.0

  b) if AS was not previously given, then

     CD = previous value of CD (if any)

7. Either RD, %D, or %P are given.

Refer to 3. above for processing details. However, the following cases are considered.

a) No previous progress had been reported for the work item. Then, RD. is calculated as previously described:

   i) If RD = 000.0 or if %D = 1.0000, no change is made to the value of CD and the message "NO ACTUAL DATES" is printed.

   ii) If RD>000.0 or if %D<1.0000, the value of AS can be assumed as follows:

       AS = DD - (OD - RD) + 1.0

       The message "NO ACTUAL START DATE" is printed. CD is calculated as follows:

       CD = DD - AS + 1.0 +RD

b) There is an Actual Start (AS) date already on the file. Then RD is calculated as previously described.

i.) If RD = 000.0 or %D = 1.0000 and there is no AF, CD remains unchanged and the message "NO ACTUAL FINISH DATE " is printed.

ii) If RD > 000.0 or %D < 1.0000, CD is calculated as follows:

$$CD = DD - AS + 1.0 + RD$$

8. If progress is being reported in terms of cost information, the time/cost code in cc. 6 can be either blank or C. For cost reporting, the following processing is performed.

a) If Actual Cost to Date (ACD) is given, it is stored in the Description file.

b) If Actual Cost this period (ACP) is given, it is stored and ACD is calculated as follows:

$$ACF = ACD(previous) + ACP$$

If both ACD and ACP are given, only ACD is accepted for processing and ACP is ignored.

Output

Updated Processing and Description files.

Diagnostics

12077: Actual finish date on the card is not equal to or greater than actual start date on the master file.

All time information set on the disk file needs to be revised.

Error type R. The card processing is continued.

12078 : No actual dates (either on the card, or on the master file) for a completed work item.

The current duration is not updated. Error type R. The card processing is continued.

12079 : No actual start date either on the card or on the master file.

a) For a completed work item (actual finish date is present on the card). In this case, the current duration is not updated.

b) For a started work item. An actual start date is estimated. Error type R.. The card processing is continued.

12080 : No actual finish date for a completed work item (actual start date is present either on the card or on the master file).

The current duration is not updated. Error type R.

The card processing is continued.

12081 : Invalid conversion factor code for remaining duration.

This code is assumed to be 0. Error type R. The card processing is continued.

12082 : Percentage completion this period (on the card) is greater than 1. This percentage is assumed to be 1. Error type R.

The card processing is continued.

12083 : Percentage completion to date is greater than 1. Either on the card or as calculated from percentage completion this period

on the card. This percentage is assumed to be 1. Error type R. The card processing is continued.

12084: No cost information on a card with time/cost code C.

Error type I. The next card is read.

12085 : Invalid actual start date on the card.

This date is not assumed. Error type I. The card processing is continued.

12086 : Actual start date on the card is greater than the data date.

This date is not assumed. Error type I. The card processing is continued.

12087 : Invalid actual finish date on the card.

This date is not assumed. Error type I. The card processing is continued.

12088 : Actual finish date on the card is greater than the data date.

This date is set equal to the data date. Error type R. The card processing is continued.

12089 : Actual start date on the card is greater than actual finish date on the card.

These dates are not assumed. Error type I. The card processing is continued.

12090 : No time information (an actual start date only is not sufficient) on a card with time/cost code of blank.

Error type I. The next card is read.

12091: Two time values appear on the card, (RD and either %P or %D).

a) If remaining duration and any percentages are present, then the remaining duration is accepted.

b) If both precentages are present, only the percentage completion to date is accepted.

12092 : Three time values appear on the card, (RD, %P and %D).

Only the remaining duration is accepted. Error type R. The

The card processing is continued.

12093 : More than one cost value given on the card.

Only the actual cost to date is accepted. Error type R. The card processing is continued.

12094 : Remaining duration on the card is greater than the original duration.

The percentage completion to date on the master file is set to 0. Error type R. The card processing is continued.

12095 : Actual start date on the master file is not in accord with the updating time values.

All the time values set on the master file need to be revised.

Error type R. The card processing is continued.

Additional error messages can be found in the Diagnostics section of INP.

Audit Trail

"EXECUTE PRPT" is printed when entering this routine.

Subroutines Used

CNTRL, DTT, EWIF, CNVDT

RES (Resource Card Routine) Chart CX

Purpose

To read K-type input cards (resource cards) and to store the information in both the Description and Resource files.

Calling Sequence

CALL LINK (RES)

Input

Cards of type K, resource cards. No particular sequence is necessary. A maximum of one resource card is allowed for each work item.

Processing

After reading a K-type card and optionally printing it, the CNTRL subroutine checks the card code, the network I. D., the modification code and work item identification for validity. Subroutine ERES is called to validate that the resource codes in the K-type card exist on the Resource Description file.

Subroutine EWIF is called to verify that the work item identification has an existing work item internal number. If not, the resource information is ignored. Otherwise, the data is stored in the Resource file. Cost data is stored in the Description file.

When RES recognizes a card code different than K, control is returned to INP.

Output

Updated Resource and Description files.

Diagnostics

12096 : Invalid units code recognized.

This code is assumed to be 0. Error type R.

The card processing is continued.

12097 : Invalid positioning code.

This code is assumed to be blank. Error type R.

The card processing is continued.

12098 : Invalid resource code (not found in resource description file).

This code is not accepted.. Error type I .

The card processing is continued.

Additional error messages for RES may be found in the Diagnostics section of INP.

Audit Trail

"EXECUTE RES" is printed when entering this routine.

Subroutines Used

CNTRL, ERES, EWIF

MILE (Milestone Card Routine) Chart CY, CZ

Purpose

To read L-type input cards (milestone cards) and to store the milestone information in the Milestone file.

Calling Sequence

CALL LINK (MILE)

Input

Cards of type L, milestone cards. No particular sequence is required within the L card group. A maximum of 240 Milestone cards are acceptable.

Processing

After reading an L-type card and optionally printing it, the CNTRL subroutine checks the card code, the network I. D., the modification code and the work item identification for validity. Subroutine EWIF is called to verify that the work item identification has an existing work item internal number.

If the Milestone is new, a check is made to ensure that no more than 240 milestones will be present on the Milestone file. Subroutine AMF is then called to store this Milestone on the Milestone file.

If the Milestone is to be deleted, subroutine DMF is called to delete it from the Milestone file.

If the Milestone data is to be modified, all data punched into the L-type card is substituted for that existing on the Milestone file.

When MILE recognizes a card code different from L, control is returned to INP.

Output

An updated Milestone file.

Diagnostics

12101 = More than 240 milestones records on the master file.

Error type I. The next card is read, and so on until another card code is recognized.

12102 = Invalid positioning code

This code is assumed to be 0. Error type R. The card processing is continued.

12104 = Milestone to add already present on the master file.

Error type I. The next card is read.

12105 = Milestone to be modified or deleted not found on the master file.

Error type I. The next card is read.

Additional error messages for MILE may be found in the Diagnostics section of INP.

Audit Trail

"EXECUTE MILE" is printed when entering this routine.

Subroutines Used

CNTRL, EWIF, AMF, DMF

DLPWI (subroutine) Chart LA

Purpose

To delete all precedence relationships with an L-type lag. This subroutine is not used by the present version of 1130 PCS. It is intended to be of use in the Initialization Procedure.

MPWIF (subroutine) Chart LB, LC

Purpose

To delete all precedence relationships (doublets) from the PWI file when a work item is deleted from the network. All relationships between work items are stored on the PWI file as PWI-WI doublets.

MPWIF is used by the FWI routine.

Calling Sequence

CALL MPWIF (MB, MB1, INWI)

MB = Total number of WI records

MB1 = Total number PWI records

INWI = WI internal number

Input

The subroutine arguments.

Processing

Processing is done in two steps:

1. All records of the PWI file are searched (sector by sector) to identify PWI-WI doublets that are to be deleted. When one is found, subroutine DPWIF is called to actually perform the deletion.
2. After all the required PWI-WI doublets have been deleted, MPWIF will decrement the work item and preceding work item internal numbers to account for the WI-PWI doublet that was removed. This serves to remove the holes left in the PWI file by the delete action.

Output

None

Diagnostics

None

Audit Trail

None

Subroutines Used

DPWIF

DRF (subroutine) Chart LD

Purpose

To delete a work item record from the Resource file. DRF is used by the MILE routine.

Calling Sequence

CALL DRF(MB, INWI)

MB = Total number of WI records

INWI = WI internal number

Processing

DRF scans the Resource file sector by sector starting with the sector containing the work item record to be deleted. Deletion takes the form of overlaying the record to be deleted with its successor records.

Output

None

Diagnostics

None

Audit Trail

None

Subroutines Used

None

DDESF (subroutine) Chart LD

Purpose

To delete a work item record from the Description file. DDESF is used by the FWI routine.

Calling Sequence

CALL DDESF (MB, INWI)

MB = Total number of WI records

INWI = WI internal number

Input

The subroutine arguments.

Processing

Beginning with the sector in which the work item record is located, DDESF shifts all records subsequent to the work item record one position up toward the beginning of the Description file.

Output

None

Diagnostics

None

Audit Trail

None

Subroutines Used

None

DPWIF (subroutine) Chart LE, LF

Purpose

To delete a preceding work item from the PWI file. DPWIF is used by the FPWI routine and the MPWIF and DLPWI subroutines.

Calling Sequence

CALL DPWIF(MB, MB1, IN1, IN2, IAD, KEY, IC)

MB = Total number of WI records

MB1 = Total number of PWI records

IN1 = WI internal number

IN2= PWI internal number

IAD = Location in PWI file or record to be deleted.

Set to 0 if Directory file requires modification.

KEY = Error key: 1 - PWI not found

2 - PWI has been deleted.

IC = Form of lag.

Input

The subroutine arguments.

Processing

Deletion is accomplished by physically shifting all WI-PWI records subsequent to the one to be deleted one position toward the beginning of the PWI file.

The Directory file is then updated by subroutine MDF.

Output

Updated subroutine arguments.

Diagnostics

Refer to error 12070 in the Diagnostics section of FPWI.

Subroutines Used

MDF

DDF (Subroutine) Chart LG, LH

Purpose

To delete a work item record from the Directory file. DDF is used by the FWI routine.

Calling Sequence

CALL DDF (MB, INWI)

MB = the Total number of WI records

INWI = WI internal number

Input

The subroutine arguments

Processing

DDF deletes a work item record from the Directory file by shifting all records, subsequent to the one to be deleted, one record forward in the file. Thus, the deletion is essentially an overlay. The internal numbers for each of the work items shifted are decremented by one, as are the address pointers for the corresponding records in the WI, Description, Processing and Resource files. The second word of the first record in the Directory file is also decremented by one.

Output

None

Diagnostics

None

Audit Trail

None

Subroutines Used

None

LPWIF (subroutine) Chart LI

Purpose

To modify the value of the lag, for end-to-end or start-to-start relationships, when the originial duration of a work item has been changed. LPWIF is used by the FWI routine.

Calling Sequence

CALL LPWIF (MB1, IN, ITIME)

MB1 = Total number of PWI records

IN = WI internal number

ITIME = WI's new original duration

Input

The subroutine arguments.

Processing

The PWI file is scanned sector by sector to determine all precedence relationships for the work item. The current value of the lag is modified on the PWI file if the relationship is expresses as a percentage, and if:

1. The WI has an end-to-end relationship with the PWI, or,
2. The WI is itself a predecessor of a WI with a start-to-start relationship to that successor WI.

Output

None

Diagnostics

None

Audit Trail

None

Subroutines Used

None

APWIF (subroutine) Chart LJ, LK

Purpose

To add a preceding work item to the PWI file. APWIF is used by the PPWI and code routines.

Calling Sequence

CALL APWIF(MB, MB1, M1, IN1, IN2, LC, IC, A, KEY)

MB = Total number of WI records

MB1 = Total number of PWI records

M1 = First record number of appropriate sector

IN1 = WI internal number

IN2= PWI internal number

LC = Current value of the lag

IC = Codes as follows:

Units position - 1 Conversion factor code for time

Tens position - 1 = percentage; 0 = time units

Thousands position - Type of relationship

1 = end-to-start

2 = start-to-start

3 = end -to-end

A = Original value of the lag

KEY = 1 - Precedence network

2 - IJ network

Input

The subroutine arguments.

Processing

The addition of a preceding work item relationship (WI-PWI doublet) to the PWI file is initiated under one of the following three conditions:

1. A new work item, and its relationship(s) to existing work items, has been introduced to the network. In this case, if the internal work item number is less than some already on file, the new WI-PWI doublet is inserted into the file in internal work item number sequence and all subsequent records are shifted one position to the foot of the file. If the new WI internal number is greater than any on the file, the new doublet record is placed at the end of the file.

2. If the internal work item number is equal to one already on file, but the PWI number is unequal to any other under this WI, then this new WI-PWI doublet record is inserted in the file in work item major, PWI minor sequence. Subsequent WI-PWI records are shifted as in 1. above.

3. If both the work item and PWI numbers of the proposed new doublet match an existing WI-PWI doublet, then:

    a) If APWIF had been called by the FPWI routine, the duplication of WI-PWI numbers is allowed as long as the form of relationship (e.g., end-to-end) does not

match what is already on file. In this case, the new doublet is inserted into the file as in 2. above.

b) If APWIF had been called by the CODE routine, the proposed addition is ignored.

Following the addition of a new WI-PWI doublet to the PWI file, APWIF calls subroutine MDF to update the Directory file.

Output

None

Diagnostics

Refer to the Diagnostics sections of FPWI and CODE.

Audit Trail

None

Subroutines Used

MDF

EWIF (subroutine) Chart LL

Purpose

To allocate or search for a work item internal number given a proposed work item external identification.

Calling Sequence

CALL EWIF (MB, M, INWI, NWI, KEY)

MB = Total number of WI records

M = Record counter

INWI = Transmitted WI internal number

NWI = WI external identification from card

KEY = 1 if WI is not found

2 if WI is found

Input

The WI disk file and subroutine arguments.

Processing

1. In the FWI routine:

   a) If a WI is to be added, EWIF ensures that no such WI is already stored.

   b) When the WI identification is to be changed, EWIF checks to see that the new identification has not already been used.

2. In all other routines:

EWIF provides all other Input Processor routines with the WI internal number corresponding to the WI identification.

EWIF scans the WI disk file sector by sector. When the appropriate number is found, the search is stopped. It is used in FWI, FPWI, SCHD, RES, PRPT and MILE routines.

Output

Updated subroutine arguments.

Diagnostics

12047 = WI missing on the master file. Error type I. The next card is read.

12057 = WI already present on the master file.

Error type I. The next record is read. (see F WI routine).

Audit Trail

None

Subroutine Used

None

CNTRL (subroutine) Chart LM, LN

Purpose

To test the validity of information common to the following input cards:

Work Item card (type G)

Preceding Work Item card (Type H)

Schedule card (Type I)

Progress Reporting card (Type J)

Resource card (Type K)

Milestone card (Type L)

CNTRL is used by the WI, PWI, SCHD, PRPT, RES and MILE routines.

Calling Sequence

CALL CNTRL (ICODE, ID, MOD, IMOD, NWI, IK, KEY, IER, LEVEL)

ICODE = Card Type code (G, H, I, J, K, L.)

ID = Network code from card

MOD = Modification code

IMOD = Dependent on the routine calling CNTRL

WI and MILE = 1

PWI = 2

SCHD and RES = 3

PRPT = 4

IMOD value enables checking for allowable modification codes

NWI = WI identification (or code)

IK = Branch key if an error has been detected

KEY = Error key

1 = No error detected

2 = An end-of-file card detected

3 = An error has been detected

IER = Error code

LEVEL = 1 for Type R error

= 2 for Type I error

Input

The subroutine arguments.

Processing

CNTRL performs a validity test on the card code, the network I. D., the modification code and the work item identification.

Output

Updated subroutine arguments.

Diagnostics

Refer to the Diagnostics section of INP - specifically to errors 12041-12046.

Audit Trail

None

Subroutines Used

None

CNVDT (subroutine) Chart LO

Purpose

To convert duration, expressed in any time units acceptable by 1130 PCS, into days. CNVDT is used by the WI, FWI, FPWI and PRPT routines.

Calling Sequence

CALL CNVDT (IDUR, NDAY, ICONV, ITIME)

IDUR = The duration to be converted

NDAY = the number of work days in the week

ICONV = the conversion factor code

ITIME = the converted duration

Input

The subroutine arguments

Processing

Refer to the listing and flow chart.

Output

Updates subroutine argument ITIME.

Diagnostics

None

Audit Trail

None

Subroutines Used

None

MDF (subroutine) Chart LP

Purpose

To modify words 3 and 4 of the appropriate Directory file record which contain the WI's corresponding PWI and Milestone file addresses (pointers).

Calling Sequence

CALL MDF (MB, INWI, IAD, IK, IKEY)

MB = Total number of WI records

INWI = Transmitted WI internal number

IAD = Location in PWI or Milestone disk file

IK = 1 for addition; 2 for deletion

IKEY = 1 for PWI; 2 for Milestone

Input

Subroutine Arguments

Processing

1. When a PWI or Milestone is added using the APWIF or AMF subroutines;

   a) Words 3 and 4 of the WI record in the Directory file are updated with the new addresses. However, if an address already exists for the PWI, no change is made.

   b) Words 3 and 4 of all following records are incremented by one as are words 3 and 4 of record 1 which keeps count of the total number of PWI's and Milestones.

2. If a PWI or Milestone is deleted using the DPWIF or DMF subroutines,

   a) Words 3 and 4 of the WI record are set equal to 0. However, the PWI pointer (word 3) is not changed if there is another PWI for this WI.

   b) Words 3 and 4 of all subsequent records in the Directory file are decremented by one as are words 3 and 4 of the first record.

The Directory file is scanned sector by sector starting with the sector containing the WI record. MDF is used in APWIF, DPWIF, AMF and DMF subroutines.

Output

Updated subroutine arguments.

Diagnostics

None

Subroutines Used

None

DMF (subroutines) Chart LQ

Purpose

To delete a milestone from the Milestone file. DMF is used by the FWI and MILE routines.

Calling Sequence

CALL DMF(MB, MB2, INWI, IAD, IK, KEY)

MB = Total number of WI records

MB2 = Total number of Milestone records

INWI = WI internal number

IAD = Location in Milestone file. Set to 0 when calling subroutine MDF.

IK = 1 - Milestone deletion

2- WI deletion

KEY = 1 - Milestone not found in Milestone file

2 - Milestone has been deleted.

Input

The subroutine arguments.

Processing

DMF scans the Milestone file sector by sector starting with the sector containing the milestone to be deleted. Deletion takes the form of

overlaying the milestone record to be deleted with its successor records.

If a WI is deleted from the network by the FWI routine, DMF will also delete any corresponding milestone record.

Following the deletion, the Directory file is updated by the MDF subroutine.

Output

Updated subroutine arguments.

Diagnostics

Refer to error 12105 under the Diagnostics section of the MILE routine.

Audit Trail

None

Subroutines Used

MDF

AMF (subroutine) Chart LR, LS

Purpose

To add a Miles tone to the Milestone file. AMF is used by the MILE routine.

Calling Sequence

CALL AMF(MB, MB2, INWI, IDES, IPC, KEY)

MB = Total number of WI records

MB2 = Total number of milestone records

INWI = WI internal number

IDES = Milestone description

IPC = Milestone positioning code

KEY = Error key : 1 = no errors

2 = any error

Input

The subroutine arguments

Processing

AMF scans the Milestone file sector by sector. When an internal WI number on the file is greater than the proposed WI internal number, AMF inserts the new record and shifts all higher records one position toward the foot of the file. If the new milestone WI internal number is

greater than any in the file, AMF adds it to the end. Following the addition, subroutine MDF is called to update the Directory file.

Output

Updated subroutine argument.

Diagnostics

Refer to error 12104 under the Diagnostics section of the MILE routine.

Audit Trail

None

Subroutines Used

MDF

ERES (subroutine) Chart LT

Purpose

To test for the presence of resource codes (inputted on the K-type card) on the Resource file. ERES is used by the RES routine.

Calling Sequence

CALL ERES (IRC, NC)

IRC = Resource code from the K-type card

NC= Resource counter for each WI

Input

The subroutine arguments.

Processing

A maximum of 4 resource codes related to a work item are compared with the resource codes currently stored in the Resource file.

Output

The resource counter. If NC = 0, no match has been found.

Diagnostics

Refer to error 12098 in the Diagnostics section of RES.

Audit Trail

None

Subroutines Used

None

DPF (subroutine) Chart LU

Purpose

To delete a work item record from the Processing file. DPF is used by the FWI routine.

Calling Sequence

CALL DPF(MB, INWI)

MB = total number of WI records

INWI = WI Internal number

Input

The subroutine arguments

Processing

DPF deletes a work item record from the Processing file by shifting all records, subsequent to the one to be deleted, one record forward in the file. The internal work item numbers of all shifted records are decremented by one.

The Processing file is scanned by sector starting with the sector containing the work item.

Output

None

Diagnostics

None

Audit Trail

None

Subroutines Used

None

DWIF (subroutine) Chart LV

Purpose

To delete a work item record from the WI file. DWIF is used by the FWI routine.

Calling Sequence

CALL DWIF (MB, INWI)

MB = Total number of WI records

INWI = WI internal number

Input

The subroutine arguments.

Processing

The same process as used by DPF is used in DWIF except that the WI file is used.

Output

None

Diagnostics

None

Audit Trail

None

Subroutines Used

None

1130 PCS

DATA PREPARATION PROCEDURE

Data Preparation

Purpose

1. To translate PERT/CPM networks into precedence lists for future processing.
2. To topologically order the work items on the WI disk file (i.e., to renumber the work items such that each number assigned to a work item is greater than that assigned to any of its predecessors).
3. To check for the presence of multiple Start or End work items.
4. To detect network loops and provide information to enable the user to correct the network.

Calling Sequence

The Data Preparation module is not called as such. See the calling sequence for the four routines making up this module.

Input

The WI, PWI (if available), and Directory disk files.

Processing

This module consists of four routines:

1. The Encoder (CODE)
2. Topological Ordering (TOP)
3. Topological Error Detection (SFD)
4. Loop Detection (LOOP)

Data Preparation is called by the Control Program when a PERT/CPM network has been specified and either a Network Generation Run or Updating Run has been specified on the Processing Control Card. In the case of an Updating Run, this module is called only if at least one work item addition or deletion has been made during Input Processing.

Data Preparation is also used for Precedence Networks. In this case, the CODE routine is bypassed by the Control Program and Data Preparation begins with the SFD routine. The same run type and work item updating restrictions as for PERT/CPM networks apply to Precedence networks if Data Preparation is to be called.

To enter Data Preparation, a call for CODE is executed by the Control Program. After CODE creates a PWI file, it in turn calls SFD which establishes an intermediate "ranking file" and checks for single start and finish work items for the network. If no errors are detected, SFD calls TOP which assigns a "ranked" number to each work item. If no errors are detected in TOP, control is returned to the Control Program. If an error is detected by TOP, LOOP is called to identify the network loop and processing is discontinued.

Output

The Processing file updated with "ranked" work item numbers, and a PWI file if a PERT/CPM network has been specified.

Diagnostics

See each routine.

Audit Trail

See each routine.

Subroutines Used

See each routine.

CODE Chart DB

Purpose

To create a Preceding Work Item (PWI) disk file from PERT/CPM input data. Each work item is associated with its preceding work item(s) by finding a match between the work item's I-label and the J-labels of all the work items in the WI file. This encoder module consists of two routines, CODE and CODE 2, with an optional intervening disk sort. The CODE routine prepares a list of J-labels in sequence for use by the CODE2 routine. CODE2 is the routine that does the matching and creation of the PWI file. (Refer to the CODE2 write-up for processing details).

Calling Sequence

CALL LINK (CODE)

Input

Work Item disk file

Directory disk file

Processing

The Directory file is read to ascertain the total number of work item records that the Encoder will process. The number of PWI's is initialized to zero. After determining the number of WI file sectors to be processed, CODE reads the WI file 10 sectors at a time, and fills the J-label vector with three words of data for each work item read. The first two words are the J- label, the third word is the work item's internally assigned number.

For every 10 sectors of WI file read, 530 J-label triplets are established in core. These are then written on the work file at 105 per work file sector. 525 J-label triplets are written each cycle, the overflow in core being shifted to the head of the J-label vector prior to the next read and write cycle.

If the number of work items to be processed is less than 106, the J-label triplets are sorted internally prior to writing them on the work file. If the number of work items exceeds 105, the J-label triplets will be sorted by the disk sort routine prior to being processed by CODE2.

CODE2 is called directly by CODE if the number of work items is less than 106 and in addition, the flag IDCN (a word in COMMON) is set negative.

Output

A work file containing 105 J-label triplets per sector.

Diagnostics

None

Audit Trail

"EXECUTE CODE" is printed when entering this routine.

Subroutines Used

None

CODE 2 Chart DC

Purpose

Refer to CODE routine.

Calling Sequence

CALL LINK (CODE2)

Input

Work Item disk file

Work file containing 105 sorted J-label triplets per sector.

Processing

This routine reads the Work Item file sequentially starting with internal work item number 1. For each work item read, the I-label (of the external work item code) is passed against the set of sorted J-labels as created by the CODE routine. Because of core restrictions, a maximum of only 525 J-label triplets (one J-label block) can be contained in core at any one time. For each match (I-label/J-label), a PWI file record is created.

Every time there is a match on the 525th J-label of any but the last J-label block, a new J-label block is read to core and the search-match process continues. When a J-label is found to be greater than the I-label, the next I-label is read and, if equal to the I-label just processed, the previous J-label block is re-read and the search-match process continues.

If, during the course of reading I-labels, an I-label is read which is less than its preceeding I-label, a test is made to see if it is greater than the first J-label of the J-label block in core. If it is, the search-match process begins at the start of this J-label block. If it is not, the first J-label block (lowest sequential J-labels) is read to core and the search-match process starts again.

Each PWI record written consists of the internal work item number of the work item whose I-label is being considered, and the internal work item number found in the third word of the J-label triplet. The remainder of the PWI record remains at zero.

In addition to writing the PWI file, CODE2 also places the PWI file pointer in the appropriate Directory file record. This pointer is the logical record number of the first PWI record (of the set of PWI records) for each work item.

Output

A newly created PWI disk file.

The Directory disk file updated with PWI pointers and the number of PWI's.

Diagnostics

Error 12068. - More than 4500 PWI relationships have been detected. This is a fatal error indicating that the network is too large.

Audit Trail

None.

Subroutines Used

None.

SFD Charts DD, DE, DF

Purpose

To create a temporary "ranking disk file" and to identify the start and finish work items of the network.

Calling Sequence

CALL LINK (SFD) SFD is called by CODE if a PERT/CPM network, and by the Control Program if a Precedence network.

Input

Directory and PWI disk files.

Processing

The first record of the Directory File is read to get the number of work items and the number of preceding work items. Two vectors of length 2000 are then cleared to zero. A sector of the PWI file is read and half this record is processed as follows. For an internal work item N in the record a 1 is placed in IWI(N), for the preceding work item K a 1 is placed in IPWI(K). After half the file has been processed the sector in core is written on the work file for possible use by the LOOP routine. The second half record is then processed, a new sector from the PWI file is read in and processing continues till the entire PWI file has been considered.

A search is then made of the IWI and IPWI vectors in parallel. Any entry having an IWI and IPWI both zero is a standalone work item and is not connected to the net. If the IWI is zero but the IPWI is 1 this is a start; conversely, if the IWI is 1 but the IPWI is zero this is an end.

If both are one this is a work item within the net. After processing every WI, the Directory file is updated with the number of starting work items and a check is made for multiple finishes or no finish work item. The appropriate error message is printed.

There should only be one starting WI, but even if this number is greater it is stored in order to initiate the loop detection process in LOOP.

SFD finally checks for a match between the start and finish work items. If a match exists, the appropriate error message is printed and SFD calls the Control Program.

Output

Ranking disk file (see SFD Processing and Notes)

Diagnostics

Error 12150: identical start and finish work items for the network. The WI number is printed just after the error message "ERROR 12150 TYPE F"

Error 12151: no start WI. The error message "ERROR 12151 TYPE F" is printed.

Error 12152: more than one start WI. These WI's are printed just after the error message "ERROR 12152 TYPE F".

Error 12153: no finish WI. The error message "ERROR 12153 TYPE F" is printed.

Error 12154: multiple finish WI's. These WI's are printed just after the error message "ERROR 12154 TYPE F".

Audit Trail

"EXECUTE SFD" is printed when entering this routine.

Subroutines Used

None

Notes

Following is the record format of the "Ranking file".

Logical Record = 6 words

Sector Contains 53 logical records

Item	Format	Length	Position
WI Internal Number	I	1	1
PWI Internal Number	I	1	2
PWI Internal Number	I	1	3
No. of Successors	I	1	4
Flag	F	2	5-6

The number of successors is only put in if a loop is found. This is done in TOP3.

TOP Chart DG

Purpose

To read in parameters and data for TOP1 and preserve COMMON for later use.

Calling Sequence

CALL LINK (TOP). TOP will be called if there is one start and one finish work item.

Input

Directory file

PWI file

Processing

The directory file is read to get the number of work items and preceding work items. The starting work item number is read from File 210. The 550 words of COMMON are written out on a disk file so there will be enough room for internal processing. The PWI file which is sorted PWI within Work Item is read into a vector in core. Each WI is stored once as a positive number; all the PWIs of the WI are stored following it as negatives. When all have been stored, the total number of entries in the vector and the number of the starting work item are stored at the end of the vector. TOP1 is then called to do the actual processing.

Output

The vector in core for use of TOP1.

Diagnostics

None

Audit Trail

None (there is insufficient room in core to load the print subroutines).

Subroutines Used

None

TOP1 Chart DF

Purpose

To assign ranked numbers to the work items. A ranked work item is one whose ranked number is greater than the ranked numbers assigned to all of its predecessors.

Calling Sequence

CALL LINK (TOP1)

Input

A vector in core set up by TOP.

Processing

The vector set up by TOP is searched for the current work item number. This is initially set to the starting work item for the net. When a match occurs between the absolute value in the vector and the current WI number if the value in the vector is negative indicating a PWI, that entry is changed to zero. If the matching value in the vector is positive (a WI), all PWI for this WI must have been set to zero; otherwise, an error in the program has been detected. To do this, a search is made from this WI to determine that all are zero. Only then is the WI entry set to zero. In this phase a count is kept for the first nonzero and last nonzero element in order that the future searches may be shortened to this range rather than the original range of the vector. (Note the starting WI will only be used as a PWI). The current ranked

work item number starts at 1 for the starting work item and is incremented by one each time the processing file is written. If there are no more nonzero values in the vector, we are through and call TOP2. The second phase consists in searching for the first work item in the vector all of whose PWI are zero; this becomes our current work item number and we proceed back through phase 1. This process continues until we have exhausted the vector. If there is no current work item, we have found a network loop.

Output

Processing disk file updated with ranked WI number.

Diagnostics

None (Printed by TOP2)

Audit Trail

None

Subroutines Used

None

TOP2 Chart DG

Purpose

To print out any diagnostics found in TOP1 and restore COMMON for use by following routines.

Calling Sequence

CALL LINK (TOP2)

Input

COMMON from a work storage file.

Processing

COMMON is restored from the work file. IVEC (551) set by TOP1 is checked and any diagnostic messages are printed. Control returns to CPGM, if IVEC (551) is zero or negative. If IVEC (551) is positive this indicated a loop. The remainder of the preprocessing necessary for loop detection is initiated. A message is given on the console typewriter to the effect that the loop identification preprocessing is to start. Termination at this point can be achieved by putting Console Data Entry Switch 2 up before pressing start. File 210 is a copy of file 204. This was generated in the SFD routine. The loop detection procedure requires that the number of successors for each work item be stored in word 4 of the psuedo PWI file on 210. Each record of 210 is read and the PWI's are written off on file 250 which is also a work file. This file will then be read by TOP3 to accumulate the number of succesors for each work item.

Output

In the event of a loop File 250 is written for use by TOP3.

Diagnostics

ERROR 12250 TYPE F is printed if a machine or program error was found in TOP1.

Audit Trail

None

Subroutines Used

None

TOP3 Chart DH

Purpose

To put the number of successors into the psuedo PWI file on 210 for use by the loop identification routine.

Calling Sequence

CALL LINK (TOP3)

Input

The work file created by TOP2 and that created by SFD.

Processing

A vector corresponding to the maximum number of work items allowed is cleared to zeros. The file for the PWI created in TOP2 is read. Each time a work item number is read that entry in the vector is incremented by 1. At the end of this pass the vector contains the number of times each work item has been used as a preceding work item and thus also indicates the number of sucessors. The psuedo PWI file on 210 is then read and word 4 of each record is set to the number of successors for that work item. CPGM is called with IDCPX set so that loop detection will be called.

Output

Updated File 210

Diagnostics

None

Audit Trail

None

Subroutines Used

None

LOOP Chart DI, DJ, DK, DL, DM, DN

Purpose

To identify loops in the network by printing the work items on the loop(s).

Calling Sequence

CALL LINK (LOOP)

Input

Ranking disk file (output of TOP)

WI disk file.

Processing

The first record of the Directory file is read to ascertain the number of WI and PWI records to be processed. LOOP then requires two phases using three procedures to detect and identify the loop(s). The first phase alternately processes procedures 1 and 2 (below) and provides the list of WI's on the loop(s). The second phase uses procedures 1, 2 and 3 to determine if there is more than one loop in the network.

1. Phase 1:
   a) Procedure 1 (Search on rows) searches for a WI having no PWI's and having one or more successors (SWI's). Each

time such a work item is found, its corresponding PWI internal number in the ranking file is set to 0 by the ERK subroutine, and the number of successors is decremented. Procedure 1 ends when SWI = 0 and Procedure 2 is entered.

b) Procedure 2 (Search on columns) searches for WI's having no SWI's but having one or more PWI's. For each such WI found, the PWI of the WI is set to 0 and the number of successors of that same PWI is decremented by 1. A return is made to Procedure 1 and the process loops until WI's can no longer meet the required conditions.

c) When no more processing can be done by Procedures 1 and 2, the ERK subroutine detects all the WI's whose SWI's = 0 (i.e., WI's on the loop) , and these WI's are printed.

2. Phase 2

a) Procedure 3 searches for a WI with PWI = 0 and with SWI = 0. The PWI's are set to 0 and Procedure 1 is entered.

b) After Procedures 1 and 2 can no longer proceed, the ERK subroutine tests the ranking file and prints a diagnostic:

- if all the PWI's = 0, the message "ONE LOOP" is printed.
- if all the PWI's are not 0, the message "SEVERAL LOOPS" is printed.

Output

None

Diagnostics

None

Audit Trail

"EXECUTE LOOP" is printed when entering this routine. The messages "ONE LOOP" or "SEVERAL LOOPS" will print along with a list of WI's on the loop(s).

Subroutines Used

ERK

ERK (subroutines) Chart DO

Purpose

This subroutine is used to support the processing of the TOP and LOOP routines. In the TOP routine, it is used to set PWI's in the ranking file to 0. In LOOP, it sets PWI's to 0, decrements the SWI count, tests for SWI's $\neq$ 0 and tests for PWI's $\neq$ 0 in the ranking file.

Calling Sequence

CALL ERK(MB1, IN, KEY, 1KEY, IND, I, N, M1)

MB1	=	Total number of PWI records
IN	=	Transmitted internal number
KEY	=	Switch (1 if search starts at 1st record of 1st sector, 2 otherwise)
1KEY	=	(See Processing Section)
IND	=	Number of successors to work item IN
I	=	Record buffer number
N	=	Record number
M1	=	First record of the sector

Input

Ranking disk file.

Processing

The processing to be done is controlled by a switch (IKEY) set by

either TOP or LOOP.

IKEY = 1, ERK sets all PWI's to 0

IKEY = 2, A WI's SWI's are decremented by 1. (LOOP call only).

IKEY = 3, Test if all SWI's are equal to 0. (LOOP call only).

IKEY = 4, Test if all PWI's are equal to 0. (LOOP call only).

Refer to the flow chart for detailed processing. Abbreviations used are : WI = Work Item, PWI = Predecessor Work Item, SWI = Successor Work Item.

Output

Updated Ranking file

Diagnostics

None

Audit Trail

None

Subroutines Used

None

ELAB (subroutine) Chart DN

Purpose

To examine all J-Labels of all Work Item records of the WI file of a PERT/CPM network and to transfer the corresponding WI internal number to the CODE routine for further processing. This subroutine is currently not used in 1130 PCS.

Calling Sequence

CALL ELAB (MB, M1, N, I, ILAB, KEY)

MB =	Total number of WI records
M1 =	First record number of the sector
N =	PWI internal number
I =	Records counter (buffer)
ILAB =	WI I-label
KEY =	Switch (1 = start search at the 1st record. 2 = continue search)

Input

The WI disk file

Processing

Each record of the WI file is examined. If the J-Label of any WI is equal to the Current Label (an I-Label), then the WI internal number of this record is transferred as a PWI internal number to the CODE routine and the search continues until the end of the WI file. The

WI file is scanned sector by sector (53 records in core storage).

Output

Updated subroutine arguments.

Diagnostics

None

Audit Trail

None

Subroutines Used

None

1130 PCS

NETWORK PROCESSOR PROCEDURE

NETWORK PROCESSOR

Purpose

This module computes early and late times and floats for all work items belonging to the network being processed. All computations performed by this module take into account the various calendar or time parameters assigned to the network itself or to the specific work items. Results of the computations performed are stored on the Processing file.

Calling Sequence

The Network Processor module is not called as such. See the calling sequence for the routines making up this module.

Input

The Description, Processing, and PWI disk files.

Processing

This module consists of four routines:

1. Time Pre-Processing (ESEF)
2. Early Times Calculations (ESEFT)
3. Late Times and Float Calculations (LSLF)
4. Start and Finish Float (SLACK)

The Network Processor is called by the Control Program by executing a call for ESEF. After all early times have been calculated, by progressing forward through the network, ESEF calls the LSLF routine. Following the computation of all late times and both start and finish floats, control returns to the Control Program. The Network Processor is called whenever a Network Generation Run, an Update Run, or a Progress Reporting

Run has been specified on the Processing Control Card.

Output

The Processing file updated with Early and Late start and finish times and floats as noted above.

Diagnostics

See ESEF, LSLF, ESEFT and SLACK Routines.

Audit Trail

Refer to Component Routines.

Subroutines Used

Refer to Component Routines.

Notes

A maximum of 1 year (365.0 elapsed calendar days) of negative float is allowed for any work item in the netowrk. To implement this feature, the Project Base Date is shifted forward in time from day 1.0 to day 365.0. All computations are based on day 1.0 or, alternatively, Project Base Date less one year.

Work days are broken down into tenths of days thereby permitting start and finish times to assume mid-day values. If,however, all work item durations and lag values are expressed in whole time units, then each work item would begin on day X at time X.0 and would finish on day Y at time Y.9.

X.0 = first tenth of day number X

X.9 = tenth tenth of day number X

where X is the Xth day of the project

All times computed on the Network Processor are first calculated based on work item duration and then adjusted, according to the various calendar constraints, through the use of the appropriate calendar subroutines.

The assignment of schedule dates to either the start or end of a work item can significantly influence the Network Processor calculations. Following is a breakdown of the effect on the calculations by schedule date type.

1. Schedule Date Type I - These dates have no influence on the Network Processor calculations.
2. Schedule Date Type II - These dates are used in ESEF calculations only. A Type II Schedule Date will replace a calculated date only if it is greater than the calculated date. Succeeding calculations will be based on the date chosen by the comparison. If the schedule date is less than the calculated date, it is ignored. Actual dates for a work item always override a Type II Schedule Date.
3. Schedule Date Type III - These dates are used in both ESEF and LSLF calculations. A Type III Schedule Date always overrides a calculated date but never replaces an actual date. If a work item is completed or if the Data Date is greated than the Type III Schedule Date, the Type III Schedule Date is ignored. Usage of this type of date must be made quite judiciously since the creation of negative floats or even negative durations for work items may occur. When assigning Type III Schedule Dates, the following conditions should be considered:
    1. A split or interrupted work item may be generated if its early start plus its duration is less than the Schedule Date.
    2. A compressed work item may occur (Negative slack generated)

if its early start plus its duration is greater than the Schedule Date. Similar results would occur if both Start and Finish Type III Dates are assigned to either the same work item or several work items whose total duration is greater than the difference between the assigned start and finish schedule dates.

Both Start and Finish Floats are calculated for every work item. Normally, Start Float would equal Finish Float for a work item. However, such factors as calendar constraints and Type III Schedule Dates may force different values for these Floats. The value of the float calculated in SLACK is in work days in accordance with the corresponding work item calendar. Lag factors may be applied to the relationships between work items in Precedence networks. A 7 day week calendar is used for all lag relationships.

The Data Date, scheduled and actual dates must be converted to processing format before being inserted for a calculated date. The following conversion is used. X is the date to be converted.

$$Y = (X-PBD + 365)\ 10$$

Start and Finish Actual dates will always override any date calculated for that work item by the Network Processor.

The following abbreviations are used in the description, formulas and flow charts of the Network Processor:

SE - Early Start time

SL - Late Start time

SA - Actual Start time

SIII - Type III Schedule Start time

SII - Type II Schedule Start time

FE - Early Finish time

FL - Late Finish time

FA - Actual Finish time

FIII - Type III Schedule Finish time

FII - Type II Schedule Finish time

FS - Start Float

FF - Finish Float

t'e - Current Duration of a work item

LAG - Current Value of the Lag Factor

WI - Work Item

PWI - Preceeding Work Item

SWI - Succeeding Work Item

PBD - Project Base Date

ESEF Chart EA

Purpose

To set up the processing file and portions of the communication area of core for time and float calculations.

Calling Sequence

CALL LINK (ESEF)

Input

Processing file, directory file

Processing

The ranked work item sequence vector is cleared to zeros. All records in the processing file are then read and operated upon. Note that this read is done on a sector basis with indexing by the program rather than on a FORTRAN record basis. In each 15 word record the 11th and 12th positions are set to 32767. This will be used as a starting value for the late start and late finish calculation. The 15th word is set to zero for possible use as a flag to indicate an optimal value. The second word is used as an index for the ranked work item vector. The first word contains the internal work item number and is placed into the vector in the position specified by the second word of the record. When all records of a sector have been treated in similar fashion the sector is written back onto the processing file. This continues until the processing file has been exhausted.

Output

Updated processing file and vector in COMMON.

Diagnostics

None

Audit Trail

ENTER ESEF

Subroutines Used

None

ESEFT Chart, EB, EC, ED

Purpose

This routine calculates the early start and early finish times for all work items in the Processing File for the network. The computations take into account Actual dates, Type II Schedule dates, Type III Schedule dates, lags and calendar constraints. The Early Start and Early Finish times for each work item are placed in the Processing File.

Calling Sequence

CALL LINK (ESEFT)

Input

1. Processing File

   Maximum of 2000 15-word items (see Disk File Data section for format)

2. PWI File

   Maximum of 4500 6-word items (see Disk File Data section for format)

Processing

A vector in core is set up by ESEF which contains internal work item numbers in ranked work item sequence. The following process occurs for each WI.

1. Existing Actual and Schedule dates are converted to represent elapsed time from the Calculation Base Date (CBD=PBD-364).

2. Values for D' and D* are calculated. (see Notes below).

3. If both SA and FA are present, SE and FE are set to the values of the actual dates, the Processing File is updated and the next record is read.

4. If only FA is present, FE is set equal to FA and SE is computed (SE=FE-D') using subroutine CALFD. The message "WI COMPLETED BUT NO SA" is printed. The Processing File is updated and the next record is read.

5. If only SA is present (WI assumed to be in progress), SE is set equal to SA. If the WI had been reported to be complete, then a message "WI COMPLETED BUT NO FA" is printed. The finish date is then computed, but taking into consideration only those PWI's having end-to-end relations with the work item. The FE selected to be written on the Processing file is then: FE=minimum (FE', Data Date). The selected date is adjusted according to the WI calendar by subroutine CALF, the Processing File is updated and the next record is read.

6. If only SA is present and FIII is also present, and the WI is reported complete, then FE is set equal to FIII, the Processing File is updated and the next record is read. If, however, there is no FIII or if FIII<Data Date, then an attempt is made to calculate FE with FE' and FE* values (see Notes below). These are obtained from the WI itself (with FE'=SE+D' using the CALSD subroutine) and by scanning the PWI file to compute FE*=FE(PWI) + LAG. When all the PWI's with finish-to-finish relationships to this WI have been considered, and a final value for FE' computed, the result is stored on the Processing file and the next record is read.

(In this case, as SA is given, PWI's with start-to-start and end-to-start relationships to this WI are ignored in the computation).

7. If no actual dates have been given and the WI is reported to be complete, then the message "NO SA AND NO FA BUT WI COMPLETED" is printed. Since the WI is completed, Schedule dates for this WI are not considered. SE is calculated as explained below, and FE is directly computed with the formula FE=SE+D'. The result is stored on the Processing file and the next record is read.

8. If the WI has not yet started (no SA), then the procedure is as follows:
A test is done to check for the presence of a Type III Schedule Start Date. If there is such a date, it must be greater than the Data Date and, if so, SE=S_{III}. SE=Data Date otherwise. The Finish date is then computed by examining each PWI having an end-to-end relationship to this WI and calculating FE accordingly. Type II and III Schedule Finish Dates are considered in the computation of the finish date. The resulting FE value is adjusted with the CALF subroutine, the result is stored on the Processing file and the next record is read. If no Type III Schedule Start date is given, then the following procedure is followed:
SE' is set equal to the Data Date or to 365 if it is the first WI of the network. With the aid of the PWI pointer on the Directory file, the PWI records tied to the considered WI are read and processed sequentially. Formulas given below are used in SE* and SE' calculations for the PWI's related to the considered WI. SE is then set to the last computed SE' and a test is

made for a possible Type II Schedule Start date. The maximum of SE' and SII is taken as the final value of SE. An attempt is made to calculate FE with the CALSD subroutine (FE=SE+D') and a test is made for either Type III or II Schedule Finish date presence and validity. When the final value of FE is found, results are written on the Processing file and the next record is read.

When all records have been processed and written on the Processing file, the LSLF routine is called for Late Time.

Output

The Processing file updated with new early start and finish times.

Diagnostics

"WI COMPLETED BUT NO SA"

"WI COMPLETED BUT NO FA"

"WI COMPLETED BUT NO SA AND NO FA"

Refer to the Processing Section for the implications of these messages.

Audit Trail

"EXECUTE ESEFT" is printed when entering this routine.

Subroutines Used

CALF, CALFD, CALSD, CALS

Notes:

The following abbreviations and formulas, not indicated in the Network Processor section, are used in the ESEFT routine.

SE*--Early Start time (intermediate value)

SE'--Early Start time (maximum intermediate value)

FE*--Early Finish time (intermediate value)

FE'--Early Finish time (maximum intermediate value)

Let D' and D* be two variables where

D'=Maximum [(te'-0.1), 0.0)]

D*=te'-D'

Then, in decreasing order of priority, the value of SE selected or computed by ESEF is obtained as follows:

1. SE=SA, if SA is available
2. SE=(FA-D'), if FA is available
3. SE=Maximum of (Data Date, SII)
4. SE=Maximum of (Data Date, SE'), where SE'=Maximum of (SE*, SII)

SE* is calculated based on the various WI/PWI relationships as follows:

1. End-to-Start

   a) LAG=000.0

      SE*=FE(PWI)+D*(PWI)

   b) LAG > 000.0

      SE*=FE(PWI)+LAG+D*(PWI)

   A 7 day/week calendar is applied to this value of the lag.

2. Start-to-Start

   a) LAG ≤ PWI duration

      SE*=SE(PWI)+LAG

      The PWI calendar is applied to this value of the lag.

b) LAG > PWI duration

$SE^{*} = FE(PWI) + LAG - D'(PWI)$

A 7 day/week calendar is applied to this value of the lag.

3. Finish-to-Finish

$SE^{*} = FE(PWI) + LAG - D'$

The WI calendar is applied to the value of the lag if LAG $\leq$ WI duration.

A 7 day/week calendar is applied otherwise.

With SE known, the Early Finish time (FE) is calculated as follows:

$FE = SE + D'$

The WI calendar is used in this calculation.

LSLF Charts EF, EG

Purpose

To calculate late start and late finish times for all work items in the processing file for the network.

Calling Sequence

CALL LINK (LSLF)

Output

1. Processing file (output of ESEF)

Maximum of 2000 15 word items

(see Disk File Data section for format)

2. PWI file

Maximum of 4500 6-word items

(see Disk File Data section for format)

Processing

1. Processing is based on a vector carried in COMMON and set up by ESEF. This vector contains the internal work item numbers of all work items in the net carried in ranked work item sequence. The first entry in the vector then is the internal work item number of the start of the net and the last entry is the internal work item number for the end of the net. Since we are performing late calculations we read this vector in inverse order.

2. The first operation to be performed is to set the late start and finish equal to the early start and early finish for the last work item in

the net.

3. Starting with the last work item in the net each entry in the vector is used as follows. The processing record for the work item is read into core. The work item numbers, type lags, relationships for the PWI's are read into core. The late start and late finish times for each PWI are computed and inserted in the processing record for the PWI if the times are earlier than those previously in the record. The value of the late start and late finish in all processing records prior to entry to LSLF had been set in ESEF at the maximum one word integer of 32767. The processing for the records is as indicated in steps 4-7.

4. For any work item if both the actual start and actual finish are given the late start and late finish are set equal to the actual dates. We then go to step 8.

5. The work item is checked to see if both start and finish have type 3 schedule dates; is so, the late start and late finish are set to the type 3 schedule dates and step 8 is performed.

6. If the conditions in 4 and 5 are not fulfilled we really have the following situations possible where S indicates the subscript for the successor and P for the predecessors, LS, LF, and LAG are self-explanatory. These computations are dependent only on the type of relationship.

On a finish-to-start relationship

$LF_p = LS_s - LAG$

In a start-to-start relationship

$$LS_p = LS_s - LAG$$

In an end-to-end relationship

$$LF_p = LF_s - LAG$$

These computations are performed by the COMLS and COMLF subroutines. All lag relations are assumed to be on a 7-day=week basis. The user may if he wishes change this to any desired algorithm by changing the two subroutines. The subroutines select the correct date for insertion, i.e., actual, type III schedule, data date, or calculated date, and set a flag to indicate whether this date is earlier than that previously in the processing file. If the date is not earlier we disregard the remainder of the computation and we go to step 8. If the date is earlier we continue to 7.

7. There now are only two situations left; we compute as follows either

$$LS_s = LF_s - DUR_s$$

or

$$LF_s = LS_s + DUR_s$$

Again these computations are performed by COMLS in the 1st case, on COMLF in the second.

8. The next PWI is considered. If there are no more PWIs for this work item another work item is brought in and processing in step 3 is done for this record. Processing continues for all work item records.

Special Cases

Normally in a finish-to-start relationship an extra 1/10 of a day is subtracted. This is done so that a work item will end at the end of the work day rather than at the beginning of the next work day. If the successor work item has a zero duration the 1/10 day is not subtracted since in this case

$$LS_s = LF_p + LAG$$

If this were not done an extra tenth day of float would be introduced.

The other special case is the first work item in the net. In this case if the duration of the first work item is zero the extra tenth day is not subtracted. The special cases are handled by using the parameters for COMLS and COMLF.

SLACK Chart EL

Purpose

To compute start and finish floats for each work item in the network.

Calling Sequence

CALL LINK(SLACK)

Input

Processing File

Processing

The Processing File is read into core storage a sector at a time. The start and finish floats are computed for each work item. If an actual start or finish date is included in the record for a work item, the corresponding float is set to zero.

Output

Updated processing file

Diagnostics

None

Audit Trail

None

Subroutine Used

CALSF

COMLS Subroutine Chart EK

Purpose

To compute the late start of a work item.

Calling Sequence

CALL COMLS (JFLAG, LCOM1, LCOM2, ISDUR, IDRFL)

JFLAG = zero on exit this value of the late start is not better than that in the processing file. If JFLAG = 1 a better value was computed and inserted in the processing file for the late start.

LCOM1 = the time to be used to compute this late start. (Either a late start or late finish).

LCOM2 = the value to be used in computing this late start. (Either a duration or lag).

ISDUR = the duration of the succeeding work item. This may also be used as an indication that the tenth day correction is not desired if zero.

IDRFL = a flag to indicate the type of relationship between this work item and its successor.

0 indicates a work item duration for LCOM2

1 " " lag is being used for LCOM2

2 " " finish-to-start relation

Input

Through the argument list and COMMON.

Processing

JFLAG is set to zero on entry. There are three major possibilities for the dates. They are considered in the following sequence.

1. An actual date. An actual start date is placed into the late start date. JFLAG is set to 1.

2. A type III schedule date. The late start date is plugged with the schedule date unless the data date is later than the schedule date in which case the data date is plugged into the late start. (Type II dates are not used in late computations).

3. The date is computed in one of two ways.

(a) IDRFL is one indicating a lag is used. In this case a putative date (LC) is computed using the equation

$$LC = LCOM1 - LCOM2$$

where LCOM1 is a start or finish date depending on the relationship and LCOM2 is a lag. The value of LC is adjusted to a work day.

(b) IDRFL is zero indicating a duration is used. LCOM1 is the late finish of the work item and LCOM2 is the adjusted duration of the work item. The putative late start is then computed using CALFG.

(c) For both computed cases the putative date is compared to that in the processing file. If the putative date is earlier it replaces that on the file. If a computed date is earlier than the data date it is replaced by the data date.

Output

As indicated in processing.

Diagnostics

None

Audit Trail

None

Subroutines Used

CALF, CALFG

COMLF Subroutine Chart EK

Purpose

To compute the late finish of a work item.

Calling Sequence

CALL COMLF (JFLAG, LCOM1, LCOM2, ISDUR, IDRFL)

JFLAG zero on exit, this value of the late finish is not better than that in the processing file. If JFLAG = 1 a better value was computed and inserted in the processing file for the late finish.

LCOM1 the time to be used to compute this late finish. (Either a late start or late finish).

LCOM2 the value to be used in computing this late finish. (Either a duration or lag).

ISDUR the duration of the succeeding work item. This may also be used as an indication that the tenth day correction is not desired if zero.

IDRFL a flag to indicate the type of relationship between this work item and its successor.

0 indicates a work item duration for LCOM2

1 " " lag is being used for LCOM2

2 " " finish-to-start relation

Input

Through the argument list and COMMON.

Processing

JFLAG is set to zero on entry. There are three major possibilities for the dates. They are considered in the following sequence.

1. An actual date. An actual finish date is placed into the late finish date. JFLAG is set to 1.

2. A type III schedule date. The late finish date is plugged with the schedule date unless the data date is later than the schedule date in which case the data date is plugged into the late finish. (Type II dates are not used in late computations).

3. The date is computed in one of two ways.

(a) IDRFL is one indicating a lag is used. In this case a putative date (LC) is computed using the equation

$$LC = LCOM1 - LCOM2$$

where LCOM1 is a start or finish date depending on the relationship and LCOM2 is a lag. The value of LC is adjusted to a work day.

(b) IDRFL is zero indicating a duration is used. LCOM1 is the late finish of the work item and LCOM2 is the adjusted duration of the work item. The putative late start is then computed using CALSG.

(c) For both computed cases the putative date is compared to that in the processing file. If the putative date is earlier it replaces that on the file. If a computed date is earlier than the data date it is replaced by the data date.

Output

As indicated in processing.

Diagnostics

None

Audit Trail

None

Subroutines Used

CALF, CALSG

1130 PCS

CALENDAR SUBROUTINES

CALSG (Subroutine) Chart AK

Purpose

To compute the finish date given the start date and the duration of a work item.

This routine performs the same function as CALSD but is much faster.

Calling Sequence

Call CALSG (NCAL, NDWW, NSTAR, NDAYS, NFIN, NPBD, NSWW)

where the explanation of the arguments is that contained in CALSD.

Input

Through argument list and COMMON.

Processing

Subroutine CALS is called to adjust the start date to a work day for this work item. If the duration is zero the finish date is set to the start date and control is returned to the calling program. If the duration is positive a seven day calendar indicating the days worked in the work week for the item is set up. Monday is day 1 etc. Sunday is day 0.

The next section of code adjusts the start date to a work day. This section may seem redundant in the light of the call to CALS above but is being retained since the call was for a special case which may be eliminated in future changes.

The integral number of weeks in the duration, in terms of work week, is computed; this is then converted to days in the same number of seven-day weeks. This is our first estimate of the finish date. The

CALFG Subroutine Chart AL

Purpose

To compute the start date given the finish date and the duration of a work item.

This routine performs the same function as CALFD but is much faster.

Calling Sequence

Call CALSG (NCAL, NDWW, NFIN, NDAYS, NSTAR, NPBD, NSWW)

where the explanation of the arguments is that contained in CALFD.

Input

Through argument list and COMMON.

Processing

If the duration is zero the start date is set to the finish date and control is returned to the calling program. If the duration is positive a seven day calendar indicating the days worked in the work week for the item is set up. Monday is day 1 etc. Sunday is day 0.

The next section of code adjusts the finish date to a work day.

The integral number of weeks in the duration, in terms of work weeks, is computed; this is then converted to days in the same number of seven-days weeks. This is our first estimate for the start date. The finish is on a work day and therefore the start is on a work day. (This does not include holidays or non-workdays). There may also be a re-

sidual number of days. This is the remainder after integral weeks have been considered for the duration. A check is then made for the holiday and non-workday (if requested) lying between the start day and the first estimate at the finish. If a holiday or non-workday is in this range the residual is incremented by a day. This is done until all holidays and non-workday have been considered. Finally CALFD is called with the first estimate as the finish for CALFD and the residual as the duration. The start date from CALFD is the correct finish date. Control is returned to the calling program.

Output

Through the argument list.

Diagnostics

None.

Subroutines Used

CALFD.

TDD (subroutine) Chart AE

Purpose

Given the elapsed time in days from O1MAR64, the Time-to-Date (TTD) subroutine determines the calendar date in the form DDMMMYY.

Calling Sequence

CALL TTD (NDAYS, ID, MA, MB, IY)

NDAYS = The elapsed time in days

ID = The day of the month

MA and MB contain the three character designation for the month (Alphameric)

IY = The two-digit year number

Input

The subroutine arguments

Processing

The basic algorithm used is as follows:

Given: Integer NDAYS

Compute: Integers D, M, Y

D = Day of Month $1 \leq D \leq 31$

M = Month $1 \leq M \leq 12$

Y = Year $64 \leq Y \leq 99$

```
      Y = INT (FLOAT(NDAYS + 1)/365.25)
100   N = NDAYS + 1 - INT (FLOAT (Y) * 365.25)
      IF (N) 110, 110, 120
110   Y = Y - 1
```

start is on a work day and therefore the finish is on a work day. (This does not include holidays or non-work days). There may also be a residual number of days. This is the remainder after integral weeks have been considered for the duration. A check is then made for the holiday and non-workday (if requested) lying between the start day and the first estimate on the finish. If a holiday or non-workday is in this range the residual is incremented by a day. This is done until all holidays and non-workdays have been considered. Finally CALSD is called with the first estimate as the start for CALSD and the residual as the duration. The finish date from CALSD is the correct finish date. Control is returned to the calling program.

Output

Through the argument list.

Diagnostics

None.

Subroutines Used

CALSD.

```
              GO TO 100
120           M = INT (FLOAT (N)/30.6)
130           D = N - INT (FLOAT (M) * 30.6 + 0.5)
              IF (D) 140, 140, 150
140           M = M - 1
              GO TO 130
150           M = M +3
              IF (M - 12) 170, 170, 160
160           M = M -12
              Y = Y +1
170           Y = Y+64
```

The input NDAYS is elapsed time from 01MAR64 in days. If NDAYS is $\leq$ 0, ID, MA, MB, and IY are set equal to 0. If NDAYS is positive, the day, month and year are calculated using the conversion algorithm.

Output

Updated subroutine arguments ID, MA, MB, and IY

Diagnostics

If NDAYS $\leq$ 0, other subroutine arguments set to 0.

Audit Trail

None

Subroutines Used

None

DTT (subroutine) Chart AF

Purpose

Given a date of the form DD MMM YY, the DATE-TO-TIME routine computes the elapsed time in days from Sunday, March 1, 1964 to the given date. An error return is given in the case of invalid input or input dates prior to March 1, 1964.

Calling Sequence

CALL DTT (ID, MA, MB, IY, NDAYS)

ID = The day of the month.

MA and MB contain the three-character designation for the month (Alphameric)

IY = The 2- digit number representing the year.

NDAYS = The elapsed time in days.

Input

The subroutine arguments.

Processing

The basic algorithm used is as follows:

Given: Integers D, M, Y

D = Day of the month	$1 \leq D \leq 31$
M = Month	$1 \leq M \leq 12$
Y = Year	$64 \leq Y \leq 99$

Compute: Integer NDAYS

```
      Y = Y - 64
      M = M - 3
      IF (M) 100, 110, 110
100   M = M + 12
      Y = Y - 1
110   NDAYS = INT(FLOAT (Y) * 365.25) + INT(FLOAT (M)
      * 30.6 + 0.5) + D - 1.
```

The input is a day (ID), month (MA, MB), and year (IY). The date is checked for validity. If invalid, NDAYS is set equal to -1. If valid, the month is represented as an integer between 1 and 12. NDAYS is then computed using the conversion algorithm.

Output

An updated value for subroutine argument NDAYS.

Diagnostics

ID out of range - NDAYS equals -1.

MA & MB invalid - NDAYS equals -1.

IY out of range - NDAYS equals -1.

Input date < 01 MAR 64 - NDAYS equals -1.

Audit Trail

None

Subroutines Used

None

CALS (subroutine) Chart AG

Purpose

To modify the start date of a work item if the date falls on a holiday, rain day, or a day not in the work week.

Calling Sequence

CALL CALS(NCAL, NDWW, NSTART, NPBD, NSWW)

NCAL = The calendar code of the work item.

NDWW = The number of days in the work week for the work item.

NSTART = The proposed start date for the work item or input, or the new start date on output.

NPBD = The project base date.

NSWW = The number of the day of the week on which this work item's work week starts.

Input

The subroutine arguments.

Processing

The subroutine arguments are checked for validity. If valid, NSTART is checked to determine if it is a work day and, depending on the calendar factor NCAL, whether a holiday or special non-work day. If NSTART is an acceptable day, its value is unchanged. If not, NSTART is advanced to the first available work day by incrementing its original value day by day.

Output

A new value for argument NSTART (if necessary).

Diagnostics

Invalid value for NSTART - NSTART set equal to -3

Audit Trail

None

Subroutines Used

None

CALF (Subroutine) Chart AF

Purpose

To modify the finish date of a work item if the date falls on a holiday, rain day, or a day not in the work week.

Calling Sequence

CALL CALF (NCAL, NDWW, NFIN, NPBD, NSWW)

NCAL = The calendar code of the work item.

NDWW = The number of days in the work week for the work item.

NFIN = The proposed finish date of the work item on input, or the new finish date on output.

NPBD = The project base date.

NSWW = The number of the day of the week on which this work item's work week starts.

Input

The subroutine arguments.

Processing

The subroutine arguments are checked for validity. If valid, NFIN is checked to determine if it is a work day and, depending on the calendar factor NCAL, whether a holiday or special non-work day. If NFIN is an acceptable day, its value is unchanged. If not, NFIN is moved back to the first available work day by decrementing its original value day by day.

Output

A new value for argument NFIN (if necessary).

Diagnostics

The same as those for CALS.

Audit Trail

None

Subroutines Used

None

CALSD (subroutine) Chart AI

Purpose

To compute the finish date of a work item, given the start date and the estimated duration. The calculation takes into account, where necessary, holidays, special non-work days, the start and the length of the work week.

Calling Sequence

CALL CALSD (NCAL, NDWW, NSTAR, NDAYS, NFIN, NPBD, NSWW)

NCAL = The calendar code of the work item.

NDWW = The number of days in the work week.

NSTAR = The proposed start day of the work item.

NDAYS = The estimated duration.

NFIN = The finish day on output.

NPBD = The project base date.

NSWW = The start day of the work week.

Input

The subroutine arguments.

Processing

The subroutine arguments are checked for validity. If valid, NSTAR is set to a true (allowable) value by subroutine CALS. NFIN is set equal to NSTAR. If NDAYS is greater than zero, then NFIN is set equal to NSTAR and subroutine CALS is called to ensure, by incrementing, that NFIN is an acceptable work day. Each time, NDAYS is decremented by one and NFIN incremented by at least one depending on CALS. The process is continued

until NDAYS is reduced to zero.

Output

A new value for both NSTAR and NFIN (where necessary).

Diagnostics

Invalid value for NCAL - NFIN set equal to -1.

Invalid value for NDWW - NFIN set equal to -2.

Invalid value for NSTAR on input - NFIN set equal to -3.

Invalid value for NDAYS - NFIN set equal to -4.

Audit Trail

None

Subroutines Used

CALS

CALFD (subroutine) Chart AJ

Purpose

To compute the start date of a work item, given the finish date and estimated duration. The calculation takes into account, where necessary, holidays, special non-work days, the start and the length of the work week.

Calling Sequence

CALL CALFD (NCAL, NDWW, NFIN, NDAYS, NSTAR, NPBD, NSWW)

NCAL = The calendar code of the work item.

NDWW = The number of days in the work week.

NFIN = The proposed finish day of the work item.

NDAYS = The estimated duration.

NSTAR = The start day on output.

NPBD = The project date.

NSWW = The start day of the work week.

Input

The subroutine arguments.

Processing

The subroutine arguments are checked for validity. If valid, NFIN is set to a true (allowable) value by subroutine CALF. NSTAR is set equal to NFIN. If NDAYS is less than zero, then NSTAR is set equal to NFIN and subroutne CALF is called to ensure, by decrementing, that NSTAR is an acceptable work day. Each time, NDAYS is incremented by one and NSTAR decremented by at

least one depending on CALF. This process is continued until NDAYS equals zero.

Output

A new value for both NFIN and NSTAR (where necessary).

Diagnostics

The same as those for CALSD.

Audit Trail

None

Subroutines Used

CALF

CALSF (subroutine) Chart AM

Purpose

To compute the number of working days in a period, given the start and finish dates of that period. The calculation takes into account, where necessary, holidays, special non-work days, and the start and length of the work week.

Calling Sequence

CALL CALSF (NCAL, NDWW, NSTAR, NFIN, NDAYS, NPBD, NSWW)

NCAL = The calendar factor of the work item.

NDWW = The number of days in the work week.

NSTAR = The start date of the period.

NFIN = The finish date of the period.

NDAYS = The number of work days in the period on output.

NPBD = The project base date.

NSWW = The start day of the work week.

Input

The subroutine arguments.

Processing

The start and finish days are checked. If the start day precedes the finish day, local variables for the start and finish are set, and these are adjusted by using CALS and CALF to work days. If the finish day precedes the start day (this is not an error and can indicate negative float) the start local variable is set to the finish date, and the finish local variable to the start date. A flag INEG is set to 1 to indicate that the roles of the

local variables have been reversed. Before exit from this subroutine if the flag is 1 the variable NDAYS is set negative.

A seven-day calendar for the days in the work week for this work item is set up.

The number of calendar weeks between the start and finish dates are computed and converted to work weeks. At this time the residual days are computed. The residual days are the remainder after dividing by seven and are the days which represent a fraction of a calendar week. For computation purposes these are assumed to fall at the beginning of the period starting as the start date. The residual days are checked to see if they fall on a work day. If they do not, each weekend non-work day decrements NDAYS by ten.

A check is then made to see if appropriate holidays and/or non-work days fall in the entire range. If they do, NDAYS is again decremented for each one. When all holidays and non-work days have been checked, the flag INEG indicated above is as indicated above and control is returned to the calling program.

Output

An updated value for subroutine argument NDAYS.

Diagnostics

None

Audit Trail

None

Subroutines Used

CALS, CALF

1130 PCS

REPORT PROCESSOR PROCEDURE

Output Report Processor

Purpose

To print a variety of output listings dependent upon the requests made via the Output Request card (type F) and the information contained in the files. The reports may be detail or summary information, and may be based on time, cost, organization, or resource information parameters.

Input

All disk files

Processing

This procedure is composed of a set of routines made up of a Report Controller, and individual routines for each report type.

Refer to the individual write-ups for more detail.

Output

Any report requested on an Output Request Card.

Diagnostics, Audit Trail, and Subroutines Used

Refer to the routines making up this procedure.

Notes:

The Report Processor Procedure can be used with any type of run providing there is at least one Output Request Card (F type) in the input card deck.

Report Processing takes place at the end of any run and provides the PCS

user with a set of Reports giving Schedule, Resource or Cost information about the project.

The possible reports and the sequence in which they appear are:

1 - Summary Bar Chart

2 - Milestone

3 - Work Status Punched Cards

4 - Work Status and Progress

5 - Schedule

6 - Bar Chart

7 - Resource Assignment

8 - Resource Utilization

9 - Lump Sum Cost

10 - Monthly Cumulative Cost

11 - Precedence

12 - Calendar

13 - Master File Listing

14 - Master File Cards

In any run in which the user desires specific reports, the Output Request card (type F) must be prepared accordingly. Following is an explanation of the Output Request Card.

The Output Request card is divided into three main fields:

A) Identification field (col.1 - 5)

During the Initialization Process the card code (F) and the Network

ID must have been accepted by the System. Otherwise the Output Request card is rejected. See Initialization Procedure.

B) Listing constraint fields (col. 11-50)

They are of 5 different types:

1- Span dates SD_1, and SD_2 (col. 11-25)

These dates restrict the listings to information about WI's which have acceptable dates as follows:

WI information is printed if:

Start Date $\leq$ SD 2

and Finish Date $\geq$ SD 1 (with SD 1 $\leq$ SD2)(both conditions must be met)

Start and Finish dates are Latest times if the Report is requested in Late Start or Finish dates; they are earliest times otherwise.

The Span dates may be used to restrict the printing of the following reports:

- Work Status Cards
- Work Status and Progress Report
- Schedule Report
- Bar Chart
- Resource Assignment (SD1 only)
- Lump Sum Cost Report
- Monthly Cumulative Cost Report
- Calendar

If Span dates are punched on an Output Request card, they are valid for all Reports which accept this restriction and which have been requested.

2- Major Output Sorting key K ($I \leq K \leq 4$)

K may be used to have the Report printed in a predetermined sequence. If K=1, the WI's falling within the Span Dates are printed in the Organization Level #1 sequence (col. 38-41 of the Schedule Card); if K = 2 it is done in Organization Level #2 sequence (col. 43-46 of the Schedule card), and similarly for K = 3 and K = 4 for Organization Levels #3 and #4 respectively.

This sorting key (K) is effective for the following reports:

- Summary Bar Chart
- Work Status and Progress
- Schedule
- Lump Sum Cost
- Monthly Cost

The Sorting key K may be used alone or in conjunction with one or several of the Organization Codes specified in the Output Request card (col. 29-42).

If it is used alone (no Organization codes are specified on col. 29-42) then the Report will list all WI's falling within the Span Dates restriction, sorted in Organization code sequence for Organization Level K.

If this K code is used in conjunction with organization code(s) punched in the col. 29-42 of the Output Request card, then only the WI's which have this code(s) in Organization Level K are printed.

3- Organization Codes (col. 29-32, 34-37, 39-42).

There may be none, 1, 2 or 3 Organization codes specified on the Output Request card. They are always used in conjunction with a K code (col. 27) in order to know to which Organization Level they relate.

These codes, when specified on the Output Request card, restrict the Report to only those WI's having them.

They are taken into account for the following reports:

- Summary Bar Chart
- Work Status and Progress
- Schedule
- Bar Chart
- Lump Sum Cost
- Monthly Cost

4- Resource Group Code (col. 44-45)

This code may be used to restrict the printing of the resource reports to only that set of resources belonging to the group. A group can have up to 20 resources. Only one group is taken into account per Output Request card.

5- Resource Code (col. 47-50)

This is a more restrictive code. It may be used to print Resource Reports concerning only this resource and no others.

NOTE: If both a Resource Group and a Resource Code are punched in the Output Request card, the Resource reports will give information concerning all resources belonging to the group and to the stand-alone resource.

If there is no Resource Group nor any Resource Code, but a Resource Report is requested, then all resources will be on the report.

C) Report requests field (col. 53-80)

The Output Request card can contain up to 14 Report Requests which take two columns each.

Col. 53-54 Summary Bar Chart

55-56 Milestone

57-58 Work Status Card

59-60 Work Status and Progress

61-62 Schedule

63-64 Bar Chart

65-66 Resource Assignment

67-68 Resource Utilization

69-70 Lump Sum Cost

71 Monthly Cost

73-74 Precedence

75 Calendar

77-78 Master File Listing (cc. 77) and Card (cc. 78)

79-80 Spare

The first column of every report request, i.e., col. 53, 55, 57, 59, 61, 63, 65, 67, 69, 71, 73 and 75, allows the PCS user to obtain the Reports with timing information expressed in Calendar date format (DD MMM YY). The second column, when available, gives Reports in Project Times (Elapsed time from Project Base Date XXXX.X) by using the col. 54, 56, 58, 60, 62, 64, 66, 68, 70 and 74.

Columns 72 and 76 are not used.

In order to request a report, within the limitations given in col. 11-50, a number N ($1 \leq N \leq 8$) must be punched into the corresponding column. This number N also serves as the minor sorting key chosen for the Report.

The N number becomes a major sorting key for a report if there is no listing limitation given for it.

If the Major key K is specified on the card, then N must be different from K (e.g., if K = 1, N = 2, 3, ..., 8 and if K = 2, N = 1, 3, 4, ..., 8).

N may be different from one report to another, and for the same report if it is requested in both Calendar and Project time format.

N = blank means that the report is not requested,

= 1 " Organization Level 1 sequence,

N = 2 means Organization Level 2 sequence,

= 3 " WI identification (or J within I) sequence,

= 4 " Earliest Start sequence,

= 5 " Latest Start sequence,

= 6 " Earliest Finish sequence,

= 7 " Latest Finish sequence,

= 8 " Start Total Float sequence,

= 9 Spare

Depending upon the type of report requested, the value used for N (minor sorting key) is limited as follows:

Summary Bar Chart

N = 4, 5, 6 or 7 only

Milestone

N = 3, 4, 5, 6, 7 or 8 only

Work Status card

N = any number $\neq 0$

Work Status and Progress

N = 1, 2, 3, 4, 5, 6, 7 or 8

Schedule

N = 1, 2, 3, 4, 5, 6, 7 or 8

Bar Chart

N = 1, 2, 3, 4, 5, 6, 7 or 8

Resource Assignment

N = 1, 2, 3, 4, 5, 6 or 7 only

Resource Utilization

N = 4, 5, 6 or 7 only

Lump Sum Cost

N= 1, 2, 3, 4, 5, 6 or 7 only

Monthly Cost

N = 4 or 5 only

Precedence

N = any number $\neq$ 0

Master File Listing or card

N = any number $\neq$ 0

If N = 1, 2 or 3, it has been assumed that Early Times were considered when compared to the Span dates.

Date Printing

In the Processing file, the calculated dates SE, SL, FE, FL are in XXXX.X format (decimal point implied).

In order to use them on various reports, the following adjustments must be made by the Report routines:

a) For Reporting Calculated Dates in Calendar Form (DDMMMYY)

Compute the value V

with V = (Date/10) - 365 + PBD

This V value can then be converted into Date with the TTD subroutine.

b) For Reporting Calculated Dates in Project Day Form (XXXX.X)

Compute the value V'

with V' = Date -364.0 (Decimal point implied)

The value V' can be printed on the report

In the Processing File, the Schedule and Actual Dates are in the XXXXX format.

In order to use them on various reports, the following adjustments must be made by the Report routines:

a) For Reporting these dates in Calendar form (DDMMMYY), read the Date (XXXXX) and convert it with the TTD subroutine.

b) For Reporting these dates in Project Day Form (XXXX.X),

compute the value Z

with Z = (Date - PBD + 1) 10

Z is the date in XXXX.X format

PBD = Project Base Date

RCONT Charts FA, FB, FC, FD, FE, FF, FG

Purpose

The Report Controller (RCONT) reads and interprets the Output Requests stored in the Output Request file. RCONT analyzes each request and then prepares a work file composed of the appropriate work item internal numbers and major and minor sorting fields in accordance with the sort keys punched into the F-type card.

RCONT then sets a program switch (IGO) in COMMON indicating which report routine is to be called and relinquishes control to the PCS Sort routine. After the work file is sorted, the Sort calls the appropriate report routine based on the value of (IGO).

Calling Sequence

CALL LINK (RCONT)

Input

The Output Request file; and the Processing, Resource and Work Item files for data to include in the work file prepared by RCONT.

Processing

RCONT analyzes each report request, (cc. 53-80) of the original F card, and processes them sequentially. Depending upon which report is requested, RCONT will consider the appropriate listing constraints under the rules explained in the write-up for the Report Processor. If no span dates are

indicated for a set of report requests, RCONT assumes that SD1 = the Project Base Date, and SD2 = the computed Project Completion Date. The work file prepared by RCONT varies as follows:

1. If the Major Sorting key "K" is available (but there is no organization code in col. 29-42), then all WI's falling within the Span dates will be assigned to the Work file, and depending upon the Minor key "N" the file will have the following organization:

if N = 1 or 2	WI int'l number	1 word
(N=K)	Org. Code (level K)	2 words
	Org. Code (level 1 or 2)	2 words
	Total -	5 words

if N = 3	WI int'l number	1 word
	Org. code (level K)	2 words
	WI external Label	5 words
	Total -	8 words

if N = 4 to 8	WI int'l number	1 word
	Org. code (level K)	2 words
	Date or Time	1 word
	Total -	4 words

The Work file is set by sequentially examining the Processing, Resource and WI files as necessary according to the Minor Sort key N.

The Processing File provides the WI internal number if the Start and Finish dates are in accord with the Span dates, and provides a date or time if N = 4 to 8.

The Resource File provides the Organization codes if N = 1 or 2.

The Work Item File provides the WI external code if N = 3. It is not used otherwise.

2. If the Major Sorting key K is present together with a limiting organization code(s), the work file will have the following organization dependent upon the minor sorting key N:

if N = 1 (N = K)	WI int'l number		1 word
	Org. code(s) (level 1)		2 words
		Total	3 words
if N = 2 (N = K)	WI int'l number		1 word
	Org. codes (level 2)		2 words
		Total	3 words
if N = 3	WI int'l number		1 word
	WI external code		5 words
		Total	6 words
if N = 4 to 8	WI int'l number		1 word
	Date or Time		1 word
		Total	2 words

The work file is composed of only those work items having the specified organization code(s) in their organization level K, and to be sorted on their organization codes found in organization level N.

It should be noted that the limiting organization codes from the original F-type card are not written on the work file but are only used to limit the work file to work items meeting the constraint of a match on these specified organization codes within organization level K.

3. If there is no Major Sorting key K nor any specified organization code(s) in the Output Request card, then the Work file organization is equivalent to the one shown in 2. above. It will be composed of all the WI's belonging to the network and falling within the Span dates.

   This file is sorted in the N sequence only.

4. If a Resource Code is present on the Output Request card, then depending upon the Minor key N the file is set accordingly. It is identical to the file organization shown in 2. above.

   This Resource code is used in the same manner as are the span dates. Thus, the work file will consist of only those work items having the specified resource.

5. If a Resource Grouping code is present on the Output Request card, the same process as in 4. above is repeated for each of the resources in the group. Thus, after a work file for a resource has been prepared, sorted, and the report printed, the next resource is used to create another work file and the process is repeated.

If no resource limitations have been specified, and resource reports are requested, all resources in the system will be processed.

Output

Internal disk work files to be sorted and used as input to the appropriate report routines.

Diagnostics

The following messages will be printed if RCONT cannot process the report request in accordance with the reporting rules.

1. "NO MAJOR KEY FOR THE ORGANIZATION CODE. REPORT BYPASSED"
   This message is printed any time an organization code has been specified, but no major sorting key "K" was present.

2. "MAJOR KEY = MINOR KEY THEN MINOR KEY = EARLY START DATE."
   This message is printed any time N = K where N is equal to either 1 or 2. RCONT sets the original punched value for N equal to 4.

3. "MINOR KEY INVALID. RESOURCE UTILIZATION IGNORED."
   This report is printed only for N = 4 to 7.

4. "MINOR KEY INVALID. RESOURCE ASSIGNMENT IGNORED."
   This report is printed only for N = 1 to 7.

5. "NO MAJOR KEY FOR THE SUMMARY BAR CHART. REPORT BYPASSED."
   The summary bar chart is only available in organization level K sequence. Thus, a major sorting key must have been specified.

6. "MINOR KEY INVALID. XXX REPORT BYPASSED."

The particular report (XXX) is skipped since the given value for N was invalid.

Audit Trail

"EXECUTE RCONT" is printed when entering this routine.

Subroutines Used

TTD

SBRGR (Summary Bar Graph Routine) Charts JK, JH, JI, JJ

Purpose

The Summary Bar Graph routine prints a bar graph for one of the 4 organization levels. For each organization code within the level a line is printed indicating those weeks in which work must be performed by the organization. In addition the earliest early start, the latest late finish and the minimum float for all considered work items are printed.

Calling Sequence

This routine is a main line program called from the report controller via SORP1. The loading and transfer statement in SORP1 is CALL LINK (SBRGR).

Input

Parameters for this routine are extracted from the COMMON area used in the Report Processor Phase. Information on the work items and organization codes are obtained from the sorted work file. This information is then used to do further extraction from the Processing File.

Processing

Parameters are set up for initial entry. At this point the span dates are checked. A graph will be printed either for 7 quarters or the 16 quarters depending on the span dates. The maximum time period is 16 quarters

or 4 years. If the span exceeds this period the 4-year graph will be printed from the lower span date.

The standard heading module is printed. Following this the column headings are printed in terms of years and quarters and months. The graph is then printed with the items listed under 'Purpose'.

Output

A Summary Bar Graph on the printer

Diagnostics

ERROR 12092 TYPE I is printed if a machine or program error occured in this routine. Control is returned to RCONT to attempt further reports.

Audit Trail

'ENTER SUMMARY, BARGRAPH' is printed on entering this routine.

Subroutines Used

NEWLN, TTD, DTT, VASIT, ALPMO, FILL

WSP (Work Status and Progress Report)

Charts HA, HB, HC, HD, HE

Purpose

To print the Work Status and Progress information concerning all or some of the work items in the network. Printing is done in the sequence of the work items in the disk work file prepared by RCONT.

Calling Sequence

CALL LINK (WSP)

Input

The disk work file prepared by RCONT, and the Directory, Processing, PWI, WI, and Description files.

Processing

By calling subroutine HP, titles and headings are printed at the start of the report and every time the major sorting code changes. Sub-headings are repeated at the top of every page.

Refer to the flow chart for specific details.

Output

The Work Status and Progress Report. Refer to the User manual for a sample printout.

Diagnostics

None

Audit Trail

None

Subroutines Used

TTD, HP

SR (Schedule Report) Charts HV, HW, HX, HY, HZ

Purpose

To print schedule information about all or some of the work items in the network.

Calling Sequence

CALL LINK (SR)

Input

The sorted work file prepared by RCONT, the Processing file, and the WI file.

Processing

Full titles and headings are printed every time the major sort code "K" changes. Sub-headings are repeated on every page.

The report is terminated with general project information, specifically: Project Base Date, Project Duration, and Project Completion Date.

The processing against the work file is based on the work item internal number. A search is done as follows:

a) The WI file to get external code.

b) The Description file to get description and conversion factor codes.

c) The Processing file to get current duration, number of days and start day of the work week, calendar, and requested time information.

Dates are stored as integers. They are interpreted as 1) full days for schedule and actual dates or, 2) units in tenths of days for all computed dates. They may be expressed on output in two formats:

a) Calendar dates. The dates belonging to the 1st category can be immediately converted with the TTD subroutine and printed in day, month and year. The others must be previously converted into days and based on the calendar origin, e. g.,

Date/10 + Project base date - 365

before this conversion with TTD.

b) Project days: All dates must be previously started from the network project base date, i.e., for the first category: Date - Project base date + 1, and for the second: Date - 3640.

They can then be printed in decimal form.

Output

The Schedule Report in day number (time) or calendar date format. Refer to the User manual for sample output.

Diagnostics

None

Audit Trail

None

Subroutines Used

NEWLN, VASIT, TTD

BARGR Charts JK, JL, JM

Purpose

This routine does the initialization for the bargraph.

Calling Sequence

This routine is called in the report controller sequence via SORP1, by a CALL LINK (BARGR).

Input

Parameters from COMMON used in the report processor phase.

Processing

The symbolic FORTRAN files are assigned numeric values corresponding to those used in 1130 PCS. Dates are converted to the report format and page parameters are generated depending on whether calendar or project days are to be used. Heading dates are also generated depending on the report format. Control is then passed to BAR1 for the report printing.

Output

Through COMMON

Diagnostics

None

Audit Trail

None

Subroutines Used

NEWLN, VASIT

BAR1 (Bar Graph Routine) Charts JK, JL, JM

Purpose

The Bar graph routine is used to print bar graphs if indicated on the output request card. Graphs may be printed by any of the 4 organization levels or without regard to Org. level. Either project or calendar dates will be printed.

Calling Sequence

This routine is a mainline program called from the report controller via SORP1 and BARGR. The loading statement and transfer statement in BARGR is CALL LINK (BAR1).

Input

Parameters for this routine are extracted from the COMMON area used in the Report Processor phase. Information on the work items to be printed and the organization codes (if used) are obtained from the sorted work file. This information is then used to obtain necessary information for a work item from Processing, Description and Work Item files.

Processing

The standard heading module is printed. The appropriate column heading for this report is then generated either in calendar days or in project days. The work file is read and if no organization level was indicated a report will be printed in the sequence specified by the minor sort key. New pages will be turned when the maximum line count for

a page has been reached. Each page is followed by succeeding sections to cover the span specified by the span dates. If an organization level is specified a new page will be turned each time there is a change in organization code in the work file. Prior to the new page however the succeeding sections will be printed for the span date period. At the conclusion of the report, control returns to the Report Controller.

Output

A bar graph is generated on the printer.

Diagnostics

None

Audit Trail

None

Subroutines Used

VASIT, IFILL

LSCST (Lump Sum Cost Report) Charts JQ, JR

Purpose

To generate a cost report totaling all costs for all work items for an organization code for a given organization level.

Calling Sequence

This routine is a main line program called from the report controller via SORP1. The loading and transfer statement in SORP1 is
CALL LINK (LSCST)

Input

Parameters for this routine are obtained from COMMON for the Report Processor Phase. Information on the work items and organization codes is obtained from the sorted work file. Further information to print the report is then extracted from the Processing and Description files.

Processing

Parameters are set up for initial entry and the standard headings are printed. The column headings for the lump sum costs are printed.
A record is read from the work file. The work item and organization code are used to bring the appropriate description record containing costs into core. The costs are printed with the work item description and the external work item number. The organization code is printed with the first work item of a set. This sequence is followed until an organization change occurs. The totals are then printed for the previous

organization and processing continues with the next organization code.

Output

A lump sum cost report is generated on the printer.

Diagnostics

'ERROR 12093 TYPE I' is printed and indicates a machine or program error.

Audit Trail

ENTER LUMP SUM COST is printed on entry to this routine.

Subroutines Used

NEWLN, VASIT, ALPMO, VASIJ

MOCST (Monthly Cost Routine) Charts JD, JE

Purpose

This report prints the actual cost and estimated cost for work items for an organization level. These are printed for each month contained in the span period. Totals for each month for estimated and actual costs are accumulated and printed. At the end of the report the totals of all estimated and actual costs for the entire project are printed.

Calling Sequence

This routine is a main line program called from the report controller via SORP1; the loading and transfer statement in SORP1 is CALL LINK (MOCST).

Input

Parameters for this routine are extracted from the COMMON area used in the Report Processor Phase. Information on the work items and organization codes are obtained from the sorted work file. Further information is then extracted from the Work Item and Description File.

Processing

Parameters are set up for initial entry and the standard headings are printed. The column headings are printed and the work file is read. The estimated cost, actual cost to date and organization code for each work item in that file is printed. The month is printed for the first work item in the month.

When the month changes the totals are printed for all work items in the preceding month. After the last month's totals are printed, the totals for the project are printed.

Output

A Monthly Cummulative Cost Report is printed on the 1132 Printer.

Diagnostics

ERROR 12093, TYPE I is printed in the event of a program or machine error.

Audit Trail

'ENTER MONTHLY COST REPORT' is printed on entering this routine.

Subroutines Used

NEWLN, VASIT, ALPMO, VASIJ

CALRP (Calendar Report) Chart JF

Purpose

The calendar report routine prints a calendar for the span period. For each day in the period the project day number and the calendar date are printed. In addition if the day is a holiday or non-work day this is also indicated.

Calling Sequence

The calendar routine is called from the Report Controller. The transfer and loading statement is CALL LINK (CALRP).

Input

Parameters are passed through COMMON used in the Report Processor Phase.

Processing

Parameters are processed for initial entry, and the heading is printed. The calendar is then printed by weeks. Each week has two lines, the first consisting of the project day numbers. These are preceded by an 'H' if the day is a holiday and are followed by 'NW' if the day is a non-work day. The second line contains the calendar days corresponding to the project days.
This report may be terminated by putting console data entry switch 0 up.

Output

A calendar report on the 1132 printer.

Diagnostics

None

Subroutines Used

TTD

HP (subroutine) Chart HH, HI

Purpose

To prepare headings to be printed on the Work Status and Progress Report (WSP routine).

Calling Sequence

CALL HP (IND, NDR, MR1, MR2, NYR, NDD, MD1, MD2, NYD, NDS1, MS11, MS12, NYS1, NDS2, MS21, MS22, NYS2, IRD1, IDADA, ISP11, ISP21, IS)

IND = 1 if major sort key is present.

= 0 otherwise.

NDR = Day of month for run date.

MR1 = First 2 alphabetic characters of run date month.

MR2 = Last alphabetic character of run date month.

NYR = Run date year.

NDD = Day of month for data date.

MD1 = First 2 alphabetic characters of data date month.

MD2 = Last alphabetic character of data date month.

NYD = Data date year.

NDS1 = Day of month for lower span date.

MS11 = First 2 alphabetic characters of lower span month.

MS12 = Last alphabetic character of lower span month.

NYS1 = Lower span date year.

NDS2 = Day of month for upper span date.

MS21 = First 2 alphabetic characters of upper span month.

MS22 = Last alphabetic character of upper span month.

NYS2 = Upper span date year.

IRD1 = Run date.

IDADA = Data Data

ISP11 = Lower span date.

ISP21 = Upper span date.

IS = Sort organization (sequence).

Input

The subroutine arguments.

Processing

The sort sequence field printed as part of the WSP report heading consists of 8 print characters indicating the major and minor sort keys. 4 print characters are set aside to print the organization code if necessary. The major sort key is found in COMMON in word IORG.

Basic dates which are stored in integer form may be printed in 2 forms:

1) Calendar dates: They are immediately converted with TTD subroutine before being printed in day, month and year;
2) Project days: They must start from the network project base date (date - project base date + 1) and they are printed in a decimal form.

Output

The headings of the Work Status and Progress Report. Refer to the User

Manual for sample printout.

Diagnostics

None

Audit Trail

None

Subroutines Used

TTD

MSPRT (MILESTONE REPORT) Chart JC

Purpose

To print the milestone report in either calendar or project day format.

Calling Sequence

This routine (MSPRT) is called in the Report Processor Sequence by a CALL LINK (MSPRT).

Input

The Processing, Milestone, Directory, Internal Work Item, Description and Work files are used.

Variables in COMMON must be set in the Report Processor Configuration.

Processing

The audit trail is printed. Variable file definitions are assigned a numeric value and local variables are selected from COMMON. The milestone report heading is then printed.

The first word of the first record of the General work file is read to obtain the record number in the Milestone file. A line describing this milestone is then printed, with the milestone description, the work item to which it is related, and either the calculated or actual date. If the date is actual an A is printed. The schedule date,if any,is printed together with a 1, 2 or 3 to indicate the type of schedule date. Finally the float is printed. In each case either the early or late value is used, depending on whether the position code of the milestone indicates the start or finish of the work item as the mile-

stone. Processing continues for all milestones selected by the report controller and inserted in the General Work file. When this file is exhausted the project start date, project duration and project completion date are printed followed by 'END OF REPORT'.

Output

A milestone report in either calendar or project day format on the 1132 Printer.

Diagnostics

'ERROR 12088, TYPE I' indicates machine or program error.

Audit Trail

'ENTER MSPRT' is printed when entering this routine.

Subroutines Used

NEWLN, VASIT, ALPMO, PUT, VASIJ

IFILL (Insert characters within Graph Limits) Chart MC

Purpose

To place characters into a print line within specified bounds corresponding to dates.

Calling Sequence

This routine is called by the FORTRAN call

CALL IFILL (PRNT, PVEC, DATE, LONG, ICHAR, IWW)

PRNT is a print line

PVEC is a vector of dimension 2

PVEC (1) is the date at the left end of the chart

PVEC (2) is the date at the right end of the chart

DATE is the starting date of the print entry

LONG is the duration of the print entry

ICHAR is the character to be printed

IWW is a character which is not to be overlaid

Input

Through argument list.

Processing

It is assumed that each print position in PRNT corresponds to a date in the form XXX. in integer format. PVEC (1) and PVEC (2) are then the dates which correspond to the left and right hand limits of the line to be printed as a bar graph. The date to be used as the start of a bar is tested against PVEC (1); if the start of the bar is earlier than the start of this page the start date

is adjusted to the page start. The end date of the bar is generated by adding LONG (the duration) to the start date. When the limits of the bar have been computed the bar is filled with the character in ICHAR. If,however, IWW is nonzero and the character IWW is already present in a print position, ICHAR will not be placed in that position.

Output

Through argument list.

Diagnostics

None

Subroutines Used

None

Audit Trail

None

PRERP (Precedence Report) Chart IA, IB

Purpose

To print a precedence report in either calendar or project day format.

Calling Sequence

The Precedence Report is called in the Report Processor sequence.

Input

The Processing, Preceding Work Item, Internal Work Item, Description and Work Files are used by this routine.

Variables placed in COMMON by the Report Controller are used.

Processing

The routine is initialized, and the symbolic file designations set to their numeric values. The general work file is read to obtain the internal work item number, the external work item number and the internal preceding work item number. The heading is then printed at the start of each new page. The files corresponding to the internal work item number and the external work item are read and the values transformed into a print line. The line is printed and processing continues until there are no more records on the General Work File to be processed. The total number of work items, the total number of preceding work items are then printed and control is returned to the Report Controller.

Output

A Precedence Report on the 1132 Printer.

Audit Trail

'ENTER PRECEDENCE REPORT'

Diagnostics

None

Subroutines Used

NEWLN, VASIT, ALPMO, VASIJ

MARYL (File Dump to binary cards) Chart JT

Purpose

This routine dumps the PCS files to binary cards. Only the necessary number of sectors to describe the files for a network are dumped. The card file is dumped with the necessary control cards and is then ready to be reloaded on the disk without further addition of DUP control cards.

Calling Sequence

This program may be called by the Report Processor sequence by inclusion of an output request card or on a stand-alone basis by the following sequence

```
//XEQ MARYL        N
*FILES                          (N Star files )
                                ( cards       )
```

In the stand-alone case the run should be stopped by pressing PROGRAM STOP after all cards are punched and the message ENTER REPORT CONTROLLER appears on the printer.

Input

All PCS files except the Output Request and Work Files are used. The first 801 locations of COMMON are reserved for use by other core loads. No input parameters are used from COMMON.

Processing

The card code values for each of the files is set up together with the FORTRAN logical file numbers for the files. The first record of the directory file is read to get the number of work items, the number of preceding work items and

the number of milestones. These values are then used to compute the number of sectors to be punched out for each file for the variable fields (all but NETID, CALEN, SYRED, SYRED). Delete cards for the files to be read in are punched in reverse order of that in which they are normally stored (this is done to minimize the time in shifting files and programs in the users area). The files in binary are then punched, preceded by a STORE to Work Storage and a STORE from Work Storage to the users area with the maximum file length in sectors. This is done so that the net can be expanded to the maximum allowable size. If storage were made directly from cards to the users area the file length would only be that read in.

Output

A deck of binary cards containing information about the network on disk at the present time.

Diagnostics

None

Audit Trail

None directly, however, the message

'NPRO, LOAD BLANK CARDS, START, PROGRAM START'

indicates this routine has been loaded.

Subroutines Used

BARDZ, PCGUS

ALPMO (Unpack Alphameric characters) Chart MA
(subroutine)

Purpose

To move alphameric information in A2 format into a print area to be printed in A1 format.

Calling Sequence

CALL ALPMO (A, I1, I2, IVEC, J1, J2)

A is a vector of information in A2 format

I1 is the first word to be unpacked from A

I2 is the last word to be unpacked from A

IVEC is a print vector into which the information from A will be stored.

J1 is the first position in IVEC into which information will be unpacked.

J2 is the last position in IVEC into which information will be unpacked.

Input

Thru argument list

Processing

Each word in A is unpacked and stored in 2 words of IVEC. Normally (J2-J1 + 1) = 2*(I2-I1 + 1); however, if this condition does not hold, then transfer will terminate when either J2 or I2 is reached.

Output

IVEC contains information in A1 format

Diagnostics

None

Audit Trail

None

Subroutines Used

None

NEWLN (Insert Sort Sequence for Printing) Chart MB
(subroutine)

Purpose

To insert the proper alphabetic characters for printing to indicate the major and minor sort sequences in reports.

Calling Sequence

CALL NEWLIN (IAREA, K, N)

IAREA is a print area to be printed with 21A1 format.

IAREA must be dimensioned 21 in the calling program.

K is the major sort indicator

N is the minor sort indicator

Input

Thru the argument list

Processing

If K has values of 1, 2, 3 or 4 the following will be inserted starting at IAREA(1) 'ORG LEV (VALUEOFK),'

If K is zero this will not be inserted

If K is greater than 4 a literal X is inserted for the value of K

If K is negative the entire area is left blank.

The following are inserted depending on the value of N either following the ORG LEV K, message or at the beginning of the area if K = 0.

N = 1 or 2	ORG LEV N
3	WI
4	ES
5	LS
6	EF
7	LF
8	STFL

For any other value of N, ORG LEV X

Output

In IAREA as indicated under processing

Diagnostics

None

Audit Trail

None

Subroutines Used

None

VASIT (Date Conversion for Printing) Chart MD
(subroutine)

Purpose

To take a date in either computation or calendar format and place this date into a print line as a calendar date or a project day in whole days or in tenths of a day.

Calling Sequence

CALL VASIT (NODAY, IVEC, IL, IV, ICORD, KEY, IPBD)

NODAY is the day number in the format specified in KEY

IVEC is the print area

IL is the 1st position in the print area

IV is the last position in the print area

(there must be at least 7 positions indicated in the print area)

ICORD is the output format

0 or 2 indicate calendar day, e. g., DDMMMYY

1 indicates project day in whole days, e. g., XXX

3 indicated project day in tenths of a day, e. g., XXX. X

KEY = 1 indicates NODAY is in computation format, e. g., early starts finishes

= 2 indicates NODAY is in calendar day format, e. g., actual starts, finishes.

IPBD is the project base date.

Input

Thru argument list

Processing

The print area is cleared; if there are not at least 7 positions indicated by IL and IV, a return is made to the calling program. Transfers are made depending on the type of input indicated by KEY and the type of output specified by ICORD. The transfer is to the appropriate code for converting the dates. Control is then returned to the calling program. If (IV -IL + 1) is 7 or 8, and calendar output is specified, the day to be printed will be inserted as DDMMMMYY. If the value is 9 or greater this will be inserted as DD MMM YY.

Output

The proper date to be printed in IVEC.

Diagnostics

None

Audit Trail

None

Subroutines Used

VASIJ, PUT, EDIT

VASIJ (Convert number for printing) Chart JS
(subroutine)

Purpose

VASIJ converts a floating point number into a print area to be printed in A1 format. This is useful in that it saves multiple FORTRAN format statements.

Calling Sequence

CALL VASIJ (A, IPRNT, I1, I2, I3, I4)

A is the floating point number to be converted

IPRNT is the print line to be printed in A1 Format. It must be dimensioned at least (I2-I1 + 1.).

I1 is the first position to be used in IPRNT

I2 is the last position to be used in IPRNT

I3 is the position of the decimal point

I4 indicates whether a dollar sign is to be printed

Input

Through the argument list

Processing

The floating point number is moved into a local vector using the PUT subroutine. The print field is then set up with editing characters depending on the input. If a zero is specified for I3, no decimal point will be printed. If I4 is non-zero, a floating dollar sign will be used. I4 equal zero indicates no dollar sign is to be used. The local vector is then placed into the print

area by using the EDIT subroutine. The program will insert positive numbers less than 10^6 or negative numbers less than 10^5. The design is especially for PCS report purposes.

Output

In the print area

Diagnostics

None

Audit Trail

None

Subroutines Used

PUT, EDIT

WSPCD (Work Status and Progress Cards) Chart HF

Purpose:

To punch out Work Status and Progress cards for selected or all work items belonging to the network.

Calling Sequence

CALL LINK(WSPCD)

Input

The disk work file prepared by RCONT, and the WI, Description, Processing files.

Processing

For each work item falling within the specified span dates on the Output Request card, the following information is punched:

Network ID number	cc. 1-4
Work Item external code	cc. 5-14
Work Item Description (1st 35 chars.)	cc. 15-49
Original Duration	cc. 50-53
Estimated duration expressed in the original time units (XXX.X)	
Calendar Information	cc. 54-56
No. days in the week	
Calendar code	
Conversion facter code	
Remaining Duration (XXX.X)	cc. 57-60
Percent Complete (0.XXXX)	cc. 61-65

Actual Date code ("A" punched if start date is actual) cc. 66

Start date (either DDMMMYY or XXXX.X) cc. 67-73

Early Start or Actual date

or

Finish Date (either DDMMMYY of XXXX.X)

Late Start date

The cards are punched in work item internal number sequence. This is the sequence in which the work items were first introduced to the system.

The routine pauses and prints a message to put blank cards into the reader. After reading the records from the various files, dates are tested for validity and converted to Calendar dated by the TTD subroutine, or to project days from project base date. After all work items are processed, control is returned to RCONT.

Output

Punched Work Status and Progress Reporting Cards.

Diagnostics

None

Audit Trail

None

Subroutines Used

TTD

RESAG (Resource Assignment Report) Chart FH, FI, FJ

Purpose:

To print resource assignment information by resource requested for a specific seven week time span during the life of the network.

Calling Sequence

CALL LINK(RESAG)

Input

The disk work file prepared by RCONT, and the Resource, Description, Processing and Calendar files

Processing

This routine prints all the necessary date for the work items having the selected resource and falling within a time span of 7 weeks starting from the first Monday prior to Span Date 1 (lower span). For each work item included in the report, the following information is printed:

Work Item external code.

Work Item description.

Start Float (in tenths of days; XXXX.X)

Resource Quantity:

For every day this resource is used by the WI, the daily rate is printed. This value is taken either directly from the Resource File or is computed from the total quantity divided by the number of work days.

In order to accomodate a 7-week time span, each logical page is divided into

three parts (3 physical pages).

Since Schedule Type II and III dates are permitted, work item computed durations may be different from the estimated ones thus introducing skewed resource information. Depending upon the minor sort key N used in the Resource Assignment Report request, the following is assumed:

N = 1, 2, 3, or 4

Only the early start date of the work item is considered. Type II or III Schedule Finish dates are ignored. Then, for every day which may be considered as a work day for the WI, the corresponding daily quantity of resource is assigned.

N = 5

Only the late start date of the work item is considered. If a Type III Schedule Finish date is present, the late finish date is recomputed from the late start date and the Type III schedule finish date is ignored. For each work day, the corresponding daily quantity of resource is assigned.

N = 6

Only the early finish date of the work item is considered. If a Type II or III start schedule date is present, the early start date is recomputed from the early finish date and calendar information, and the possible Type II or III schedule start date is ignored. For each work day, the corresponding daily quantity of resource is assigned.

N = 7

Only the late finish date of the work item is considered, and the late start date is always recomputed ignoring any possible Type III schedule start date for this work item.

For each day of the report, a daily total is reported. However, this total may not equal the sum of all the daily amounts printed since it represents the total quantity of resource requested that day. For example, if 2 men per day were required to perform a specific job, and if the job starts at noon, only one man-day would be necessary that day, not 2. If this were the only work item considered for that day, the daily amount for the work item would print as 2.0, although the total for that day would print as 1.0.
Every day on which the resource is used for a work item, its daily rate is printed, although only the actual quantity required would be accumulated to the daily total.

Daily Actual Work Load $AWL_{wi} = DR_{wi} * Y$

with DR_{wi} = Daily rate for the WI

or $DR_{wi} = TU_{wi} / CD$

TU = Total Units for the WI

CD = Current Duration

Y = Percentage of the day this resource is used

The daily total printed is the accumulated sum of the individual daily actual work loads (or resource quantities) for all the work items needing that resource.

The report is restricted to those work items having their start and finish dates falling within the calculated lower span date and the 7th Sunday from it.

Output

A separate Resource Assignment Report for each resource requested in the Output Request card.

Diagnostics

None

Audit Trail

None

Subroutines Used

HPP, TTD, OP, DRP, TP

RESUT (Resource Utilization Report) Chart HK, HL, HM

Purpose

To print all necessary information concerning the forecasted utilization for a single resource. One report is printed for each resource specified on the Output Request card(s). For the complete duration of the project, this report presents the amount of resource needed for each period, and the corresponding cumulative amount from the start. This information is presented graphically and analytically.

Calling Sequence

CALL LINE(RESUT)

Input

The work file prepared by RCONT, and the Resource, Calendar and Processing files.

Processing

Each line of the report presents data by period where a period is one week, if the request is for calendar format, and a 10 day span, if the request is in project day format.

The data presented is:

Period Starting:

In Calendar format (DDMMMYY), the date is the Monday of the considered week.

In the Project day format (XXXX.X), the date is the first day of a

10 day period, and then 1, 11, 21, 31, etc... The Project Base date is day 1.

Amount for this period:

This is the quantity of resource, expressed in resource units and tenths (XXXXX.X), for the period stated (1 week or 10 project days).

Cumulative Amount:

This is the quantity of resource expressed in resource units and tenths (XXXXXXX.X), used from the start of the project up to the end of this period.

Graphic Signs:

The information concerning the quantities are shows graphically on the report according to particular scales.

Amount this period:

It is represented by an asterisk (*) placed properly on a graphical scale on the report.

Depending upon the highest computed value from any of these amounts, a scale is selected from the following set:

0 - 80	resource	units,	then	the	pitch	is 1
0 - 400	"	"	"	"	"	5
0 - 800	"	"	"	"	"	10
0 - 2000	"	"	"	"	"	25
0 - 4000	"	"	"	"	"	50
0 - 8000	"	"	"	"	"	100

Note: The "pitch" is actually a scaling factor.

Cumulative Amount:

It is shown with a "C" placed properly on a graphical scale. Depending upon the highest computed cumulative amount, a scale is selected from the following set:

0 - 80	resource units,	then the pitch is			1
0 - 400	"	"	"	"	5
0 - 2000	"	"	"	"	25
0 - 8000	"	"	"	"	100
0 - 40, 000	"	"	"	"	500
0 - 80, 000	"	"	"	"	1000
0 - 400, 000	"	"	"	"	5000
0 - 800, 000	"	"	"	"	10, 000

These scales are chosen by the routine in such a way that all variations of the quantities may be shown on the report.

If the resource amount exceeds the highest value allowed by the scales (e. g. ,8000 and 800, 000), the sign is not printed but the value itself is printed.

Amount per period $\leq$ 99999.9

Cumulative amount $\leq$ 9999999.9

The "pitch" is the value attributed to each column on the report. For example, if the chosen scale for Amount per period is 0 - 80, the pitch is 1. This means that the first column of the report on the left is used and printed with an asterisk (*) if the amount value is between 0.1 and 1.0.

If, for a Cumulative Amount calculation, the chosen scale is 0 - 400, the pitch is 5. This means that the first column of the report on the left is

used and printed with "C" if the Cumulative Amount is between 0.1 and 5.0, the second column would be printed with "C" if the cumulative amount is between 5.1 and 10.0, etc.

Scale information is printed as headings on the report, and at the top of every page.

A horizontal line is printed in front of the period which contains the Data Date, if the Data Date is greater than the Project Base date.

If there is a conflict in sign printing (asterisk, C, period) the following priority rule has been observed: 1) asterisk, 2) C, 3) period. Periods are used as scaling delineators.

The following has been assumed:

N = 4 or 5, Start date sequence.

The Start date is taken and resources are assigned every work day. Any Type II or III Finish Schedule date the WI might have is ignored.

N = 6 or 7, Finish date sequence.

If there is any Type II or III Start Schedule date the Finish date is taken and the Start date is recomputed according to the Calendar restriction. Any Type II or III Start Schedule date the WI might have is ignored in this computation. Otherwise, the start date is taken directly.

Resources are then assigned every work day.

Output

The printed Resource Utilization Report

Diagnostics

None

Audit Trail

None

Subroutines Used

RCI, RAC, TTD, OPP, HPP

Examples of Resource Assignment and Utilization Reports:

Resource Assignment and Resource Utilization reports are requested for a particular resource R of a network N. They are requested in early start sequence and in calendar date format. The Report Controller has created the work file composed of the following work items sorted in the appropriate sequence:

WI	ES	Computed Finish	Daily Rate	Days/ Week	Type III Schedule Dates Start	Finish	
A	1.0	2.9	2.0	6	1.0		
C	1.4	5.5	3.0	6		5.9	Proj Base Date
D	2.0	10.8	2.7	5			04JUL66
B	4.2	4.7	1.0	6			
F	4.8	13.9	2.0	6			
I	5.1	6.9	3.0	6			
J	6.0	8.9	1.5	7			
M	8.0	9.9	2.0	6			

Then the Resource Assignment Report is as follows:

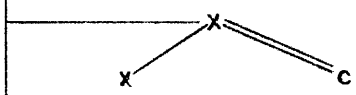
WI	M 1	T 2	W 3	T 4	F 5	S 6	S 7	M 8	T 9	W 10	T 11	F 12	S 13	S 14	Day N
A	2.0	2.0													
C	3.0	3.0	3.0	3.0	3.0										
D		2.7	2.7	2.7	2.7			2.7	2.7	2.7					
B				1.0											
F				2.0	2.0	2.0		2.0	2.0	2.0	2.0	2.0	2.0		
I					3.0	3.0									
J						1.5	1.5	1.5							
M								2.0	2.0						
Tot	3.8	7.7	5.7	6.7	9.2	6.5	1.5	8.2	6.2	4.7	2.0	2.0	2.0		

Note: On Monday, day #1, it is stated that the A and C WI's require 2.0 and 3.0 resource units respectively. However, the total for the day is 3.8 resource units. The implication is that during day #1, 2.0 plus 3.0 units of resource R will sometimes be required; however,

the work load for this resource for day #1 is:

$$DT = \sum AWL = (2.0 \times 1.0) + (3.0 \times 0.6) = 3.8 \text{ units of resource R}$$

The Resource Utilization Report would be:

Period Starting	Period Amount	Requirements	Cumulative Amount	Period Starting
	Scale	10 20 30 40 50 60 70 80	Pitch 1	
Cumulative	Scale	10 20 30 40 50 60 70 80	Pitch 1	
04JUL66	41.1		41.1	04JUL66
11JUL66	25.1		66.2	11JUL66

The amount for the first period (PA_1) is equal to 41.1. This is the cumulative amount for the complete week:

$$PA_1 = \sum_{i=1}^{7} DT = 3.8+7.7+5.7+6.7+9.2+6.5+1.5 = 41.1$$

The cumulative amount of resource units CA up to the nth period is:

$$CA = \sum_{j=1}^{n} PA_j$$

BARDZ (Punch a binary card) (subroutine) Chart JV

Purpose

To punch cards from a FORTRAN program if the cards contain binary information. BARDZ is used by routine MARYL.

Calling Sequence

CALL BARDZ

Input

The data to be punched must be stored in bits 0-11 of the 80 words starting in FORTRAN COMMON work 802.

Processing

FORTRAN common work 802 is the FORTRAN punch area start. When the data is punched the information in Common word 802 will be punched into column 80, that in FORTRAN into 801 into col. 79, etc. The buffer area must be set up with this in mind. The index registers are saved and restored before returning. Card feed and punching is handled by this routine. One 80 column card will be punched. This is an assembly language routine.

Output

An 80 column card punched on 1442 card read punch.

Diagnostics

None

Audit Trail

None

Subroutines Used

None

BARNY (Convert numeric data for punching) Chart JW
(subroutine)

Purpose

To convert a positive integer less than 1000 into 3 words for punching. BARNY is used by routine MARYL.

Calling Sequence

CALL BARNY (INO, IPARA, IND)

INO = the integer to be converted

IPARA= the punch area

IND = the right-most card column into which the integer is to be placed.

Processing

A test is made to see that the number is positive or zero. If it is negative it is made positive. If the number is greater than 999 a division is made followed by a multiplication. In effect this retains the three low order digits. This number is converted and stored in the proper position. Note that this routine requires the card column (IND) to be the punch area. to convert it to a FORTRAN index.

Output

Via argument list

Diagnostics

None

Audit Trail

None

Subroutines Used

None

PCGUS (Convert Binary Data for Punching) Chart JW
(subroutine)

Purpose

To convert 54 words in binary format (16 bits per word) to 72 words in punch format (only using bits 0-11). PCGUS is used by routine MARYL.

Calling Sequence

CALL PCGUS

Input

The 54 words to be converted must be stored starting in FORTRAN COMMON. word 883.

Processing

Assembly language is used to shift and store the words from the input area to the output area.

For example, the first 3 hexadecimal digits of Input word 1 are stored in output word 1, the last hexadecimal digit of this word is zero. The last hexadecimal digit of input word 1 and the first two hexadecimal digits of Input word 2 are stored in the 3 high-order hexadecimal digits of output word two and so on.

Output

The output is stored starting at FORTRAN COMMON word 810.

This word is the equivalent of card column 72 and in the calling program MARYL is IPSTO(9).

All index registers are saved and restored.

Diagnostics

None

Audit Trail

None

Subroutines Used

None

NZONE (Subroutine) Chart: SeeProcessing

Purpose

Interrogates the zone and returns with a code as to what the zone is. This subroutine is not directly called by the PCS routines but is required for EDIT and PUT.

Calling Sequence

CALL NZONE (JCARD, J, NEWZ, NOLDZ)

JCARD	A vector in A1 format
J	the word in JCARD to be interrogated
NEWZ	the new zone to be inserted in JCARD(J)
NOLDZ	the previous value of the zone in JCARD(J)

Input

Thru the argument list

Processing

Refer to "FORCOM-1130 IBM 1130 FORTRAN COMMERCIAL SUBROUTINES" 3.0.001.

Output

Thru argument list.

Diagnostics

See Processing

Subroutines Used

None

EDIT (Subroutine) Chart: See Processing Section

Purpose:

To edit data in A1 format for printing. Used by VASIT and VASIJ.

Calling Sequence

CALL EDIT (JCARD, J, JLAST, KCARD, K, KLAST)

JCARD	A vector of information is A1 format to be edited.
J	The subscript of the first word in JCARD to be edited.
JLAST	The subscript of the last word in JCARD to be edited.
KCARD	A vector of edit characters.
K	The subscript of the first word in KCARD to be used for editing.
KLAST	The subscript of the last word in KCARD to be used for editing.

Input

Thru the argument list

Processing

Refer to "FORCOM-1130 IBM 1130 FORTRAN COMMERCIAL SUBROUTINES" 3.0.001

Output

The edited information in A1 format in KCARD.

Diagnostics

See Processing.

Subroutines Used

NZONE, FILL

FILL (Subroutine) Chart: See Processing

Purpose

To place an EBCDIC character into a vector. Used by SBRGR, PUT and EDIT.

Calling Sequence

CALL FILL (JCARD, J, JLAST, ICHAR)

JCARD	is the vector to be used
J	is the lower subscript in JCARD
JLAST	is the upper subscript in JCARD
ICHAR	is the character to be inserted

Input

Thru the argument list.

Processing

Refer to "FORCOM-1130 IBM 1130 FORTRAN COMMERCIAL SUBROUTINES" 3.0.001

Output

Thru the argument list.

Diagnostics

See Processing

Subroutines Used

None

PUT (Subroutine) Chart: See Processing Section

Purpose

To convert a floating point variable to an EBCDIC integer number. This is used in the PCS reports module to set up print lines in Subroutines VASIT and VASIJ.

Calling Sequence

CALL PUT (JCARD, J, JLAST, VAR, ADJST, SHIFT)

JCARD An array name into which information will be placed in A1 format.

J The subscript of the first position in JCARD to be filled.

JLAST The subscript of the last position in JCARD to be filled.

VAR The floating point variable to be interpreted

ADJST A half adjustment factor

SHIFT The number of digits to the left of the decimal to be used

Input

Thru the argument list.

Processing

Refer to "FORCOM-1130 IBM 1130 FORTRAN COMMERCIAL SUBROUTINES" 3.0.001

Output

The variable VAR in EBCDIC in JLAST in A1 format.

Diagnostics

See Processing

Subroutines Used

NZONE, FILL

FILPR (Print PCS Files) Chart JG

Purpose

FILPR prints all current information on the PCS disk file.

Calling Sequence

This routine may be called either by a CALL LINK (FILPR) in the normal report sequence or on a standard basis by a // XEQ FILPR followed by the appropriate * FILES cards.

Input

The PCS disk files.

Processing

The first five files are printed in all cases. Remaining files are printed if desired. At any point in printing the remaining files the Console Entry Switches can be used to terminate printing that file. If Console Entry switch I (where I=FORTRAN File Number -200) is up the data in the file will not be printed. The headings will, however.

Output

A report on the 1132 Printer.

Diagnostics

None

Audit Trail

None

Subroutines Used

None

SORP1 (PCS Sort Entry and Return Routine) Chart ME

Purpose

To enable the PCS sorting routines to be called from different modules and return control to the proper program module.

Calling Sequence

CALL LINK (SORP1)

Input

Data to be sorted is located in Working Storage. Program parameters are located in COMMON.

Processing

SORP1 serves to control the program flow to and from the sort and also to define the work storage file as File 300.

Upon entry to this routine the first of the sort subroutines SORTA is called. On return from SORTA a computed GO TO based on the value of IGO calls the next program in the sequence. Statement numbers in SORP1 have a 1-to-1 relation with the value of IGO. Special cases are if IGO = 13. A further test is made on the value of MM for routing. If IGO = 14 this is presently a return to RCONT. This allows for the addition of new reports. The SORT package assumes that the necessary parameters are stored starting in COMMON (701). The values are as follows:

COMMON (701) = 1st key (i.e., the 16 bit word number in the record)

⋮

COMMON (709) = 9th key

COMMON (710) = Number of keys

COMMON (711) = No. of records in all sectors but last

COMMON (712) = No. of records in last sector

COMMON (713) = Length of record in 16 bit words

COMMON (714) = The sector number of the 1st record in file 300 to be sorted.

COMMON (715) = The sector number of the last record in file 300 to be sorted.

COMMON (716) = The FORTRAN logical file to be sorted. In PCS is is always 300.

Output

A sorted file in the same file position as the original.

Diagnostics

None

Audit Trail

None

Subroutines Used

SORTA

sector

SORTA (Sort a ~~record~~ internally) Chart MF, MG
(subroutine)

Purpose

To check parameters for the PCS sort and to sort each record internally.

Calling Sequence

CALL SORTA

Input

Work storage file and COMMON defined by SORP1.

Processing

SORTA checks for validity of parameters specifying input to the SORT. If parameters are incorrect appropriate diagnostics are printed. Each sector is read in and sorted internally in ascending algebraic sequence in accord with the keys requested. The last 5 words of the record are used as follows:

Word:; 316 Record index in file

317 Ø

318 Record index in file + 1

319 Record length in 16 bit words

320 Number of records in this sector

The sector number is written in a string table in columns 7 and 8 of that part of COMMON used by this program. When all sectors have been sorted, SORTC is called.

Output

Sorted sectors on the Work Storage file.

Diagnostics

ERROR 12201 TYPE F	Record count negative, machine or program error
ERROR 12202 TYPE F	More than 642 sectors to be sorted
ERROR 12203 TYPE F	Element length is zero or negative
ERROR 12204 TYPE F	Number of sort keys zero or negative
ERROR 12205 TYPE F	Number of sectors to be sorted zero or negative
ERROR 12206 TYPE F	Negative number of records in record
ERROR 12207 TYPE F	A key is zero or greater in value than the record length.
ERROR 12208 TYPE F	Machine or program error.

Subroutines Used

SORTC

SORTC (Merge sorted sectors) Chart MH, MI
(subroutine)

Purpose

To merge sorted sectors produced by SORTA.

Calling sequence

CALL SORTC

Input

Working storage from disk.

Processing

Sector Records produced by SORTA are read in and merged in accordance with the Keys. On the first pass the string table is read from columns 7 and 8 of X. On succeeding passes the string table is stored in column 5.

Output

Merged strings on the working storage file written in scattered form.

Diagnostics

ERROR 12209 TYPE F	Machine or program error in SORTC
ERROR 12210 TYPE F	Records considered in SORTA do not agree with the number of input program error.

Subroutines Used

SORTD, CRITE, CREAD

SORTD (Resequence scattered strings) Chart MJ
(subroutine)

Purpose

To reorder the strings written by SORTC in ascending sequence.

Calling Sequence

CALL SORTD (X, NFILE, INREC, IFREC, NOERR)

X is the starting address of the COMMON area used by the Sort.

NFILE is the FORTRAN logical file number.

INREC the number of sectors to be sorted.

IFREC the first sector number in working storage.

NOERR the error indicator.

Input

The working storage file.

Processing

If the string table in column 7 or X is ordered, control is returned to SORTC. If not, then the sectors are rearranged until they are in sequence and control is returned to SORTC.

Diagnostics

ERROR 12211 TYPE F Machine or program error in SORTD.

Subroutines Used

None

CRITE (Write a sector onto disk) Chart MK
(subroutine)

Purpose

To write a sector in a string from the output buffer into the next empty sector.

Calling Sequence

CALL CRITE (IOUT, INUM)

IOUT = Column number of the output buffer

INUM= 1 if this is in a string

= 0 if at the end of a string

Input

Through argument list and COMMON.

Processing

The next free sector number is selected, the number of free sectors are reduced by one and the remaining free sector numbers are pushed up by one. The 320 words in the output buffer are then written and control is returned to the calling program.

Output

The sector in the output buffer is written into work storage file.

Diagnostics

None

Subroutines Used

None

CREAD (Read a sector from disk) Chart MK
(subroutine)

Purpose

To read a sector from disk into an input buffer and record the sector number for writing in a push-up list.

Calling Sequence

CALL CREAD(IIN, INIT, I1, I2, N1)

IIN is the input buffer column number in X

INIT is 0 if this is the first time we enter to read

1 if it is not the first time

I1 is the sector to be read (Input)

I2 is the next sector to be read (Output)

N1 is the number of records read into the input buffer (Output)

Input

Argument list, COMMON and work storage file.

Processing

The sector indicated in I1 is read into the buffer specified by IIN. The sector number is then inserted as the first element of the free sector table if INIT = 0 or as the last element if INIT = 1. The free sector counter is incremented. The next sector address in this string is placed in I2. If there are no more sectors in the string this variable is set to zero. Control is then returned to SORTC.

Output

Thru argument list and in buffer area.

Diagnostics

None

Subroutines Used

None

1130 PCS
DISK FILE DATA

Disk Utilization

IBM 1130 PCS requires that a major portion of the disk be available for storage of the 1130 Disk Monitor System, the 1130 PCS programs, and the 1130 PCS required data files. The Permanent data files are detailed in this appendix.

Item	Approximate Disk Sectors Required
1130 Disk Monitor System (Complete)	375
1130 PCS Programs	276
Initialization Files	
Network ID File	1
Calendar File	2
Resource Description File	3
Resource Grouping File	2
Output Request File	2
Directory File	26
Processing File	96
PWI File	85
WI File	38
Description File	251
Resource File	101
Milestone File	24
Required Maximum Work File	251
Total	1523

Initialization Files

Network ID File (NETID)

DEFINE FILE 1(1, 27, U, IAD1)

Item	Format	Length	Position
Network ID	I	1	1
Network Description	A2	22	2-23
Data Date	I	1	24
Project Base Date	I	1	25
Run Date	I	1	26
Run Sequence Code	A1	1	27

Calendar File (CALEN)

DEFINE FILE 2(2, 241, U, IAD2)

Item	Format	Length	Position
Network ID	I	1	1
Number of Holidays & non-work days	I	1	2
Holidays & Non-wk. days	I	480	3-241 1-241

Resource Description File (SYRED)

DEFINE FILE 3(3, 315, U, IAD3)

Item	Format	Length	Position
Number of Resources	I	1	1
Resource Code	I	1	2
Resource Description	A2	8	3-10

In the Resource Description file, 100 resource codes and descriptions are stored sequentially in 9-word couplets, beginning at word 2 of sector 1.

Sector 1 = No. of Resources + 34 Resource couplets

Sector 2 = 35 Resource couplets

Sector 3 = 31 Resource couplets

Resource Grouping File (SYREG)

DEFINE FILE 4(2, 300, U, IAD4)

A maximum of 19 resource grouping codes, each containing a maximum of 20 resource codes, can be stored in the Resource Grouping File. Each resource grouping code requires 30 words in the file. The first 30 words are reserved to store the number of resource grouping codes stored. Each resource grouping code and related information is stored as follows:

Item	Format	Length
Resource Grouping Code	I	1
Description	A2	8
No. of resource codes	I	1
Up to 20 resource codes	I	20

Output Request File (SYOUR)

DEFINE FILE 5(24, 18, U, IAD5)

The information from each of a maximum of 16 output requests in any one run are grouped in 24 word segments. The first 24 words of this file are reserved for the number of output requests contained in the file. Each output request is stored as follows:

Item	Format	Length
Span Date (lower)	I	1
Span Date (upper)	I	1
Major O/P Sorting Key	I	1
Up to 3 Organization codes	A2	6
Resource code	I	1
Resource Grouping Code	I	1
Up to 14 Output Requests	I	7 (packed-2 requests/word)

Directory File (DIREC)

DEFINE FILE 202(2080, 4, U, I202)

Logical record = 4 words

Sector contains 80 logical records

26 sectors required

1st record

Item	Format	Length	Position
Reserved	I	1	1
Total # of WI records	I	1	2
Total # of PWI records	I	1	3
Total # of Milestone rcds.	I	1	4

2nd record

Item	Format	Length	Position
Total # of Work File records	I	1	1
Reserved	I	1	2
Reserved	I	1	3
Reserved	I	1	4

Records 3-10 are reserved

Record 11 through 2080

Item	Format	Length	Position
Internal WI #	I	1	1

Pointers:

Processing, WI, Description, & Resource Files	I	1	2
PWI File	I	1	3
Milestone File	I	1	4

Processing File (PROCS)

DEFINE FILE 203(2016, 15, U, I203)

Logical record = 15 words

Sector contains 21 logical records

96 sectors required

Item	Format	Length	Position	Comments
Internal WI #	I	1	1	
Ranked WI #	I	1	2	
Current Duration	I	1	3	
Flags: Start of Work week	I	1	4	units = 1-7
days in work week				tens = 1-7
type of sched. start date				hundreds = 1-3
type of sched. complete date				thousands = 1-3
calendar to use				ten thousands = 0-2
Scheduled Start date	I	1	5	
Scheduled Complete date	I	1	6	
Actual Start date	I	1	7	
Actual Complete date	I	1	8	
Early Start date	I	1	9	
Early Complete date	I	1	10	
Late Start time	I	1	11	
Late Complete time	I	1	12	
Start float	I	1	13	
Complete float	I	1	14	
Reserved	I	1	15	

PWI File (PWIWI)

DEFINE FILE 204 (4505, 6, U, I204)

Logical Record = 6 words

Sector contains 53 logical records

85 sectors required

Item	Format	Length	Position	Comments
Internal WI #	I	1	1	
Internal PWI #	I	1	2	
Current Lag	I	1	3	
Codes:	I	1	4	
Time Unit				units = 0-6
Type of Time Unit				tens = 0-1
Type of Relationship				thousands = 1-4
Initial Lag	F	2	5-6	

WI File (WIWIW)

DEFINE FILE 205 (2014, 6, U, I205)

Logical Record = 6 words

Sector contains 53 logical records

38 sectors required

Item	Format	Length	Position
Internal WI #	I	1	1
External WI code	A	5	2-6

Note: If a PERT/CPM network, the predecessor event node is stored in words 2-3, and the successor event node is stored in words 5-6.

Description File (DESCR)

DEFINE FILE 206 (2008, 39, U, I206)

Logical Record = 39 words

Sector contains 8 logical records

251 sectors required

Item	Format	Length	Position	Comments
WI Description	A	22	1-22	
Original Duration	I	1	23	
Conversion Factor codes:	I	1	24	
Original Duration				units = 0-6
Remaining Duration				thousands = 0-6
Remaining Duration	I	1	25	
Percent Complete this period	F	2	26-27	
Percent Complete to date	F	2	28-29	
Actual Cost this period	F	2	30-31	
Actual Cost to date	F	2	32-33	
Total Estimated Cost	F	2	34-35	
Cost Position Code	A	1	36	
Reserved for growth	I	3	37-39	

Resource File (RESOR)

DEFINE FILE 207 (2020, 16, U, I207)

Logical Record = 16 words

Sector contains 20 logical records

101 sectors required

Item	Format	Length	Position	Comments
Organization Codes (4)	A2	8	1-8	
Resource code 1	I	1	9	
Resource units 1	I	1	10	see note
Resource code 2	I	1	11	
Resource units 2	I	1	12	see note
Resource code 3	I	1	13	
Resource units 3	I	1	14	see note
Resource code 4	I	1	15	
Resource units 4	I	1	16	see note

Note: The spreading code is stored in the ten-thousands position of the resource units word. Its value is either 0 or 1.

Milestone File (MILES)

DEFINE FILE 208 (240, 31, U, I208)

Logical record = 31 words

Sector contains 10 logical records

24 sectors required

Item	Format	Length	Position
Internal WI #	I	1	1
Milestone Description	A	22	2-23
Positioning Code	I	1	24
Reserved for growth	I	7	25-31