STAR Release 1
Product Software
Functional Specification

Number: 596P81779
Version: 5.3
Date: April 15, 1981

RELEASE STATUS: Draft/RELEASED/Issued

THIS DOCUMENT IS XEROX PRIVATE

Approved by:

Charles Irby
Chairman, User Interface Board

Tim Townsend
Manager, Star Development
Systems Development Department

XEROX

OFFICE PRODUCTS DIVISION
Palo Alto El Segundo Dallas

TABLE OF CONTENTS

DOCUMENT CREATION

DOCUMENT MANAGEMENT

APPENDICES

3. HARDWARE OVERVIEW

This section provides a brief *functional* overview of the Star hardware components that make up a user workstation. This overview provides a basis for reading the *Functional Specification*. Readers not familiar with the Star baseline hardware should read this section as an aid to understanding the functions, command invocation, responses and information display.

For a complete description of the Star hardware configuration see *8000 Series Workstation Configuration Specification,* Version 1.0, Systems Development Department, November, 1980, and *Engineering Response to OIS Product Goals*, Version 2.0, Systems Development Department, May 1979.

STAR WORKSTATION

The *workstation* consists of the following mandatory components:

an 8000 (Dandelion) processor

a CRT for display of documents, menus, and system responses;

a keyboard for text input and command invocation, and a tone generator to alert the user that an event has occurred that may require his attention;

a pointing device to select displayed objects for processing and to invoke commands;

either a small or medium capacity non-removable, rigid disk; and

a floppy disk drive.

A workstation optionally includes:

a character printer;

an Ethernet transceiver and controller, used to connect workstations to an Ethernet network; and

a communication port, including an RS232-C/DTE interface.

A typical configuration of workstations, their attached devices, and associated Ethernet-connected servers is illustrated in figure HOV-1.

CRT Display

The basic display for Star is the Large Format display (LF). Its usable screen area measures 10.0" in the vertical dimension by 12.8" in the horizontal. With only minor scaling, this display can render two full 8.5" by 11" pages (with one inch margins).

The LF displays bit-map images from the refresh buffer. On the screen, the refresh buffer is displayed as 809 lines of 1024 dots, called *pixels* ("picture elements"). Each pixel may be individually set to white or black by altering the contents of the refresh buffer. On the surface of the display, 72 pixels represent one inch of image, both horizontally and vertically. Thus, an 11" tall image will occupy 792 vertical pixels, and will be reduced in size by 11% on the screen surface. In the subsequent sections, specific screen coordinates will be represented assuming the origin is in the upper-left corner. Thus, this corner is represented by (0, 0), and the lower-right corner is represented by (1023, 808).

Keyboard

The Star keyboard is shown in Figure HOV-2. The layout is subject to the addition/deletion of functions and to modifications for human factors and multi-national considerations for subsequent releases.

The keyboard is the primary input device. It serves four major functions.

It is used to enter *text characters*, described in *Text Editing*.

It is used to enter *formatting characters*. Examples of such characters are New Paragraph, Discretionary Hyphen, Page Format, and Tab. They are described in *Text Editing* and *Formatting and Layout.*

It is used to change character properties (such as bold-face, italics, underlined) providing a shorthand alternative to the Character Property Sheet described in *Formatting and Layout*. The Defaults key is equivalent to the Defaults command on the Character Property Sheet.

It is used to insert *special objects* into a document, such as pictures, tables or footnotes. These objects are described in general in *Keyboard Sets* and in detail in the specific section for each object.

It is used to invoke commands. Commands are invoked with function keys such as MOVE, UNDO, BACKSPACE, or PROPERTIES. Commands are described throughout this document.

The keyboard is not rigidly encoded with a specific character set. Thus, the software can interpret each key or combination of keys to provide greater flexibility, handle multiple fonts, and address multi-national considerations. The text keys can be mapped into different fonts, allowing the characters to take on different sizes and shapes. The text keys may also be mapped into different alphabets. This allows Greek characters or math symbols to be entered. Switching fonts and alphabets is described in *Keyboard Sets* and *Formatting and Layout.*

Typing Ahead - The system retains all keystrokes, even if it is still processing the previous keystroke. When the user has typed ahead of the system, responses to keystrokes will lag the actual keystroke in time.

Repeating Keys - All the printing keys on the keyboard repeat (generate additional instances of the "character") if held down for a longer period than in normal typing. Repetition begins after a key has been held down .25 seconds (will be adjusted empirically). The character is generated at 10 cps. In addition to the printing keys, the following "function" keys repeat: TAB, NEXT, BACKSPACE, BACKWORD, NEW PARAGRAPH, and NEW LINE.

Tone Generator

The tone generator (part of the keyboard) produces an audible tone of variable volume. This tone is activated by program control. The tone is sounded on certain error conditions or to alert the user that an event has occurred that may require his attention. The volume is adjustable by the user.

Pointing Device

The Star pointing device is the *mouse*. The mouse rests on the user's working surface with the keyboard, and is moved by the user. As the mouse moves, a displayed cursor reflects its logical position on the display. The cursor takes on different shapes depending on the function being performed. A drawing of the mouse is presented in Figure HOV-3. The various cursor shapes are shown in Figure HOV-4.

The mouse has two buttons labeled SELECT and ADJUST. The SELECT button is on the left. The buttons are the same size and shape.

When using the mouse for most operations, pressing a button down is a separate action from releasing the button. While the button is down, Star provides feedback about what will occur, but the operation is not started until the button is released. Thus, in many cases, if the button is pressed with the mouse cursor pointing at the wrong spot, it can be moved to the correct spot before the button is released. The principal exception to this rule is in scrolling, where the action occurs continuously while the button is held down.

The term *click* connotes the sequential actions of pressing and releasing a mouse button.

Select - The primary function of the SELECT button is to select operands for commands. In certain applications, multiply clicking the SELECT button causes successively larger objects to be selected.

Within text domains, the first click of SELECT selects a character. The second click selects a word (and an adjacent space). The third click selects a sentence, including its punctuation and spacing. The fourth click selects a complete paragraph. The fifth click selects everything in the document, or when operating inside a frame, the entire contents of the frame.

In graphics, the first click selects a control point. Subsequent clicks select successively larger enclosing graphics objects.

The SELECT button is also used to select destinations for the Move and Copy commands, to raise hidden menus, and to select menu items, both in regular window menus and in hidden menus.

Adjust - The ADJUST button is used to adjust selections begun with the SELECT button. In general, Star extends (or contracts) the nearer end of the selection to the point where ADJUST was pushed. Since no ambiguity exists, ADJUST may also be used to select the destination for Move or Copy commands.

Pointing Ahead - The mouse cursor always follows mouse motion, regardless of what processing is going on within Star. If the user presses a mouse button when the system is still processing previous inputs, the mouse action is retained for subsequent processing. However, the user receives no feedback about what will occur. The user should be aware that the object he pointed at might have moved by the time the system actually processes his pointing action.

Rigid Disk

A rigid disk is required with each work station. It contains the user's Desktop (see *System Overview)* and other user-related storage as well as storage for the software components of the system. The rigid disk in Star-1 is available in discrete sizes, small (8.3 million bytes formatted capacity) and medium (23 million bytes formatted capacity). Neither type of disk is removable except by a Xerox representative. The small disk resides in the processor housing, while the large disk requires its own housing, outboard of the processor.

Floppy Disk

A floppy disk drive is also attached to each work station. It allows the contents of removable floppy disks to be accessed in either Star or specific Non-Star formats (see *Removable Storage Media*).

Printer

A printer is optionally attached to a workstation. Several types of printers are available ranging from relatively low-speed character printers to high speed Raster Output Scan (ROS) devices (see *Printing*). For higher speed printers, a large rigid disk must be attached to the same workstation.

Ethernet

A workstation is optionally interfaced to other workstations and different system elements via the Ethernet. This is a high-speed communication line -- 10 million bits per second -- that allows Star workstation users to utilize resources at "network servers" (e.g., printers or non-local file storage) and enables the sending and receiving of mail among workstation users.

Communication Port

The communication port provides a capability for lower speed communication, primarily to enable Star workstations to interact with host computer systems (see *Terminal Emulation*). It includes an RS232-C/DTE interface.

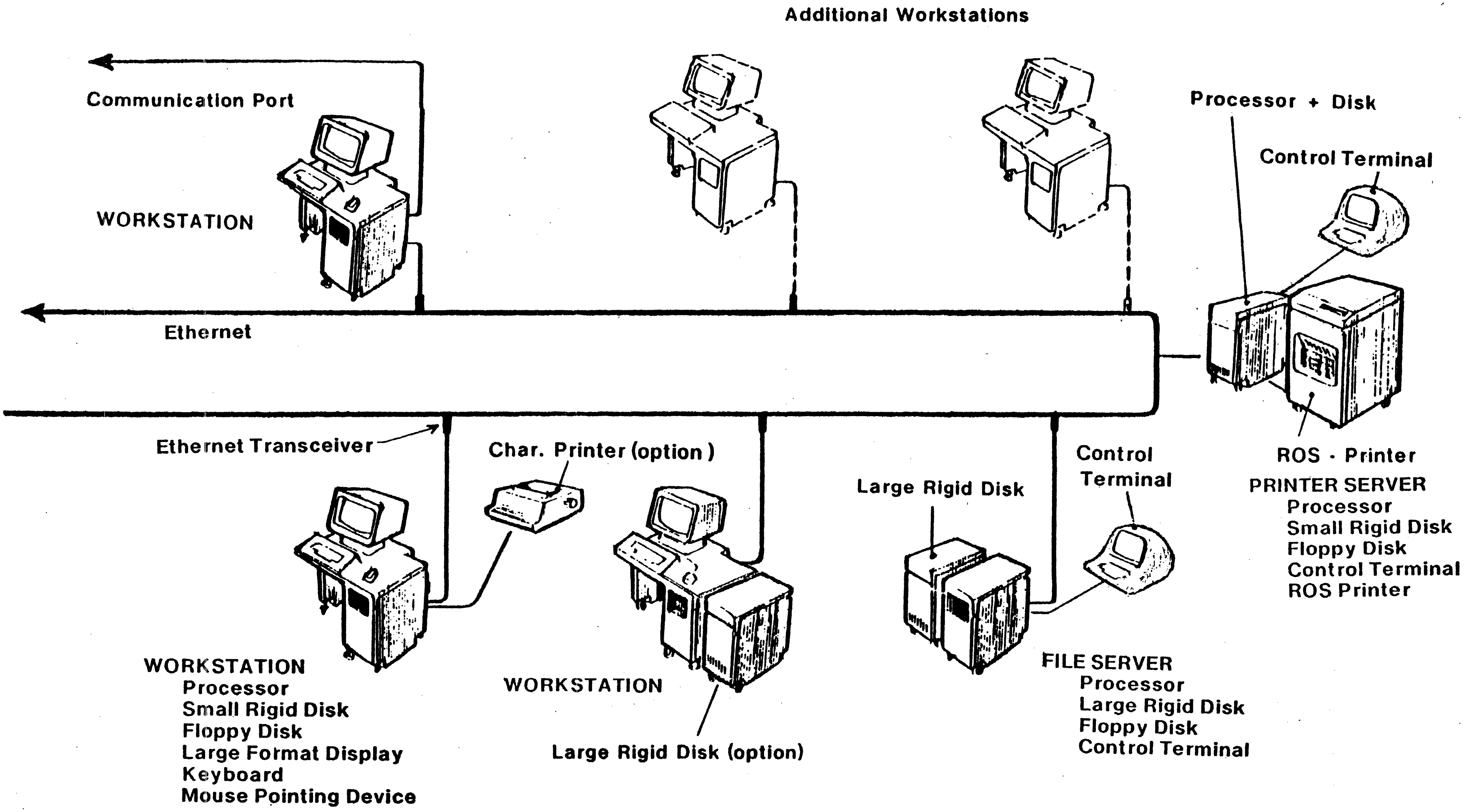
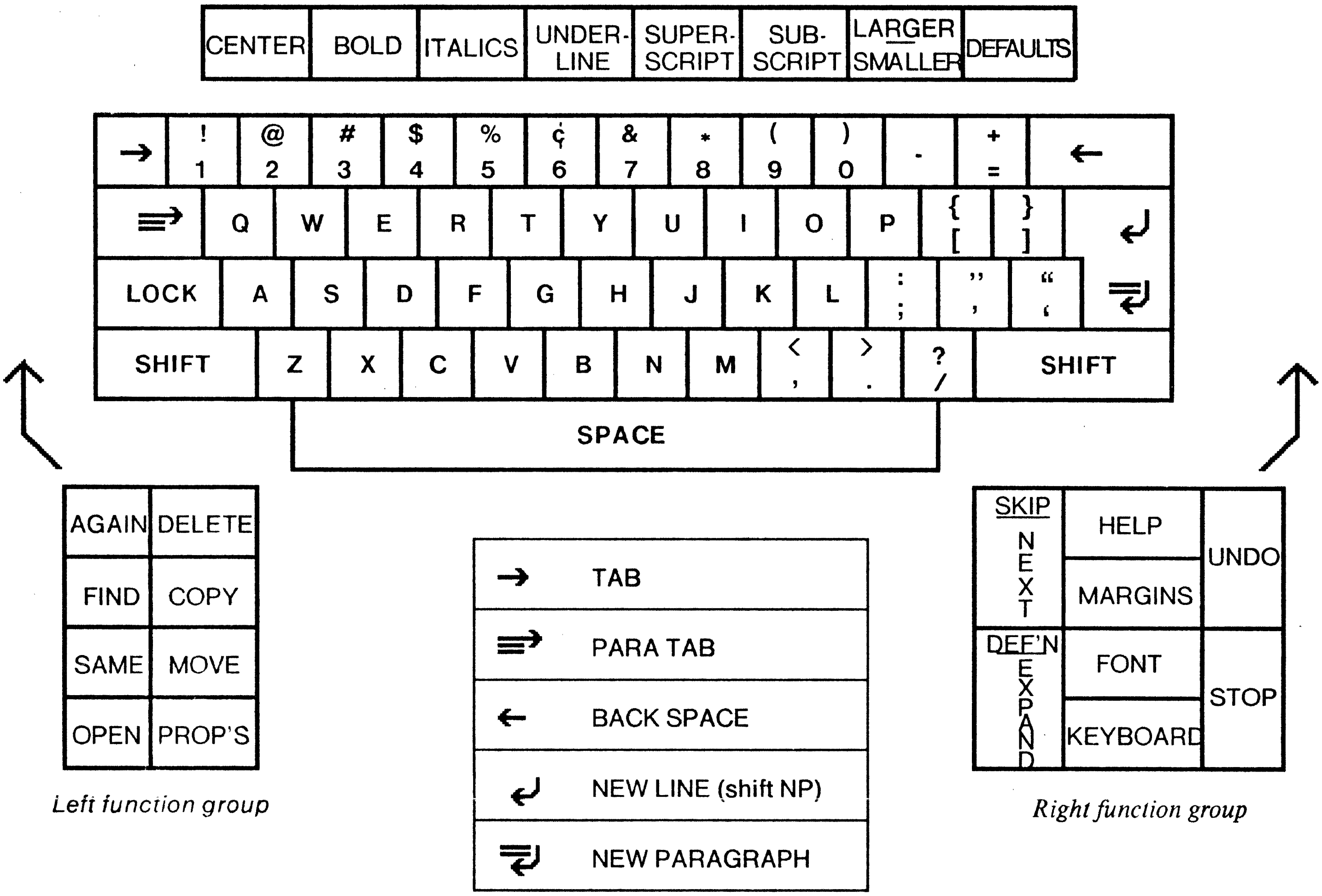


Figure HOV-1 -- Typical Installation



→	TAB
⇛	PARA TAB
←	BACK SPACE
↵	NEW LINE (shift NP)
↵	NEW PARAGRAPH

Left function group

Right function group

Notes:

1. *Electronics subassembly allows for:*
   a) A key to the left of Z
   b) An additional outer column to the left of the left function group
   c) An additional key at each end of the top function group

2. *Keyboard key provides access to additional character sets.*

3. *Font key used to switch fonts.*

4. *Left, top, and right function groups separated from main typing keyboard by ~ 3/8 inch*

5. *All keys except SHIFT and LOCK in the main typing keyboard repeat when held down*

6. *Shift may be used:*
   a) with - to get non-breaking hyphen
   b) with backspace (<--) to get Back Word
   c) with RETURN to get New Line (normal meaning is New Paragraph)
   d) with Next to get Skip; Expand to get Define; Smaller to get Larger
   e) with other top function group keys to get opposite function

Figure HOV-2 -- Star-1 USA Keyboard Layout

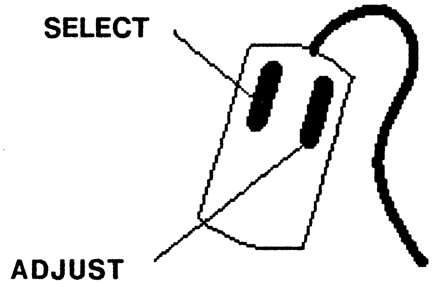


Figure HOV-3 -- Star Mouse

Normal	↑
Move mode	↑■
Copy mode	↑◧
Copy Properties mode (after SAME)	↑□
Transient Menu mode	←
Jumping Mode (Scrolling)	page 10 of 32
File Object Placement	→
Move or Copy Row	⇒
Move or Copy Column	⇑

Figure HOV-4 -- Display of Mouse Cursor (not to scale)

4. STARTING AND TERMINATING A SESSION

This section describes how a user begins and terminates a session at a Star workstation. It presumes that the *System Administrator* has enabled the user as described in *Directories*. This section briefly addresses procedures for powering up the workstation, the initial start up of the Star system software, and transferring contexts between machines.

USER OVERVIEW

At the beginning of every session a user must identify himself to the system. Star uses the identification to associate the Desktop and other resources with the user from session to session.

User identification is also used to determine the proper set of access privileges. Security in Star is based on the access privileges of the individual user with respect to a given file drawer or device. Star security features are optional. That is, a particular installation may choose not to use these security features, in which case all users have the same set of privileges. The ramifications of security are discussed further in *Document Filing* and *Directories.*

Security is enforced by means of a *password*. Each user's password prevents someone else from masquerading as him and accessing his documents and exercising his access privileges. The use of passwords requires the cooperation of users. When a user discloses his password, he potentially compromises the system's security.

DETAILED SPECIFICATION

OBJECTS

Idle Workstation

An idle workstation is one which is in the proper state for a user to initiate a session. To be in this state, the workstation must be powered up, and running the Star system software.

When a workstation is idle, a "*test pattern*" moves randomly about the screen at one second intervals. The pattern consists of a 1" square containing the time of day and the words, "Press any key."

User Identification

Each user has a *name* he uses to identify himself to the system. Star supports user names up to 40 characters long. A name must be composed of characters that appear on keytops, and may not include the "@" or "*" or <comma> characters.

The assignment of names is entirely up to the customer. Each user's name normally corresponds to his real name, but need not. The customer may choose to create a series of pseudonyms where the user name describes the function being performed, e.g. "Payroll." However, a user name must be unique within a directory, and a user may be logged on at only one workstation at one time.

The process of adding new users to the system and the assignment of names is described in *Directories*.

User Password and Account

Each customer may optionally require that a user enter a password and/or an account name (see *Accounting*) when starting a session. Star supports passwords up to 40 characters long. However, extremely long passwords are not recommended due to the difficulty in entering them correctly. The password must be composed of characters that appear on keytops.

Passwords and accounts are assigned when a user is added to the system and may be changed by the system administrator.

ACTIONS

Put Workstation in Idle State

To begin a session, the workstation at which the user wishes to work must be in the idle state. If it is already idle, the user may proceed as described in "Begin Session," below.

If the workstation is powered down, the user must power up the workstation by turning the power switch on the processor maintenance panel to the "1" position. After the system automatically starts itself, the test pattern will appear.

If the workstation is displaying someone else's work, but is not being used, the user must terminate the previous user's session (see below). The test pattern will be displayed. (Only common courtesy need be exercised in terminating someone else's session, as Star will retain the exact state of his work for a subsequent session.)

Begin Session (LOGON)

To begin a session (on a workstation in the idle state), the user presses any key or mouse button. The test pattern disappears and the Logon Option Sheet appears, displaying three text parameters labeled "Name," "Password," and "Account."

The text insertion caret is placed in the "Name" parameter. The user types his name and presses NEXT, advancing the caret to the "Password" parameter. The user types his password, if he has one, terminating it with NEXT, then types his account, if he has one. (Characters typed into the Password parameter are not displayed.) He then invokes Start on the option sheet (or presses NEXT again). The Name, Password, and Account parameters are all checked, ignoring case. If any of the three is invalid, the Message Area displays an appropriate message (e.g., "That is not a valid user-name. Please try again.") and the first invalid entry is selected. NEXT advances the caret to the next parameter, and Start causes all three entries to be checked again.

("Reset" sets all the parameters to empty and places the insertion caret in the Name parameter. "Cancel" returns the system to the idle state.)

When the user has entered a valid name, password, and account (as appropriate), the system displays the user's context at the point he terminated the prior session, except that all open windows have been closed. If the prior session was terminated by a hardware failure, the display may reflect a point slightly before the failure. (A description of the contents of the Desktop for the first session of a new user is given in the *Desktop* section.)

Once the user has successfully logged on, the name he typed into the Logon option sheet (perhaps an alias) and a one-way encoded form of his password are stored in the entry record of his Desktop.

If the user fails to enter any keystroke for one minute while the system is awaiting a user name or password, the workstation returns to the idle state.

Building the User's Desktop at Logon

Ordinarily, a user will work at his "home" workstation, where his Desktop will reside. Occasionally he may log on to a different workstation (e.g., because his "home" machine is unavailable, or because he is at a different site and wishes to read and send mail). The following rules describe the system's behavior in the various cases that may arise:

If the user logs on to a workstation that does not contain his Desktop, the logon facility attempts to retrieve it from his home file server. Once it has been copied to the workstation, the file server copy is deleted. The process of moving the Desktop to the workstation will significantly increase the time required to log on.

If the user's Desktop cannot be retrieved from the server (either because there is no Desktop there, or because the server is down), a new Desktop is created on the workstation for him, after confirmation. The Message Area displays "Cannot find your Desktop. Do you want a blank Desktop to be created for you?" and the menu commands YES and NO. If the user invokes NO, the workstation returns to the idle state.

This new Desktop will be exactly like the environment created for any other "new" user at the same installation. The user may copy his In Basket from the Directory to read his mail, or open a file drawer to which he has access, etc.

If the user's Desktop already exists on the workstation, the logon facility does nothing special, and his Desktop appears as the user last saw it. Normally, the logon process includes checking the user's name (perhaps an alias) and password with the Clearinghouse. If the Clearinghouse is down, the workstation software looks on the rigid disk for the user's Desktop, matching the name and password stored in the Desktop entry record with the user's name and password as entered on the Logon option sheet. If the user enters a name different from the name he used in his previous session (e.g., a different alias), his Desktop will not be found. In this case, the Message Area displays "Cannot find your Desktop under the name <username>. Do you want to log on with a different name?" and the menu commands YES and NO. If the user invokes YES, the option sheet is redisplayed with its three parameters. If he invokes NO, the system returns to the idle state. If the typed-in password (encoded by the one-way function) does not match the encoded password stored on the Desktop, the system responds as described above under "Begin Session".

Terminate Session (LOGOFF)

At any point in a session where command input is legal the user may terminate the session by invoking the END SESSION command in the Desktop menu. All open windows are closed. The Logoff Option Sheet appears, showing two choice parameters. The first, called "Desktop," has three alternatives: Retain (the default), Delete, and Move To File Server. The user may either leave his current Desktop at the workstation as it is, or delete it from the workstation, or move it to the file server (causing it to be deleted from the workstation after the copy is complete). The Desktop may be moved to the file server even if documents and/or folders are directly "on" it, but the user may not delete his Desktop if it contains any data icons or other private data. The Message Area advises the user either to delete such objects or to move them to a file drawer.

The second choice parameter, called "Printing & Mailing" governs the disposition of objects in the print or mail queues (if any) at logoff. The two choices are Finish (the default) and Suspend. If the user chooses Suspend, the current printing and/or mailing transactions are aborted, leaving all unsent objects in the queue to be dealt with the next time the user logs on.

If the user chooses Finish, all objects waiting to be printed or mailed will be sent to the appropriate server before logoff is complete. If either queue cannot be emptied (e.g., because the server is down), the Message Area displays "Your mail cannot be sent (or "Your print request cannot be completed..."). Do you want to end the session anyway?" followed by the YES and NO menu commands. If the user invokes YES, the print or mail request is suspended until the next logon; if he invokes NO, the Logoff Option Sheet disappears, leaving the user logged on. After logoff is complete, the workstation returns to the idle state.

"Reset" sets the Retain and Finish parameters on; "Cancel" causes the Logoff Option Sheet to disappear, leaving the user logged on.

If a session is terminated by a power, hardware, or software failure, it may be resumed once the problem is corrected. This is done as described in the "Begin Session" discussion above. Since Star uses atomic transactions to execute user commands, little or none of the user's work will be lost (unless there was a catastrophic failure of the rigid disk).

CHANGES

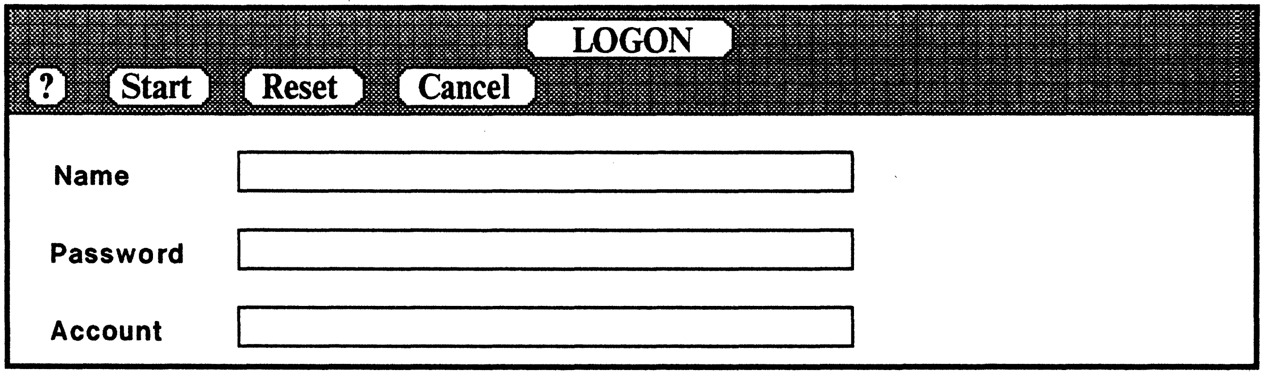
Changed user names and passwords from 100 to 40 characters, prohibited "@," "*," and <comma> in user names.

Clarified logon when Clearinghouse is down: User may log on to a workstation containing his Desktop, using the same name he used at previous logon.

Clarified Logon and Logoff Option Sheets. Print and mail queues emptied (if possible) before Logoff is complete.

Added ability to store and retrieve Desktop from file server.

Added Account parameter.

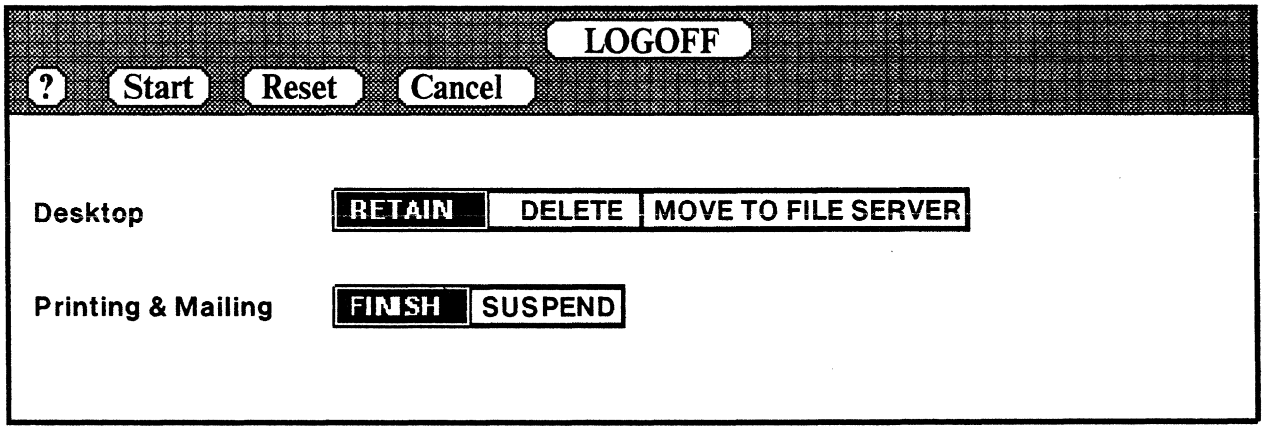


Name

Password

Account

Figure STS-1 - Logon Option Sheet



Desktop [✓ RETAIN] DELETE MOVE TO FILE SERVER

Printing & Mailing [✓ FINISH] SUSPEND

Figure STS-2 - Logoff Option Sheet

5. SYSTEM OVERVIEW

This section presents an overview of the Star system. It describes a set of general commands that apply uniformly throughout Star. These commands simplify the system for users, making expertise gained in one area applicable to others.

This section also describes some system-wide conventions, such as principles for responding to user errors. By discussing these conventions here, the prose in other sections is reduced and the concepts are presented in a consistent fashion.

USER OVERVIEW

Every user's initial view of Star is the *Desktop*. The Desktop resembles the top of an office desk, together with surrounding furniture and equipment. On the screen are familiar office objects, such as file drawers, printers, in-baskets, out-baskets, documents and folders. These objects are represented as small pictures or *icons*.

A user may "open" an icon to deal with what it represents. This enables him to read documents, inspect resources such as the contents of file drawers, etc. When opened, an icon immediately expands to a larger form called a *window*, which displays the icon's "contents." Windows are the principal mechanism for displaying and manipulating information. For example, a document window allows a user to see and edit the text and illustrations in a document.

Commands in Star take the form of an object followed by an action. Specifying the object to be operated on (the "operand") is called making a *selection*. Star provides powerful selection mechanisms, which reduce the number and complexity of commands in the system. Typically, a user will exercise more dexterity and judgment in making a selection than in invoking a command.

The operand is almost always specified before the action to be performed. This makes commands "modeless"; a user can change his mind as to which object to affect simply by changing the selection. No "accept" function is needed to terminate or confirm commands, since invoking the command is the last step. Inserting text does not even require a command; a user simply makes a selection and begins typing. Nevertheless a few commands require more than one selection and hence are modal; for example, the Move and Copy commands require a destination to be specified as well as a source.

The following commands are used throughout Star: Move, Copy, Delete, Show Properties, Copy Properties, Again and Undo. In addition, the Help command is always available. These commands all operate in the same fashion, regardless of the object involved: the user makes a selection and invokes the command. Thus, a user follows the same set of actions to move text in a document as to move a document in a folder.

The Again command repeats the effects of the previous command(s) on a new selection. The Undo command reverses the effects of the previous command; Undo provides protection and increases the user's confidence. Only a few Star commands cannot be repeated or undone.

Although Star is a large and powerful system, much of the command complexity is removed from the keyboard and placed in displayed command *menus.* Every window has a set of commands appropriate for that window's contents. Window menus are simplified by placing specialized commands in *transient menus* which are displayed only when specifically requested by a user.

DETAILED SPECIFICATION

OBJECTS

Desktop

The *Desktop* is the primary view of Star that the user sees on the display screen. The Desktop resembles the top of an office desk, together with surrounding furniture and equipment (see Figure DTP-1). On the screen are a number of pictographs or "icons" representing the user's current working environment. The Desktop surface is displayed as a distinctive gray pattern, called "Desktop gray." The Desktop always occupies the entire display screen; even when windows appear on the screen, the Desktop continues to exist "beneath" them. A complete description appears in *Desktop*.

Icon

An *icon* is a pictorial representation of a Star object which can exist on the Desktop. On the Desktop, an icon is approximately 1" square. Within a container window such as a folder, an icon is approximately 1/4" square. Icon images are intended to suggest familiar office objects (see Figure DTP-2). They may represent documents, folders, record files, file drawers, printers, in- and out-baskets, or various other objects. A complete description appears in *Desktop*.

Message Area

The *Message Area* is always present on the Desktop (see Figure SYS-1). It displays messages to users. It is 15 pixels high and 961 pixels wide, centered at the very top of the screen. It is bordered on all sides by a one-pixel, black line. Each time a user makes a selection or invokes a command, the Message Area is cleared. Users cannot alter its size or location, nor can windows overlay it.

The system displays the following categories of messages in this area.

> **Prompt** - Star displays prompts to suggest the next action a user should take. This is particularly true for modal commands, such as "Please specify a destination" for a

Move command.

Error - Star displays error messages when a user's actions are erroneous or cannot be processed successfully.

Warning - Star displays warning messages when a user's actions can be processed but may not be, in the system's judgment, what the user expected.

Notification - The Attention Service displays notifications of asynchronous events, as described below.

Sticky Message - In certain modes, whenever the message area would normally be cleared, a message is instead displayed reminding the user of the mode he is in.

The philosophy in Star is to be as tolerant as possible of user errors. This promotes the notion of a "friendly" system. To some extent this is provided by the Undo command. Of equal importance is the way Star responds to unusual conditions. In general, Star makes the minimum intrusion except in the case of serious errors. Star takes one of the actions described below, depending on the situation. (Throughout this document, system responses are identified using the phrases in brackets below; no further elaboration is given in describing each response.)

Completed Action without Warning - The system carries out a possibly-erroneous action without displaying a message. These situations occur when (a) the system is incapable of detecting an error; (b) the system detects an error but chooses to interpret it in a standard way (e.g. n-key rollover); or (c) the system prevents an invalid action while providing visual feedback and removing erroneous choices. ["... is ignored."]

Completed Action with Warning - The system carries out a possibly-erroneous action but posts a warning message in the Message Area. The message disappears at the next user action. ["A warning message is displayed."]

Aborted Action with Warning - The system aborts the action, displays an error message in the Message Area, and flashes the Message Area rapidly three times. The message disappears at the next user action. ["An error message is displayed."]

Suspended Action with Warning and Confirmation Required - The system suspends the action, displays an error message in the Message Area, sounds the audible alarm once for a short duration, and flashes the Message Area rapidly three times. The words "Yes" and "No" appear as menu commands following the message. The user must select one of these by pointing to it with the mouse and pushing either mouse button. No other action, including invoking Help, is allowed before confirming or cancelling. ["An error message is displayed, and confirmation is requested."]

Attention Flag

The *Attention Flag* is a reserved area at the left end of the Message Area. When an asynchronous message awaits a user, the letters "ATTN" are displayed in reverse video, bounded by a 32-by-17 pixel rectangle. If no message is queued, the "ATTN" disappears (the Desktop gray pattern is displayed).

Currently there is only one class of asynchronous events that cause the Attention Flag to be turned on.

> **System events** - Hardware or software errors are asynchronous events. See *Recovery and User Diagnostics.*

When a user invokes the Show Message function, the system displays a description of the asynchronous event and turns off the Attention Flag.

Desktop menu

The *Desktop menu* consists of the "?" Help command and a transient menu. They appear at the right end of the Message Area. The "?" command displays Help information describing the message currently in the Message Area, if any. The transient menu contains the commands Convert, Date & Time, End Session (logoff), Show Size, and Test.

Window

A *window* provides a view of the contents of a Star icon. A user can display a window as described in *Desktop.* Except for their contents, all windows are identical in overall appearance, regardless of the icons they represent. A sample Document window is shown in Figure SYS-1. All windows have the following attributes:

> **Border** - A window is bordered by a black line, two-pixels in width.
>
> **Header** - Every window has a header, which contains the title of the window and a menu of window-specific commands. (The header of a Document window may be suppressed via the Set Window command.) The header occupies the full width of the window and is 48 pixels high, including the top border. The remaining 46 pixels are allocated as follows: 3 pixels of gray filler, 19 pixels of title area, 3 pixels of gray filler, 17 pixels of menu area, 3 pixels of gray filler, and a one-pixel lower border. The gray filler is a 50% gray pattern, easily distinguished from the "Desktop gray."
>
> **Title** - The title area contains the name of the icon opened. The title is displayed in a 12-point, bold, sans serif font, centered within the header. If an icon is opened within another window, such as a folder window, the icon's name is appended to the existing title. The width of the title area depends on the width of the contents; it is always as wide as necessary. (However if the contents would be wider than the window's full width, they are truncated and replaced with an ellipsis.)

Command menu - The menu area contains commands which allow users to manipulate the window's contents. Individual menu items are just wide enough to contain the identifying text, which is displayed in 12-point, bold, sans serif font. The commands present in a window menu depend on the type of window. For example, mailing commands are present in the In Basket window. If the window is not wide enough to display all of its commands, the less frequently used commands are placed in the transient menu.

The following commands appear in all window menus. More specialized commands are described in the appropriate sections.

> **?** - This displays Help documentation describing the window and the type of object it represents (see *Help*).
>
> **Close** - This closes the window. It is described under "Close Window".
>
> **Close All** - This appears in the menu of windows which have been opened within a container window, such as a folder window. It closes the window and all higher-level containing windows. It is described under "Close Window".
>
> **<transient menu>** - All windows have a transient menu. The transient menu contains at least the following commands.
>
>> **Set Window** - Invoking Set Window causes the Window option sheet to be displayed, enabling a user to set various characteristics of the window. Changes take effect only while the window is open; when the window is closed, all options return to their default settings. The following options appear for all windows. Some other windows, such as the Document window, have additional specialized options.
>>
>>> **Position:** LEFT/RIGHT - This choice parameter appears for all windows. It specifies whether the window is to appear on the left or right side of the display, as described under *Desktop*. The default is LEFT.
>>>
>>> **Scale:** NORMAL/MAGNIFIED - This choice parameter appears for all windows. It allows a user to magnify the contents of a window (see SYS-7). The default is NORMAL. When Magnified is chosen, the material inside the window appears at 1.5 times normal size in both dimensions. The window's width is increased to 1.5 times the width of the widest portion of the contents being displayed, plus overhead, subject to window size limitations. If the full width cannot be displayed, a horizontal scroll bar appears. The height is unaffected.

Make Document - Invoking Make Document creates a document containing a textual representation of the contents of the window and attaches it to the cursor as if the document had been moved or copied. For example, it might be used to create a list of the contents of a folder window. The entire contents of the window are rendered in the document, not just the portion currently being displayed. The created document is a snapshot only; it is not updated if the object changes. (Make Document does not appear in Document windows.)

Scroll Bar - A white rectangle at the right edge of windows allows vertical scrolling to a new place in the window's contents. The scroll bar is 16 pixels wide and is separated from the window's contents by a black vertical line, one pixel in width (see Figure SYS-4). For horizontal scrolling, another scroll bar appears at the bottom of the window.

Window Size Control Point - A control point in the lower-right corner of the window allows a user to alter the height of the window. The control point is displayed as an open square, 16 pixels on a side. When selected, the square changes to solid black.

Body - The main area of a window displays an icon's contents. The contents vary with the type of icon. The contents of a document are text, graphics, equations, etc. The contents of a folder are documents, record files, and other folders. The contents of a record file are records. And so forth.

Command Menus

Star displays commands in *menus* to augment the commands provided by the function keys on the keyboard. There are two types of menus: visible menus and transient menus.

Visible Menus

Visible menus appear in the header area of windows, property sheets, and option sheets. Star places a command in a visible menu (a) if it is frequently invoked, or (b) to emphasize its availability. If there are more commands than will fit in a visible menu, the less frequently used ones appear in a transient menu at the right end of the visible menu.

Transient Menus

Transient menus (also called "auxiliary menus") are repositories for less frequently used, miscellaneous commands. They are represented by a menu symbol displayed as three horizontal lines (see the upper-right corner of Figure SYS-2). When a user points to this symbol and presses either mouse button, a list of commands appears. The commands are in alphabetical order and are aligned left flush in the list. (Exception: Commands moved to a transient menu from a visible menu, because for example the window became narrower, appear *first* in the list.) When the user releases the mouse button, the commands disappear. The list of commands in a given transient menu never changes, even if the selection changes to different types of objects. This regularity helps users to become familiar with the various menus.

The Selection

Most Star commands operate on the *selection*. The precise definition of what is selectable depends on the context, but, generally speaking, icons are selectable (see *Desktop*), and so is anything displayed in a window. Selectable objects range from a single character to paragraphs, tables, illustrations, documents, and sets of documents (folders). Only one selection can be in effect at a time, although the selection can include multiple objects such as multiple paragraphs. The following summarizes the types of selections. Relevant sections later in this specification discuss each type in greater detail.

Icons - Individual icons may be selected wherever they are displayed. Contiguous sequences of icons may be selected within a container window such as a folder window.

Text - Text may be selected in various units, including characters, words, sentences and paragraphs. Text selections are always contiguous and are confined to a single document.

Graphics - Graphic symbols may be selected individually or multiply. Graphics selections need not be contiguous.

Frames - Frames may be selected individually or as part of an extended text or graphics selection. If selected individually, a frame is treated as a graphics selection. The contents of a frame are selectable, but the selection may not extend beyond the border of the frame.

Tables - Rows, columns, ruling lines, and entire tables may be selected. The contents of a table entry are selectable, but the selection may not extend outside the entry.

Fields - The contents of a field are selectable. A field itself is selectable only when its bounding characters are visible and the document containing the field is not *locked* (see *Field Definition*).

Equations - An entire equation may be selected individually or as part of an extended text selection. The elements in an equation are selectable, but the selection may not extend outside the equation.

Footnotes - A footnote and/or its marker may be selected. The contents of a footnote are selectable, but the selection may not extend outside the footnote.

Property Sheet

A *property sheet* is a window containing a set of parameters. The parameter settings represent the properties of the selection. Most objects in Star have properties. The properties fall into two categories: (1) fixed information about the object, which users may not change, and (2) user-modifiable information. The former are called "information-only" properties, the latter "modifiable" properties.

The method for displaying and changing properties is the same regardless of the type of object involved (see "Show Properties"). A property sheet is a special window in that it may overlay ordinary windows. The property sheet "depends on" the window containing the selection in the sense that if that window is closed, the property sheet also disappears (or the Close is rejected). The border, header, and scroll bar are identical to those for ordinary windows. There is no window size control point. The window title reflects the nature of the selected object, e.g. "Document Properties".

In some cases, the amount of information to be displayed is quite large and is therefore split over several related property sheets. In such cases, a choice parameter named Display appears at the top of the property sheet to allow users to switch between the sheets. The title area remains the same regardless of the Display option chosen.

There are three types of parameters:

> **Text Parameters** - A text parameter contains a textual value. The parameter name is followed by a rectangular outlined area in which a user enters the value. (See "Search For" in Figure TXT-3.) In general the value may be arbitrarily long, including having multiple lines, although some individual parameters restrict their values.
>
> When a numeric value is required in a text parameter, a user may enter either an optionally signed decimal value, possibly including a single decimal point (e.g. "8.5"), or an optionally signed decimal integer followed by a fraction (e.g. "8 1/2"). In the latter case, the integer is followed by one or more spaces, the numerator, a "/", and the denominator.
>
> Most text parameters have a single value. Some, called "repeating" text parameters, have a list of values. Each non-empty value contributes to the overall value of the parameter. There may be any number of values, and they are filled in much like other text parameters, including using the NEXT and SKIP keys. There is always a blank value at the bottom of every repeating text parameter, ready for further type-in.
>
> **Choice Parameters** - A choice parameter has a small number of possible values, which are mutually exclusive. Exactly one choice is in effect at any one time. All the choices are displayed in the property sheet. The parameter name is followed by the choices in connected rectangles, one choice per rectangle. Each choice consists of one or more words or a graphic symbol. The choice in effect is displayed video reversed. (See "Font" and "Size" in Figure SYS-3.)

State Parameters - A state parameter may be either on or off. A state parameter is shown as one or more words or a graphic symbol in a rectangle. If several state parameters are related, they are shown on the same line, with space between them. If a state parameter is on, it is displayed video reversed. (See "Face" in Figure SYS-3.)

Properties that are "information only" are displayed without a surrounding rectangle.

In some cases the selection may be a sequence of objects with differing properties. For example, in a string of characters, some may be boldface and some not. In such cases, the property sheet reflects the properties of the *first* object in the selection. However changing the setting of a parameter affects *all* the objects in the selection.

Property sheets have the following commands in their visible menu.

? - This displays Help documentation describing the property sheet (see *Help*).

Done - The property sheet disappears, and the original selection is restored, reflecting all the changes to its properties.

Apply - The property sheet remains displayed, but the properties that the user has changed are applied immediately to the original selection. This enables users to see the results of property changes, without having to reinvoke Show Properties if corrective action is needed. (This is not present in all property sheets, as it is not necessary.)

Defaults - The property sheet remains displayed, but all parameters in it are restored to their system default values. Default values are described for each property sheet in the relevant sections.

<transient menu> - Property sheets that contain distance-related parameters have a transient menu of commands for setting the units in which the values are expressed. The commands are: Use Inches, Use Millimeters, Use Points, Use Micas, Use 10-Pitch, and Use 12-Pitch. (See Figure FMT-5.)

Option Sheet

An *option sheet* is exactly like a property sheet, except that it allows a user to specify parameters to a command, rather than properties of a selection. For example, the Search command displays an option sheet in which a user indicates what to search for.

Option sheets have the following commands in their visible menu.

? - This displays Help documentation describing the option sheet (see *Help*).

Start - The option sheet disappears, the original selection is restored, and the command begins executing. In some cases, a user can also start the command by pressing a key again, as in the Search command where he can press the FIND key again. This allows him to examine the parameters to a command and, if he doesn't need to make any changes, proceed without using the mouse.

Reset - The option sheet remains displayed, but all parameters in it are restored to the values they had when the option sheet was first displayed.

Cancel - The option sheet disappears, all changes to it are forgotten (i.e. a Reset is done), and the command is aborted.

Fonts

Star is a multiple-font system. Over time there will be a large number of fonts optionally available to customers. There is one font, called the *standard font*, which is available at all workstations. The current choice is 12 point Frutiger. Star uses the standard font to display system-generated information such as error messages and prompts, and it is the default font for typing in text parameters. Star uses the bold version of the font to display window titles, menu commands, and property and option sheet parameters. (Users can use the standard font in documents just like any other font.)

Help Facility

The Help facility provides users with on-line information about the system. When a user wishes such information, he can push the HELP key or click a mouse button over a *Help symbol* ("?") in a menu. (Every visible menu contains a Help symbol.) The Help facility is described in *Help*.

Names

Network Object Names

A *network object* is one that can be addressed from any local or remote connection to an Ethernet. Examples are file servers, printers, users and public distribution lists. A System Administrator registers names for these objects in a Clearinghouse (see *Directories*). A name for a network object has the form "local name@domain name@organization name", for example "John Paul Jones@Palo Alto@Xerox". (For more details, see the Clearinghouse documentation listed in Section 2.)

A *domain* is a grouping of names within an *organization*, either physically or logically. For example, "Palo Alto", "El Segundo", "Dallas" and "Webster" might be domain names within Xerox. Local names must be unique within a domain, and domain names within an organization. Organization names are managed by the Xerox Corporation, which assures their uniqueness. If a company registers its name with Xerox, then (at least conceptually) it can exchange electronic mail with any other registered company.

One local name for each network object is designated as its *full name* (e.g. "John Paul Jones"). As a convenience, in addition to its full name, a network object may have zero or more *aliases* (e.g. "Jones", "John Jones", "J. P. Jones", "JJ"). All aliases are registered in the Clearinghouse, must be unique within the domain (just like the full name), and may be used wherever the full name can be. A network object's *full legal name* has the form "full name@domain name@organization name".

When a user logs on, he may supply either his full name or an alias. He may omit the domain and/or organization, and the system will supply them from the workstation he is presently using. (This works only if the workstation's domain and organization match his own. If he is at a workstation in other than his own domain/organization, he must supply them himself.) The log-on facility then retrieves the user's full legal name from the Clearinghouse.

Star always *stores* the full legal name of all network objects. However it does not necessarily *display* all the parts. Star trims "@domain@organization" or "@organization" if they are the same as the logged-on user's.

When a user types a user name, e.g. in a mail recipient list, he may enter either the full name or an alias. Star *does not* automatically replace user-entered aliases with full names, except at log-on. He may also leave off the domain and/or organization, and his own will be automatically appended (internally) by Star.

Local names may be up to 40 characters (bytes) long, domain and organizations names up to 20 characters (bytes) long. Each may contain any Star character except @, *, and comma. Thus a network object name (local@domain@organization) may be up to 82 bytes long. The actual number of characters allowed depends on the character sets used. Some characters may not be displayable on every workstation or server control terminal.

Other Names

Users assign the names of documents, folders and record files. System Administrators assign file drawer names. Each may be up to 100 characters (bytes) long. Again any Star character may be used.

Document, folder and record file names do not have to be unique (see *Document Filing*); they can even be empty. File drawer names are registered in a file server, not the Clearinghouse, and must be unique within the file server.

ACTIONS

Make Selection

A user can select most displayed objects, including individual characters, words, paragraphs, graphic illustrations, symbols within illustrations, and entire documents. The precise method of selection depends on the object being selected and is described in the appropriate section. In most cases, a user makes a selection by pointing with the cursor to the desired object and clicking the SELECT mouse button. Sometimes he must repeatedly click SELECT to "expand" the selection to the desired level, for example to expand a character selection to a word. Sometimes selection is performed by a command, such as the Search command. There are two cases in which the only way to make a selection is to invoke a command: Select Row and Select Column in tables.

When an object is selected, it is highlighted, generally by reversing the video image. Each time a selection is made, the previous selection is unselected and unhighlighted. There is never more than one selection at a time.

For most Star commands, a user must make a selection before invoking the command. If he has not done so, the command is ignored and an error message is displayed.

Adjust Selection

Once a user has made a selection, he can adjust (shorten or lengthen) it. For example, he can adjust a text selection to include more or fewer characters. In graphics, adjustment means that graphics objects are added to or removed from a multiple selection. In certain contexts the selection cannot be adjusted, for example when selecting icons on the Desktop.

To adjust a selection, a user points with the cursor to an object and pushes the ADJUST mouse button. Within a text selection, this causes the nearer end of the selection to be extended to the cursor position. If the user moves the mouse before releasing the mouse button, the end of the selection moves with the cursor until the button is released. In other contexts, the means of adjusting selections differs slightly. Specific techniques are described under *Frames*, *Tables* and *Graphics*.

Move

A user can move an object from one location to another by selecting it and pressing the MOVE key. The cursor changes to the shape shown in Figure HOV-4, the original selection is unhighlighted, and the message "Please indicate a destination with either button." is displayed. For text selections, a dotted underline appears beneath the original text.

At this point the user may scroll vertically or horizontally to bring the desired destination onto the screen, but he may not make a selection or invoke any other command except Stop.

The user indicates the destination by moving the cursor toward the destination and pressing either mouse button. As long as the mouse button is held down, the system provides feedback as to the exact location of the destination. For example, when moving text, the system inverts the currently indicated destination, appends a caret to the inverted image, and provides automatic spacing control, as described in *Text Editing*. When moving icons or graphic objects, the system moves the entire object in real time. When the user releases the mouse button, the object is placed at the destination. For text, the source is deleted, and the dotted underline disappears. The newly positioned object remains selected.

If the user indicates an illegal destination for the selected object, the cursor changes into a question mark (?) until the user moves it to a legitimate destination. If the mouse button is released while the cursor is in the question mark form, the Move operation is terminated without further action.

Move is a *mode*. If the user wishes to abort it after pressing the MOVE key, he may do so by invoking Stop before specifying the destination. The original selection will become highlighted again and will remain unchanged. The user can cancel a completed Move command by invoking Undo (although not all Moves are undoable -- see *Undo and Again*).

Any object that can be selected can be moved, subject to protection and access controls, but there are restrictions as to what can be moved where. In general, an object can be moved from one place to another place containing the same type of objects. For example, a paragraph can be moved into text in a document, but it cannot be moved into a folder window. These restrictions are discussed in the relevant sections of this document.

Copy

A user can copy an object to a location by selecting it and pressing the COPY key. The Copy command proceeds exactly as with Move, except that a slightly different cursor shape is used (see Figure HOV-4), and the selection is not deleted from its original location. When the user releases the mouse button, the copy becomes selected, and the original object is unselected. The original object is not changed in any way.

Delete

A user can delete an object by selecting it and pressing the DELETE key. There is no selection following a Delete operation, except that in text an insertion caret remains at the point of deletion. The Delete command is always reversible with Undo, even when a document has been deleted.

Any object that can be selected can be deleted, subject to protection and access controls. Some system objects such as the Directory are permanently protected against deletion; these are described in the relevant sections of this document.

Show Properties

A user can display the properties of any object by selecting it and pressing the PROPERTIES (PROP'S) key. The appropriate property sheet is displayed. The original selection is unhighlighted and underlined with a dotted line (or enclosed in a dotted box in the case of equations, tables and graphics). If the property sheet contains one or more text parameters, the contents of the first text parameter are automatically selected, ready for editing. If the first text parameter is empty, a caret is placed in it. If there are no text parameters, the caret is left where it was.

Change Properties

A user can alter property sheet parameters by pointing at them with the mouse or by typing. Altering a parameter may cause other parameters to appear or disappear in the sheet. For example, in Figure SYS-3, changing the FONT parameter causes the SIZE parameter to show only the valid sizes for that font.

> **Change Text Parameter** - The contents of a text parameter are entered and edited as ordinary text. (Editing details appear in *Text Editing*.) A user can step through text parameters by pressing the NEXT key, activating each one in turn, just as he does with fields and table entries. The contents of the parameter are selected or, if the parameter is empty, an insertion caret is placed in it. When the caret is in the last text parameter, NEXT reactivates the first one. Except in repeating text text parameters (see below), the SKIP key is treated exactly like NEXT while in a property sheet.
>
> Invoking NEXT in a repeating text parameter activates the next parameter in the list. (For example, see the Tab Settings property sheet, Figure FMT-3.) The last parameter is always empty. Whenever a user types in the last parameter, a new blank one is automatically created and put at the end. If he invokes SKIP in a repeating text parameter, the caret/selection is moved out of the list to the next text parameter, if any.
>
> **Change Choice Parameter** - A user selects the desired choice by pointing to it and pressing and releasing the SELECT mouse button. That choice becomes the one in force, and the previous choice is turned off.
>
> **Change State Parameter** - A user changes a state parameter by pointing to it and pressing and releasing the SELECT mouse button. The parameter is turned on if it was previously off, and off if it was previously on. (But see the discussion in *Text Editing* of text properties in the Search, Substitute, and Define Abbreviation option sheets.)

By their nature, choice and state parameters cannot be set to invalid values. However, a user can enter an invalid value in a text parameter. When a text parameter has been modified, the user proceeds by (1) pressing NEXT, (2) setting or selecting another parameter with the mouse, (3) making a selection elsewhere, or (4) invoking a command. If a user makes a parameter invalid and then pushes NEXT, an error message is displayed, and any input typed ahead is discarded. The parameter remains selected so that the user can correct it. If he pushes NEXT again without changing the parameter, the system resets it to its previous value, and a warning message is displayed. If the user proceeds via any means other than NEXT, the validity checking is deferred until he attempts to close the property sheet.

As a user modifies parameter settings, the properties of the selection are not actually changed until he invokes Done or Apply. Both commands cause the properties to take effect and the display to be immediately updated.

While a property sheet is displayed, all information in the window containing the original selection is "read-only." A user may open and close other windows, edit in other windows, make selections and copy them into the property sheet, scroll a window (except in the unlikely case that the window is completely overlayed), and in general perform any action that does not modify the contents of the window containing the original selection. Commands that attempt to do so are rejected with an error message.

When the user invokes Done, the property sheet disappears, and the original selection is automatically reselected. This is the normal conclusion of Show Properties. A user may also signify that he is done by pressing PROPERTIES again, which is equivalent to invoking Done. This allows him to examine the properties of a selected object and, if he doesn't need to make any changes, proceed without using the mouse.

Copy Properties

A user can copy the properties of one object to some other (compatible) object by selecting the object to be changed and pressing the SAME key. The cursor changes to the shape shown in Figure HOV-4, the original selection is unhighlighted, and the message "Please indicate the source of the properties with either button." is displayed. For text selections, a dotted underline appears beneath the original text.

At this point the user may scroll vertically or horizontally to bring the desired destination onto the screen, but he may not make a selection or invoke any other command.

When the user presses either mouse button, the object under the cursor, if it is of the same type as the selection, is highlighted. The user may move from object to object before releasing the button; each object is highlighted in turn and the previous one unhighlighted. If the source and destination objects are not compatible, an error message is displayed. When the button is released, certain of the properties are copied to the original selected object, which is rehighlighted. The properties copied are object-dependent, and are described in subsequent sections. If an object has a *name* (i.e. an icon, field, frame or table column), the name is never copied.

Copy Properties is a *mode*. If the user wishes to abort it after pressing the SAME key, he may do so by invoking Stop before specifying the destination. The original selection will become highlighted again and will remain unchanged. The user can cancel a completed Copy Properties command by invoking Undo.

Change Option Sheet Parameters

A user changes option sheet parameters exactly like property sheet parameters. Invoking Start causes the command to begin executing with the values specified.

Again

A user can repeat the last command or sequence of commands on a new selection by pressing the AGAIN key. Again will repeat all of the commands done between the time the current selection was made and the time the previous selection was made. For example, Again can be used to continue a Search or Substitution, repeat a Search or Substitution over a new range, repeat an insertion, repeat a Copy or Move to the same destination, repeat a "replace from the keyboard" operation (Delete + type-in), repeat a series of property changes made with the top row of keys, repeat a series of property sheet changes, etc. The specific commands that can be repeated are described in *Undo and Again.*

Undo

A user can undo the effect of most commands by pressing the UNDO key. The system attempts to restore the user's environment to the state that existed before the command was invoked. Not all commands are totally reversible. A table of commands for which Undo is applicable is provided in *Undo and Again.*

A user may undo several successive commands by repeatedly invoking Undo. This is also described in *Undo and Again.*

Open Window

A user opens a window by selecting an icon and pressing the OPEN key. The contents of the icon are displayed in a window, which overlays all or part of the Desktop. If the icon was on the Desktop, its image becomes white indicating that that location on the Desktop is reserved. If the icon was in another window such as a folder window, the opened window overlays the container window, and an additional command, Close All, appears in its window menu. Additional information on opening windows appears in *Desktop.*

When a window is opened, its contents are displayed at their beginning (i.e. it is scrolled to the top). If a Locked document is opened, the contents of the first *field*, if any, in the fill-in order of the document are automatically selected. Otherwise the first *object*, if any, in the window is automatically selected -- for example, the first character in a document, the first icon in a folder, etc.

Close Window

A user closes a window by invoking the Close command in its menu. The window disappears, the icon returns to its former appearance, and the icon becomes selected, ready for subsequent commands. If it originally resided in another window such as a folder window, that window is redisplayed. Additional information on closing windows appears in *Desktop.*

Adjust Window Height

A user can adjust the height of a window by selecting its window size control point (in the lower-right corner) and invoking the Move command. When he depresses a mouse button to indicate a destination, a horizontal line the width of the window follows the cursor. The line is constrained such that neither the window being moved nor the one below is reduced to less than 208 pixels in height including the header, or 160 pixels in height if margins are being displayed. When the user releases the mouse button, the window is set to the indicated height. The window immediately below it, if any, also adjusts its height so as to maintain the normal 12-pixel separation between windows.

When a window is closed or a new window opened, the automatic partitioning rules (see *Desktop*) override any user-specified adjustment.

Scroll Vertically

Scrolling vertically moves the contents of a window upward or downward within the window. A user invokes vertical scrolling by placing the cursor in the white rectangle at the right edge of the window. When the cursor enters the rectangle and remains inside for at least 300 milliseconds, the *vertical scroll bar* appears. [ALTERNATIVE: Implementation may preclude changing the display in the absence of a button click. If this is the case, the scroll bar would always show the five regions.] For normal windows, the vertical scroll bar is divided into five regions, as shown in Figure SYS-4. From top to bottom, they are the *downward region*, the *previous page region*, the *jumping region*, the *next page region*, and the *upward region*. If a property sheet is scrollable, its vertical scroll bar contains these regions, also.

> **Continuous Scrolling** - The upward and downward regions are used to invoke continuous scrolling. To scroll the contents upward (toward the end), a user points into the upward region and depresses either mouse button. The region is highlighted, and the contents of the window move upward at a constant rate so long as the button is held down, even if the cursor moves out of the upward region. Star scrolls the contents of a window "smoothly," a few raster lines at a time, in such a way that the contents remain readable. The reverse operation, moving the contents downward (toward the beginning), is performed in the same way with the downward region. When the start or end of the contents is reached, scrolling stops and a message is displayed.
>
> If part of a window's contents are obscured by overlayed windows (e.g. a property sheet), the window cannot be continuously scrolled. Attempting to do so results in an error message.
>
> **Turning to Next/Previous Page** - The N and P regions are used to "turn the page." For example, if a user pushes a mouse button while the cursor is in the N region, the region is highlighted. When he releases the button, the region is unhighlighted and the next page of the document is displayed. Similarly, the P region is used to display the previous page. For unpaginated documents and other windows, the N and P regions are used to display the next or previous window-full,

respectively. If the cursor is moved outside the N and P regions before the button is released, no page turning action will occur.

Jumping - Within a document window, if a user wishes to see a particular page, he places the cursor in the jumping region and pushes either mouse button. A small diamond shape appears in the scroll bar to indicate the relative position of the presently displayed page or window-full within the document (see Figure SYS-5). In addition, a box is attached to the cursor, showing the page number of the page that will be displayed if the button is released. The user may hold the button depressed and move the cursor up or down in the jumping region until the box contains the desired page number, at which time he releases the button. If the cursor is moved outside the jumping region before the button is released, no jumping action will occur.

A user may jump immediately to the beginning or end of a document by moving the cursor to the top or bottom of the jumping region and clicking either mouse button. The top and bottom areas are each allocated 16 pixels in the jumping region to make them easy to hit.

If the document has not been paginated, the diamond shaped symbol and the cursor will still indicate the respective present and jump-to relative window-full positions within the document, but no page number is displayed with the cursor. This is also the case for non-document windows, such as folder windows.

Scroll Horizontally

Scrolling horizontally moves the contents of a window to the left or right within the window. This allows users to see information that is too wide to fit. When a window's contents are too wide, a white rectangle appears along the bottom of the window. A user invokes horizontal scrolling by placing the cursor in that rectangle. When the cursor enters the rectangle and remains inside for at least 300 milliseconds, the *horizontal scroll bar* appears. The horizontal scroll bar is divided into three regions, as shown in Figure SYS-6. From left to right, they are the *rightward region*, the *jumping region*, and the *leftward region.*

The leftward region and rightward region operate similarly to the upward and downward region in the vertical scroll bar. To scroll the contents of the window leftward, a user points into the leftward region and depresses either mouse button. The region is highlighted, and the contents of the window move to the left at a constant rate as long as the button is held down, even if the cursor moves out of the leftward region. [ALTERNATIVE: Satisfactory performance may require moving the information in increments of up to an inch, rather than continuously.] As new area is exposed, it is left blank until the button is released. Scrolling stops automatically when the right margin becomes visible. The rightward region operates in a similar fashion, stopping when the left margin becomes visible.

The jumping region also operates in the same manner as its counterpart in the vertical scroll bar, except that no page number appears with the cursor. If a user clicks either mouse button while the cursor is between the two ends of the jumping region, the displayed document is repositioned horizontally, corresponding to the relative position of the cursor between the region end-points. If a user points into the left end of the jumping region and clicks a mouse button, the document including the left-most edge is displayed. Similarly, clicking a mouse button with the cursor in the right end of the jumping region causes the right edge of the document to be displayed. The left and right areas are each allocated 16 pixels in the jumping region to make them easy to hit.

Invoke Visible Menu Command

To invoke a command in a visible menu, a user points with the cursor to the desired command and pressing either mouse button. The command is highlighted (video reversed). If he moves the cursor to a different command, the previous command is unhighlighted and the new one highlighted. If he moves the cursor off of all commands, no command is highlighted. When the user releases the mouse button, the highlighted command, if any, is invoked. When the command completes, it returns to its normal unhighlighted appearance.

Invoke Transient Menu Command

To invoke a command in a transient menu, a user points with the cursor to the appropriate menu symbol and presses either mouse button. As long as he holds the button down, a list of commands is displayed adjacent to the menu symbol. Except for the Desktop menu, the list is always centered vertically with respect to the menu symbol and positioned horizontally just to the left of it. Items within the list are left flush. If the user moves the cursor over a command in the list, that command is highlighted (video reversed). If he moves the cursor to a different command, the previous command is unhighlighted and the new one highlighted. If he moves the cursor out of the list entirely, no command is highlighted. When the user releases the mouse button, the menu disappears and the highlighted command, if any, is invoked.

Show Message

When the Attention Flag is on, a user can invoke the Show Message function to determine the event that triggered the attention. To do so, the user points to the Attention Flag and presses either mouse button. While the button is held down, the Attention Flag is displayed video reversed. When he releases the mouse button, a summary message is displayed in the Message Area. There is only one summary message for each class of asynchronous events (described under "Attention Flag"), not one for each individual event. After a user has displayed a summary message, that class of asynchronous events will not turn on the Attention Flag again until another event in that class occurs.

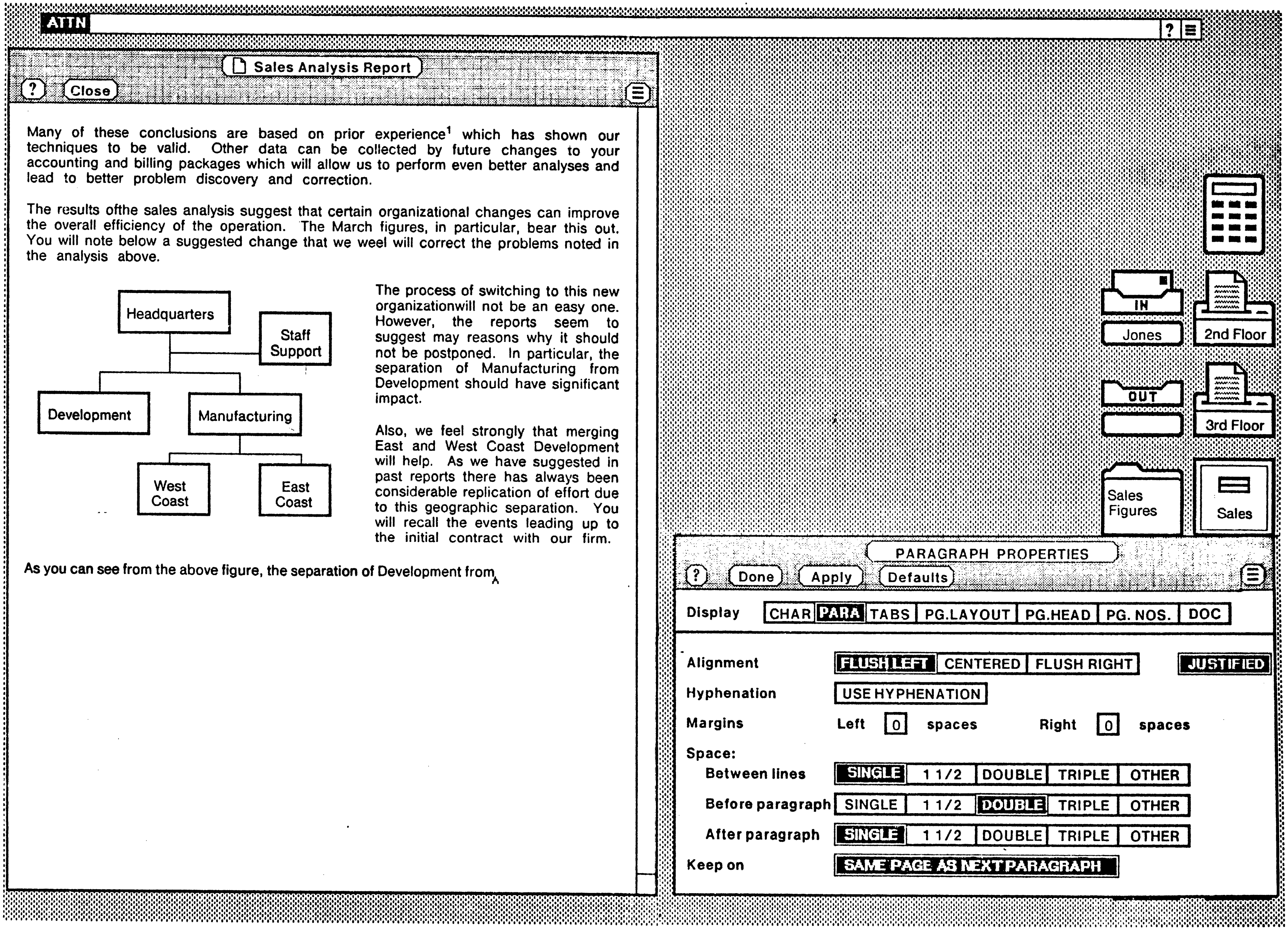
A user may proceed in any fashion he desires after reading the summary messages. He may continue work or interrupt his task to expand the context relevant to the message.

Stop

The STOP key is used to stop interactive activities such as Search and Substitute commands or to abort modes such as Move and Copy modes. (The Move and Delete commands are used to stop background activities such as formatting a document for printing.) When Stop is invoked, all changes made by the stopped command remain in effect. For example, if a Substitute command is stopped, all substitutions already made remain in effect. The user may invoke Undo if he wishes to return to the state that existed before the stopped command was invoked.

Clear Screen

A user may temporarily remove his information from the display by turning down the brightness control located on the front of the display.



Many of these conclusions are based on prior experience[1] which has shown our techniques to be valid. Other data can be collected by future changes to your accounting and billing packages which will allow us to perform even better analyses and lead to better problem discovery and correction.

The results ofthe sales analysis suggest that certain organizational changes can improve the overall efficiency of the operation. The March figures, in particular, bear this out. You will note below a suggested change that we weel will correct the problems noted in the analysis above.

The process of switching to this new organizationwill not be an easy one. However, the reports seem to suggest may reasons why it should not be postponed. In particular, the separation of Manufacturing from Development should have significant impact.

Also, we feel strongly that merging East and West Coast Development will help. As we have suggested in past reports there has always been considerable replication of effort due to this geographic separation. You will recall the events leading up to the initial contract with our firm.

As you can see from the above figure, the separation of Development from

PARAGRAPH PROPERTIES

? Done Apply Defaults

Display: CHAR [✓ PARA] TABS PG.LAYOUT PG.HEAD PG. NOS. DOC

Alignment: [✓ FLUSH LEFT] CENTERED FLUSH RIGHT [✓ JUSTIFIED]

Hyphenation: USE HYPHENATION

Margins: Left 0 spaces Right 0 spaces

Space:

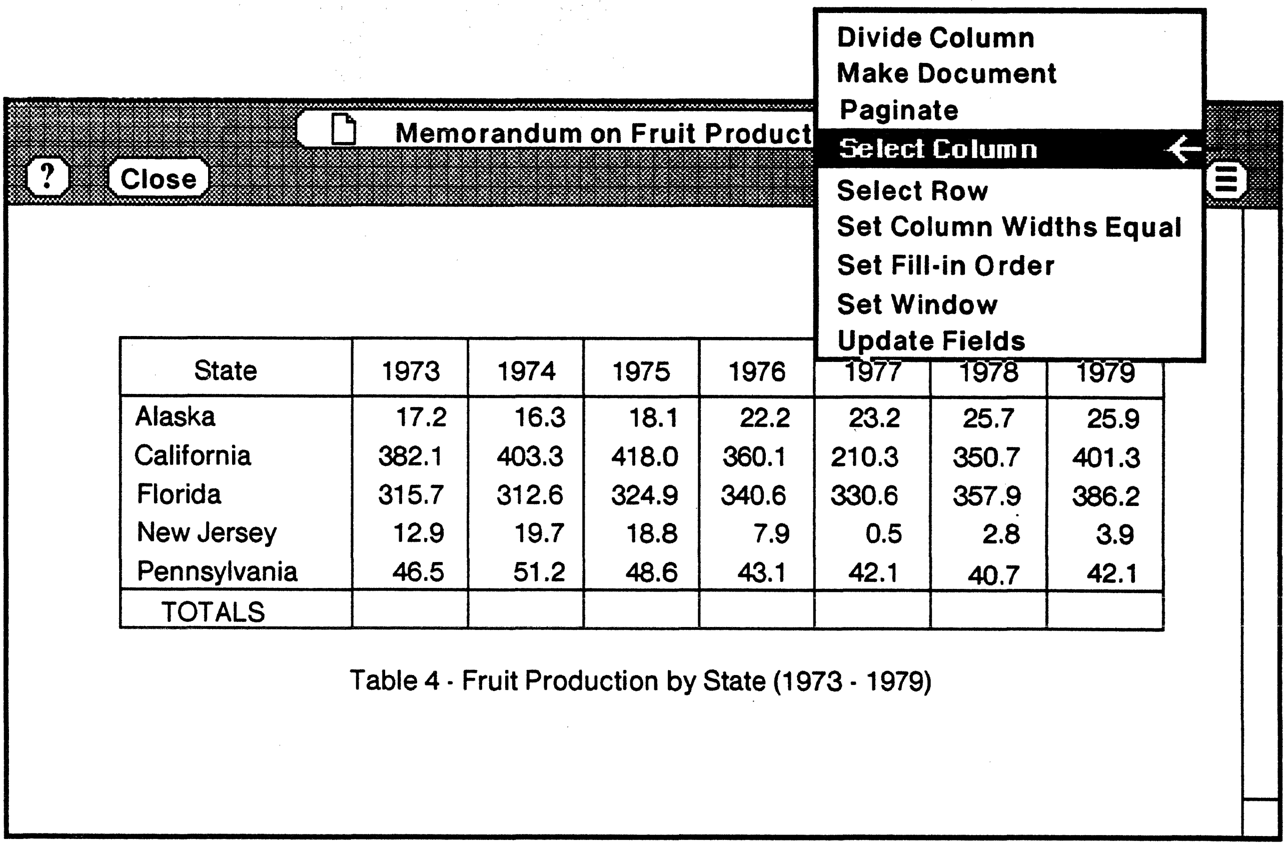
Between lines: [✓ SINGLE] 1 1/2 DOUBLE TRIPLE OTHER

Before paragraph: SINGLE 1 1/2 [✓ DOUBLE] TRIPLE OTHER

After paragraph: [✓ SINGLE] 1 1/2 DOUBLE TRIPLE OTHER

Keep on: [✓ SAME PAGE AS NEXT PARAGRAPH]

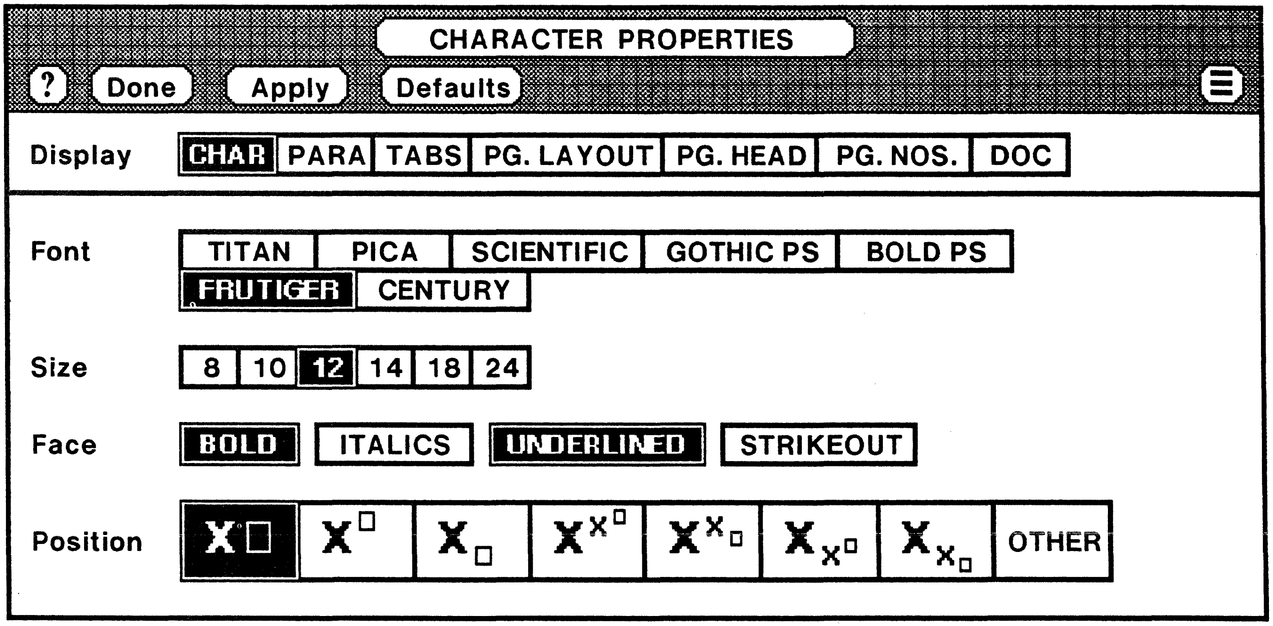
Figure SYS-1 -- The "Desktop" showing icons, document window, property sheet and message area.



State	1973	1974	1975	1976	1977	1978	1979
Alaska	17.2	16.3	18.1	22.2	23.2	25.7	25.9
California	382.1	403.3	418.0	360.1	210.3	350.7	401.3
Florida	315.7	312.6	324.9	340.6	330.6	357.9	386.2
New Jersey	12.9	19.7	18.8	7.9	0.5	2.8	3.9
Pennsylvania	46.5	51.2	48.6	43.1	42.1	40.7	42.1
TOTALS							

Table 4 - Fruit Production by State (1973 - 1979)

Figure SYS-2 -- Transient (auxiliary) menu for a Document window



Display [✓ CHAR] PARA TABS PG. LAYOUT PG. HEAD PG. NOS. DOC

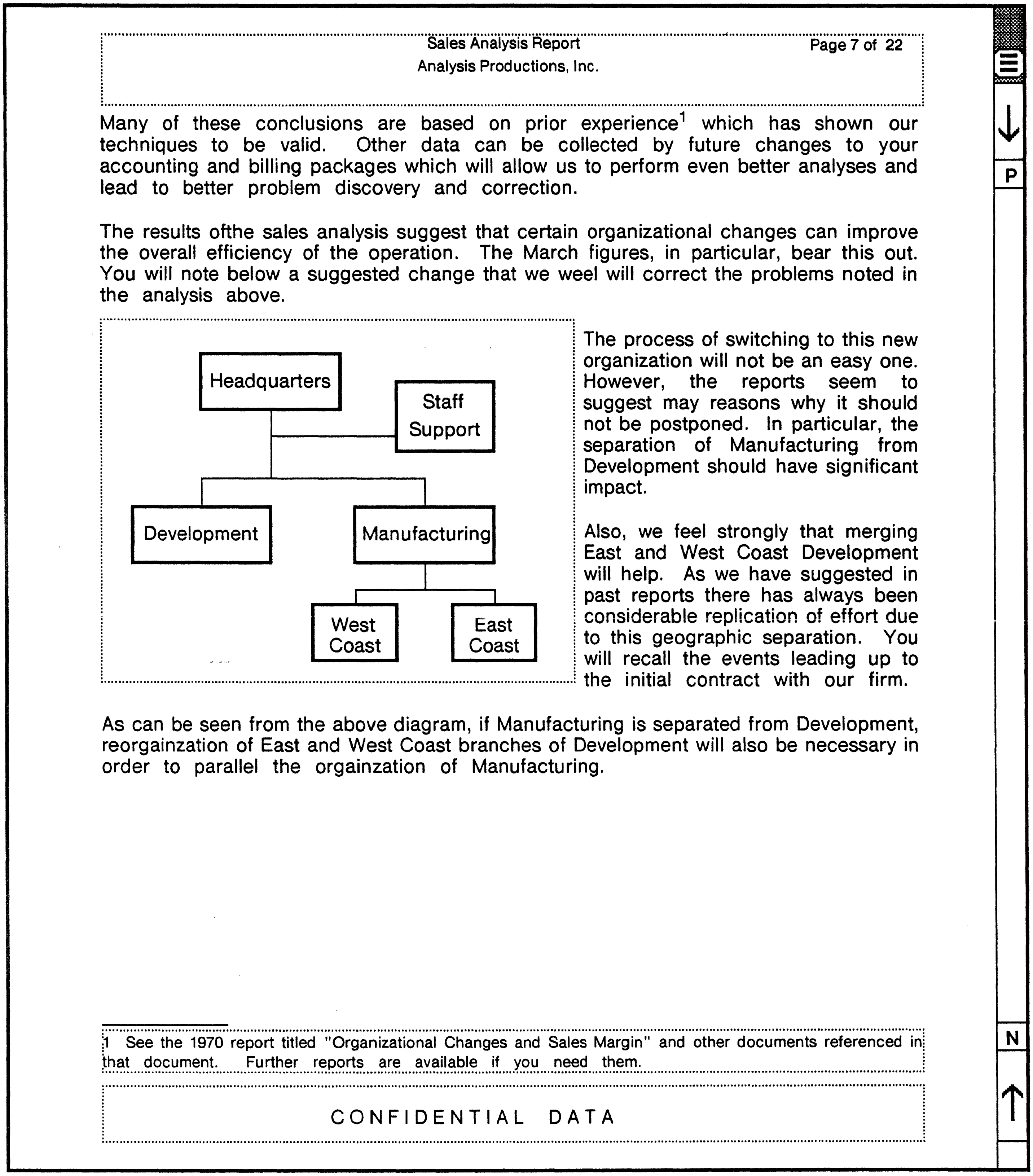
Font TITAN PICA SCIENTIFIC GOTHIC PS BOLD PS [✓ FRUTIGER] CENTURY

Size 8 10 [✓ 12] 14 18 24

Face [✓ BOLD] ITALICS [✓ UNDERLINED] STRIKEOUT

Position [✓ X□] $X^{□}$ $X_{□}$ $X^{X^{□}}$ $X^{X_{□}}$ $X_{X^{□}}$ $X_{X_{□}}$ OTHER

Figure SYS-3 -- Sample property sheet



Sales Analysis Report Page 7 of 22

Analysis Productions, Inc.

Many of these conclusions are based on prior experience[1] which has shown our techniques to be valid. Other data can be collected by future changes to your accounting and billing packages which will allow us to perform even better analyses and lead to better problem discovery and correction.

The results ofthe sales analysis suggest that certain organizational changes can improve the overall efficiency of the operation. The March figures, in particular, bear this out. You will note below a suggested change that we weel will correct the problems noted in the analysis above.

The process of switching to this new organization will not be an easy one. However, the reports seem to suggest may reasons why it should not be postponed. In particular, the separation of Manufacturing from Development should have significant impact.

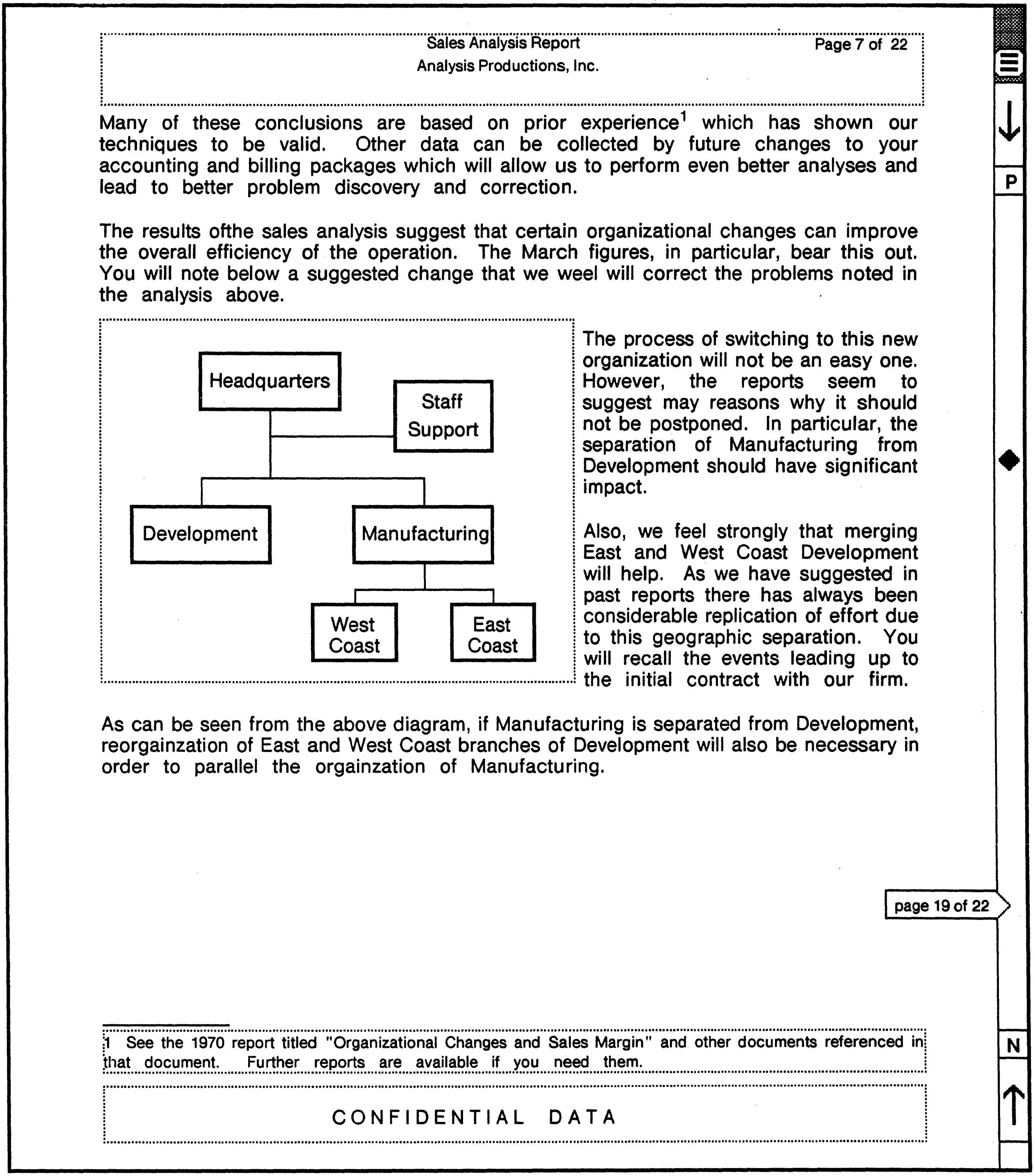
Also, we feel strongly that merging East and West Coast Development will help. As we have suggested in past reports there has always been considerable replication of effort due to this geographic separation. You will recall the events leading up to the initial contract with our firm.

As can be seen from the above diagram, if Manufacturing is separated from Development, reorgainzation of East and West Coast branches of Development will also be necessary in order to parallel the orgainzation of Manufacturing.

[1] See the 1970 report titled "Organizational Changes and Sales Margin" and other documents referenced in that document. Further reports are available if you need them.

CONFIDENTIAL DATA

Figure SYS-4 -- Document window showing the scroll bar
(Show MARGINS and Show BOUNDARIES are also turned on.)



Sales Analysis Report Page 7 of 22
Analysis Productions, Inc.

Many of these conclusions are based on prior experience[1] which has shown our techniques to be valid. Other data can be collected by future changes to your accounting and billing packages which will allow us to perform even better analyses and lead to better problem discovery and correction.

The results ofthe sales analysis suggest that certain organizational changes can improve the overall efficiency of the operation. The March figures, in particular, bear this out. You will note below a suggested change that we weel will correct the problems noted in the analysis above.

The process of switching to this new organization will not be an easy one. However, the reports seem to suggest may reasons why it should not be postponed. In particular, the separation of Manufacturing from Development should have significant impact.

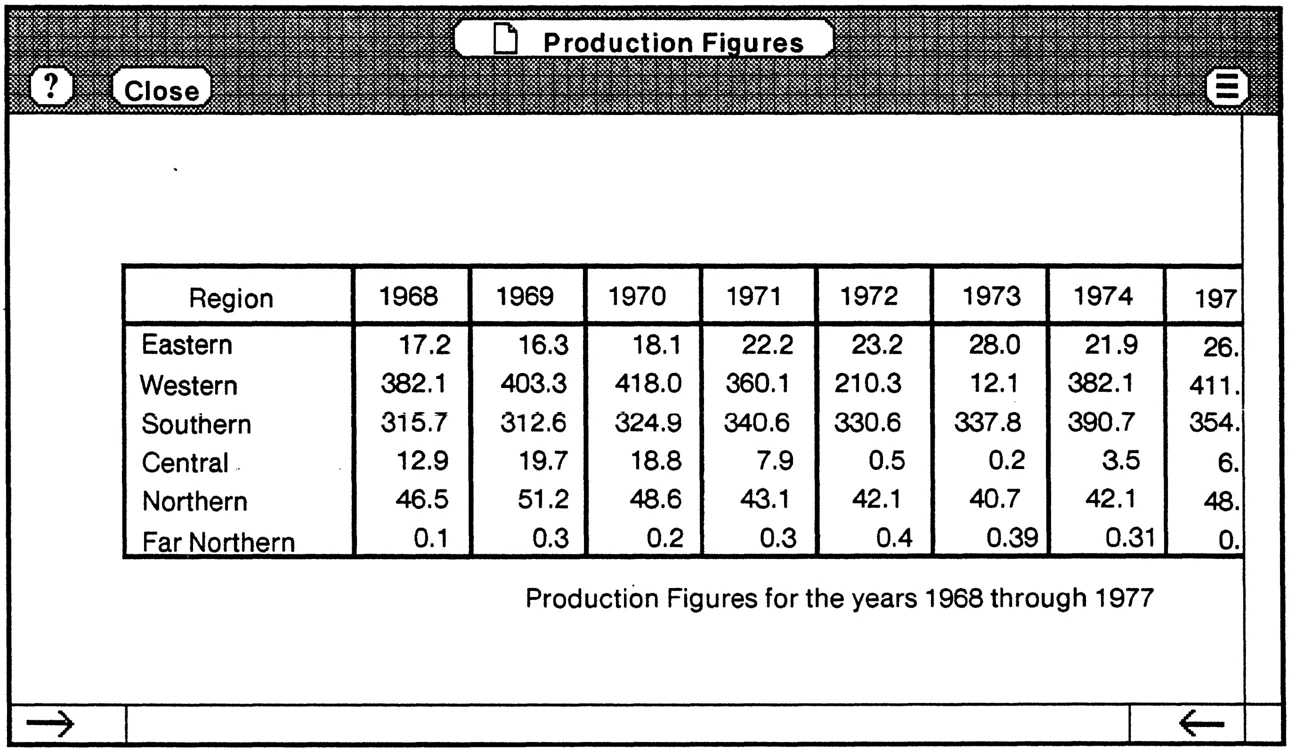
Also, we feel strongly that merging East and West Coast Development will help. As we have suggested in past reports there has always been considerable replication of effort due to this geographic separation. You will recall the events leading up to the initial contract with our firm.

As can be seen from the above diagram, if Manufacturing is separated from Development, reorgainzation of East and West Coast branches of Development will also be necessary in order to parallel the orgainzation of Manufacturing.

[1] See the 1970 report titled "Organizational Changes and Sales Margin" and other documents referenced in that document. Further reports are available if you need them.

CONFIDENTIAL DATA

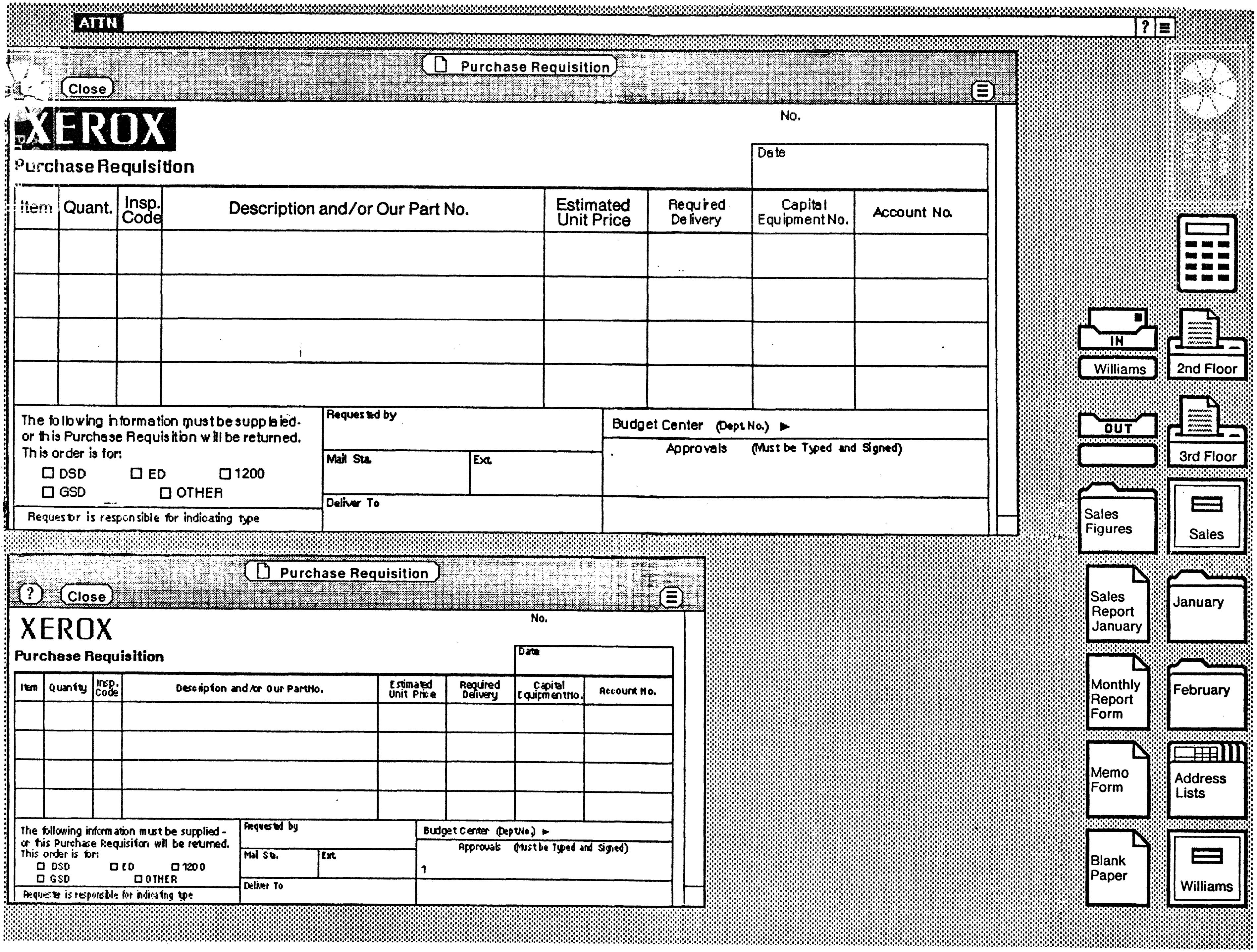
Figure SYS-5 -- Document window showing page scrolling



Region	1968	1969	1970	1971	1972	1973	1974	197
Eastern	17.2	16.3	18.1	22.2	23.2	28.0	21.9	26.
Western	382.1	403.3	418.0	360.1	210.3	12.1	382.1	411.
Southern	315.7	312.6	324.9	340.6	330.6	337.8	390.7	354.
Central	12.9	19.7	18.8	7.9	0.5	0.2	3.5	6.
Northern	46.5	51.2	48.6	43.1	42.1	40.7	42.1	48.
Far Northern	0.1	0.3	0.2	0.3	0.4	0.39	0.31	0.

Production Figures for the years 1968 through 1977

Figure SYS-6 -- Document window showing Horizontal Scroll Bar



XEROX

Purchase Requisition

No.

Date

Item	Quant.	Insp. Code	Description and/or Our Part No.	Estimated Unit Price	Required Delivery	Capital Equipment No.	Account No.

The following information must be supplied - or this Purchase Requisition will be returned. This order is for:

□ DSD □ ED □ 1200
□ GSD □ OTHER

Requestor is responsible for indicating type

Requested by

Mail Sta. Ext.

Deliver To

Budget Center (Dept No.) ▶

Approvals (Must be Typed and Signed)

XEROX

Purchase Requisition

No.

Date

Item	Quantity	Insp. Code	Description and/or Our PartNo.	Estimated Unit Price	Required Delivery	Capital Equipment No.	Account No.

The following information must be supplied - or this Purchase Requisition will be returned. This order is for:

□ DSD □ ED □ 1200
□ GSD □ OTHER

Requester is responsible for indicating type

Requested by

Mail Sta. Ext.

Deliver To

Budget Center (DeptNo.) ▶

Approvals (Must be Typed and Signed)

1

Figure SYS-7 -- Normal and magnified Document

6. DESKTOP

The *Desktop* is the primary view of Star that the user sees on the display. The Desktop is not only a collection of images on the screen, but is also part of a basic approach to presenting office information in a recognizable and easily remembered way. This section provides a detailed description of the use of the Desktop.

USER OVERVIEW

The Desktop is the user's working environment, where current projects reside. On its surface are displayed *icons*, which represent documents, folders, file drawers, printers, in-baskets, out-baskets, etc. They can be moved about the Desktop, to arrange them in groups, or, in the case of documents and folders, to deposit them in another object such as a file drawer. Icons can be *opened* to view their contents through *windows.* Documents and folders may be left on the Desktop indefinitely, just as on a real desk.

DETAILED SPECIFICATION

OBJECTS

The Desktop

The Desktop can always be thought to occupy the entire display. Even when windows appear to fill the screen, the Desktop continues to exist "beneath" them.

The Desktop surface is represented by a distinctive gray pattern, called "Desktop gray." Each Desktop icon occupies one of an invisible, discrete matrix of 1" squares that fill the screen. The large-format display provides room for 11 rows by 14 columns. A typical Desktop is shown in Figure DTP-1.

The user may move any icon from one square to another when the Desktop is in view, but two icons cannot occupy the same square. Although the action of moving a document to a Desktop resource such as the printer involves transferring the document icon to the same square as the printer, this action immediately causes the printer to "absorb" the document, queuing the document for printing.

Initially a user's Desktop consists of a completely blank document ("blank paper") and a Directory. From this initial state, the user can modify his Desktop by adding documents and folders from the Directory as his work dictates. The user can add resources such as printers and In/Out-Baskets by copying them from the Directory. Star places no limits on the complexity of the user's Desktop except the limitation imposed by absolute screen area.

At the option of the system administrator (or Marketing), a user's Desktop contents can be pre-initialized to contain such things as a folder of blank forms, etc.

Desktop Objects - Icons

The Star objects which may reside on the desktop are called icons. All icons can be either closed or open; windows are open icons. When an icon is "on" the Desktop, it is represented by a 1" square symbol. Their forms are shown in Figure DTP-2. When the icon is contained in another object--e.g., a document in a folder--it is represented by a 16x16 dot symbol. When the icon is open, it's title appears in the window header following a 16x16 dot icon.

The large icon consists of a picture of the object and the name associated with the object (if any). For data icons (see below), the name consists of up to three lines, each of which begins a word; if the last word on a line cannot be displayed in its entirety, the last character is replaced by a tilde (~). Other icons can display only one line, which contains as much of the object's name as will fit, using the tilde to indicate truncation.

Star provides the following types of icons. Each type of icon is described more fully in the section shown in parentheses.

Document (*Documents*) - A document is the fundamental object in Star. A Star document corresponds to the standard notion of what a document should be. It can be rendered on paper, shown on the screen, sent to another user, filed or deleted.

A document most often contains text, but can include graphic illustrations, form fields, statistical tables, etc. All objects referred to as *forms* in Star are documents.

Folder (*Document Filing*) - A folder is normally a collection of documents. However, folders can contain any objects represented by data icons: record files and other folders, as well as documents. Folders are the principal mechanism for grouping documents in Star. Folders can reside on the Desktop or can be filed, printed or mailed to another user.

(The term "data icon" is used in this Functional Specification as an aid in describing actions dealing with objects that are collectively represented by this class of icons: documents, record files, and folders. Thus, we may say that a folder contains data icons, as a concise means of referring to the collection of objects grouped by that folder.)

File Drawer (*Document Filing*) - The file drawer is a place to store data icons. A user's file drawer is distinguished from the other storage places (folders, and the Desktop itself) only in that certain other users may access the drawer's contents. Other users cannot directly access objects on a user's Desktop. (Initially, filedrawers reside only on file servers.)

The organization of the file drawers or folders is entirely up to the user. It can vary from a simple linear list to a multi-level hierarchy.

Printer (*Printing*) - A printer icon on the Desktop represents a printing resource. The actual printing device may be directly connected to the user's processor or may be remotely located. The user may choose to keep more than one printer icon on his Desktop to represent different printers. All data icons can be placed in a printer.

In-Basket and Out-Basket (*Electronic Mail*) - These objects provide the principal mechanism for receiving and sending data icons to other users. A data icon placed in the Out-Basket is queued for sending to another user; data icons are received in the In-Basket and can be moved or copied from the In-Basket to the Desktop itself or to other Desktop objects.

Floppy Disk (*Removable Storage Media*) - The floppy disk icon can be used to move a document between Star and the media inserted in the device. This provides a means of manually moving documents to other systems or for storing documents off line.

Directory (*Directories*) - The Directory serves as a source of icons representing shared, possibly remote objects that the user may access. When opened, it lists these objects in categories. The categories include users to whom mail can be sent, files that may be shared, printers, etc.

Record File (*Records Processing*) - A record file is a collection of information organized as a set of records. Frequently this information will be the variable data from forms. These records may be sorted, filtered (subset via logical criteria), and used to produce reports. In general, a record file is treated just like a document relative to printing, mailing, and so forth.

Calculator (*Calculator*) - The Calculator operates like a hand-held calculator. It allows the user to visually perform arithmetic operations. The Calculator may be used by itself, as an aid in form fill-in, or to indicate arithmetic operations in CUSP.

User and Distribution List (*Electronic Mail*) - User and Distribution List icons copied to the Desktop can be used (in 1982) as the source of addresses for electronic mail.

Spelling Checker (*Spelling Checking*) - The Spelling Checker is used to check the spelling of text in documents and record files. Facilities are provided for assisting the manual correction of misspelled words.

Data Icons

Documents, folders and record files are collectively called *data icons*, to make describing them easier. In general, anything that can be done to one type of data icon can be done to all. They can all be filed in file drawers or folders, mailed through out-baskets, printed with printers, etc. The few exceptions are described in the appropriate sections below. Note: Both the "real" and "reference" forms (see below) of documents, folders and record files are considered to be data icons.

(The term "data icon" is used in this Functional Specification as an aid in describing actions dealing with icons. It also helps to keep those actions consistent. It may or may not be presented to users.)

Real and Reference Data Icons

Data icons may be *"real"* or *"reference"*. A reference data icon has no contents, only a "pointer" to the real data icon which must be contained in a remote file drawer. This provides for the convenient sharing of documents, record files and folders.

Reference icons have the same shape as their referents. For example, a reference document icon has the same shape as a real document icon. However the reference forms are distinguished by being grey; users can always ascertain by inspection whether an icon is real or reference. (Reference icons are discussed more fully in the section on *Document Filing*.)

In general, the rules for reference icons are the same as for the corresponding real icons. A reference icon can be moved or copied to any destination that is legal for the corresponding real icon. Therefore real and reference icons are not distinguished in most places in this document. Only in cases where their semantics differ (such as *Document Filing*) is the distinction mentioned.

Size and Positioning of Windows

When an icon is opened to become a window, its size and position are initially determined by Star, although the user can adjust them, within bounds.

The display screen can accommodate up to 6 ordinary windows, divided into two columns of three windows each (see DTP-3 and DTP-4). When more than one window is open, the area allocated to each may be reduced below its default size.

The default width for a document window is the distance between the margins, plus at least two pixels of white space on each side, plus the width of the scroll bar and window borders (see DTP-5). The width including borders is constrained to be one more than a multiple of 4 pixels. Where the margin widths vary, the window's width is the widest space between margins for the page(s) being displayed, plus the overhead. The width of a window is not changed during a scrolling operation, but is adjusted as necessary when scrolling ceases. For non-document windows, the default width is determined by the implementation.

The default height for a document window is exactly the height of the main text area on a formatted page, plus at least two pixels of white space at the top and bottom, plus the height of the window header. It is constrained to be a multiple of 4 pixels. If the sum exceeds 780 pixels, the main text area is reduced to comply with this limit.

Although ordinary windows always overlay the Desktop, they never overlay one another. The system maintains 12 pixels vertical separation and 11 pixels horizontal separation between ordinary windows, and allows the Desktop to "show through" this gap. However, Property Sheets, Option Sheets, and the Help window may overlay ordinary windows.

Size and Positioning of Property Sheets and Option Sheets

When a property sheet or option sheet is displayed, it always occupies the lower-right corner of the screen, except when this would cover the upper left corner of the selection, in which case it is moved to the lower-left corner of the screen (see DTP-6 and DTP-7). It always overlays the current contents of the screen without disturbing them. Property/option sheets never participate in the automatic partitioning scheme as ordinary windows do.

All property/option sheets are subject to a maximum width of 509 pixels, and a maximum height of 772 pixels. The lower-left corner is positioned at coordinate (514, 801), or at (2, 801) if on the left side.

A property/option sheet is surrounded by a black border, two pixels in width. The header and vertical scroll bar (if necessary) are identical to those on ordinary windows. Horizontal scroll bars and window-size control points never appear.

Size and Positioning of Help Window

When the Help window is displayed, it always occupies the upper right corner of the screen, unless the stimulus for the Help window is also on the right side, in which case the Help window appears on the left side. ("Stimulus" means the current selection, unless Help is called through a Help symbol on a menu. In the latter case, the Help symbol is the stimulus.) The window overlays normal windows without disturbing them or participating in screen partitioning.

The Help window is subject to a maximum width of 509 pixels, and a maximum height of 772 pixels. The upper-left corner is positioned at coordinate (514, 29), or at (2, 29) if on the left side.

The Help window is surrounded by a black border, two pixels in width. The header and vertical scroll bar (if necessary) are identical to those on ordinary windows. Horizontal scroll bars and window-size control points never appear.

ACTIONS

In this section, reference is made to a set of system-wide commands that are used in all Star contexts. The commands are Move, Copy, Delete and Show Properties, together with making selections. These commands (and other system-wide commands) have already been described in *System Overview*. It is assumed in the following descriptions that the meanings of these commands are fully understood.

Select Icon

The user selects a desktop icon by pointing at it and pressing and releasing the SELECT button. Star responds by video reversing the display of the icon. Only one icon may be selected.* Moving from icon to icon with SELECT held down changes the selection.

ALTERNATIVE: Multiple icons may be selected with the ADJUST button.

Show Icon Properties

The user may display and modify the properties of an icon by selecting the icon and pressing the PROPERTIES (PROP'S) key. The properties for each icon are described in the appropriate sections. For example, the properties of file drawers and folders are described in the *Document Filing* section.

Move Icon

An icon may be moved within the Desktop. The user selects an icon, presses and releases the MOVE key. The cursor changes to the standard "move cursor" shape (Figure HOV-4), and the message area displays the message, "Please indicate the destination with either button." When the user presses either mouse button, the mouse cursor is blanked, and the entire icon jumps to the cursor position and moves continuously until the user releases the button. The icon overlays all other information on the screen. The position of the icon when the mouse button is released determines the destination of the icon.

If the destination is an empty square, the icon appears in the new location when the button is released. The icon remains selected.

Data icons on the Desktop may also be moved "onto" other icons. This has the effect of placing the object being moved "inside" the second icon. For example, a document may be moved to the file drawer, thus "filing" the document. Similarly, moving a document onto a printer icon puts it in the queue for printing. (Note: Moving a reference data icon onto a printer prints the corresponding real icon.)

If the user points at another icon when specifying the destination of a move, the destination icon flashes to indicate that it is about to "absorb" the object being moved. If the user releases the mouse button while the cursor is "over" the icon, the object being moved disappears from the display. There is no selection following this operation, since the moved object is no longer visible.

Data icons may also be moved from the Desktop into windows. This is similar to moving an icon "onto" another icon and conforms to the table below. The exact meaning is described in the appropriate sections.

If the center of the icon enters a container window that can accept it, the cursor changes to a miniature of the icon being moved and the large icon disappears from the display. If the cursor subsequently leaves that window, the large icon is restored.

Icons may be moved from windows onto the Desktop. These operations are identical in terms of the sequence of operations the user performs when moving icons within the Desktop. However, in this case, the icon begins moving in small form and grows to large form when it leaves the window.

There are restrictions as to what may be moved "onto" a given icon or window. The following table indicates legal destinations for icons to be moved (copied).

Destination	Icons
Desktop	Any icons (as discussed above)
Directory	None
File Drawer	Data icons
Folder	Data icons
In/Out-Baskets	Data icons
Printer	Data icons
Spelling Checker	Data icons
Star Floppy Disk	Data icons
Record File	Data icons
Document	None

If the user points to an illegal destination when specifying the destination of a move, the cursor becomes a question mark. If the user releases the button over an illegal destination, an error message is displayed and the icon returns to its original location.

ALTERNATIVE: If implementation considerations preclude moving icons in their large form, the fallback position is as follows: When the user presses the SELECT button, the cursor changes to a miniature of the icon being moved and the icon disappears from the display. The miniature form of icon is used throughout the move operation.

Copy Icon

The user may copy icons from one place on the Desktop to another by selecting the icon and using the COPY key. The destination is any empty square. This command operates exactly like Move except that the original icon is not deleted. Upon completion, the copy is selected.

Icons can be copied either to an empty square or "onto" other icons. Icons may also be copied between the Desktop and windows. These operations are identical to those described under "Move Icon."

When an icon is copied, the entire object it represents is copied as well. Thus, copying an icon representing a folder containing five documents creates a second folder containing copies of the five documents. If the disk space for the user's Desktop is insufficient to hold the copy, an error message is displayed.

Icons can be copied, also, from a directory to the Desktop. Normally, this is how a user gains access to system resources, such as printers, file drawers, blank forms, etc. Once a function icon (e.g., printer, file drawer) has been copied to the user's Desktop, the Desktop copy of the icon cannot be copied again; it can, however, be moved to a different location on the Desktop. There is no such constraint on data icons copied from a directory, such as blank forms; these may be recopied on the desktop, edited, etc. (See *Directories*.)

Create New Document

Creating a new document is accomplished by making a copy of an existing document. In the most basic case, the user makes a copy of a blank sheet of paper. Every user begins with such a document named "Blank Paper." He can create a new document from this or any other document with the Copy command described above.

Typically a user will have a number of "blank documents" or forms on his Desktop as document icons (like form pads). Each time he wants a new document, he simply makes a copy of it. (Star will provide a variety of blank forms. The set of forms to be provided is to be defined by Marketing. The individual system administrator will be able to add to this set of forms and to pre-initialize user Desktops with desired subsets of these forms.)

Delete Icon

The user may delete any icon on the Desktop except the Directory. Should the user attempt to delete the Directory, an error message is displayed. Special rules govern the deletion of icons representing shared or protected objects (see *Document Filing*).

Open Window

The user opens a window by selecting an icon and pressing the OPEN key. The icon then changes into a window, which overlays all or part of the Desktop, and the "interior" of the icon on the Desktop becomes white. This "ghost" indicates that the square is reserved. When the window is closed, the window disappears and the icon returns to its former position.

The display screen can accommodate up to 6 open windows, three on each side. When a window is opened, it always is placed on the left side if fewer than three windows are currently there. The first window is positioned with its origin at (2, 29). Its maximum possible height is 780 pixels, including the header. Its width is the page width of the information being displayed, or 1021 pixels, whichever is less.

When subsequent windows are opened, the screen is automatically divided into equal-height horizontal bands. Windows on the left are positioned with their x-coordinate at 2 pixels; windows on the right are positioned at 518. The widths of the windows are determined as described above. However, if any windows are positioned on the right, all windows -- both left and right -- are truncated to a maximum width of 505 pixels. (This may mean that horizontal scrolling is required to view their complete contents.)

The heights and y-coordinates of a set of windows are as follows:

Number of Windows on a Side	Height	Y-coordinate(s) of Origins
1	780	29
2	384	29, 425
3	252	29, 293, 557

If a window is opened when there are already three windows on the left side, the new window is assigned to the right side. (The assignment to the right is "sticky" in that the window will remain there even if windows on the left are subsequently closed.) If the right side also contains three windows, the attempted Open command is rejected with an error message.

Move Window Right/Left

The user may request that a window be moved to the opposite half of the screen via the Set Window command. The associated option sheet contains a choice parameter labeled Position, with choices Right and Left. In no case may a window be moved to a side that already contains three windows.

When the user selects Right, the window is repositioned with its origin at x-coordinate 518. The window's size and y-coordinate depends on how many other windows are being displayed on the right side of the screen. Whenever a window is on the right, all windows are reduced in width, and the space is reapportioned as described under "Open Window." Thus, if there were three windows on the left and none on the right, after moving one to the right, each window on the left would occupy half of the available space, and the window on the right would occupy the entire right side.

If the user changes the Position parameter from Right to Left, the x-coordinate of the origin is set to 2, and the screen is reapportioned as above. When the last window is moved from right to left, all windows are free to enlarge themselves to their default width.

Invoke Desktop Menu Command

Commands in the Desktop menu are invoked as with any transient menu (see DTP-8). The effects of these commands are described below.

Convert - Causes the selected document to be converted to Star format, as described in *Electronic Mail.*

Date and Time - Causes the system to display the current time and date in the Message Area. It is presented in the format "February 24, 1978 12:01 pm."

End Session - Logs the user off from Star, as described in *Starting and Terminating a Session.*

Show Size - Causes the size in characters of the selected object to be displayed in the Message Area. Also, the amount of free and used space within that object's container (those with a fixed allocation: workstation, filedrawer, or in-basket) is shown. For example, Size: 54,000 characters, Available: 200,000, Total used: 2,000,000.

Test - Initiates the verification diagnostics routines, described under *Recovery and User Diagnostics*.

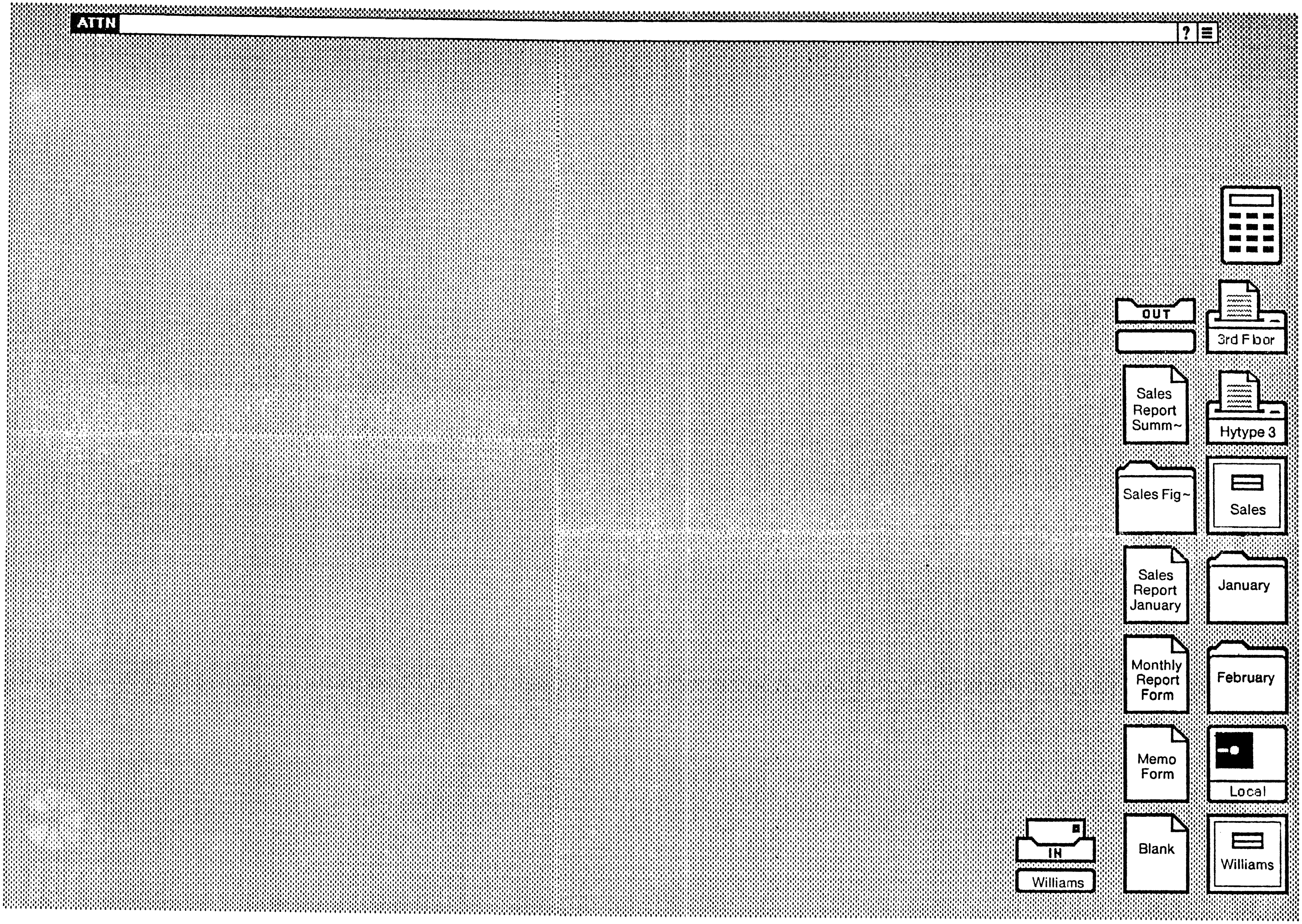


Figure DTP-1 -- "Desktop" showing icons and message area

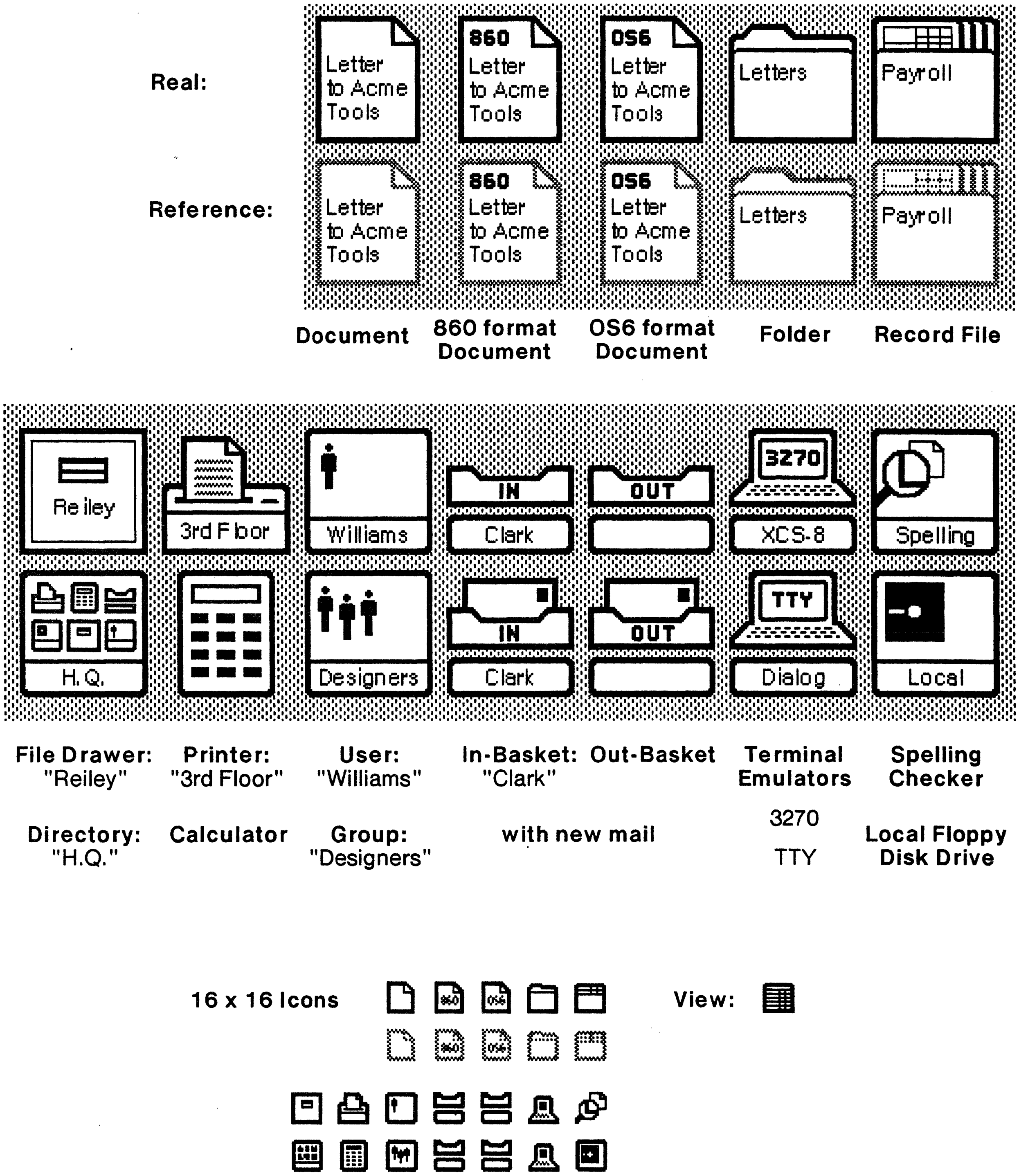
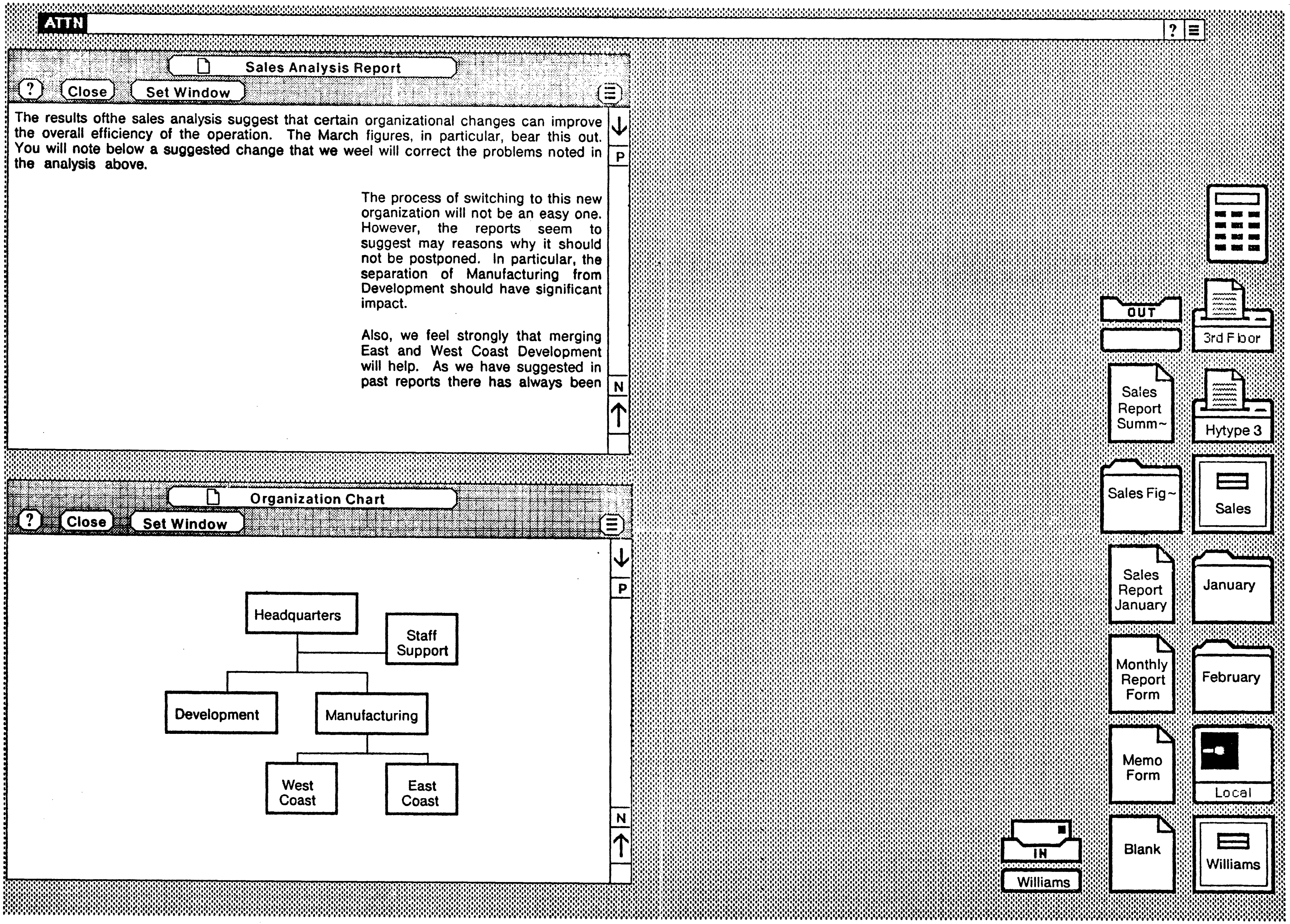


Figure DTP-2 Icons for Star Objects

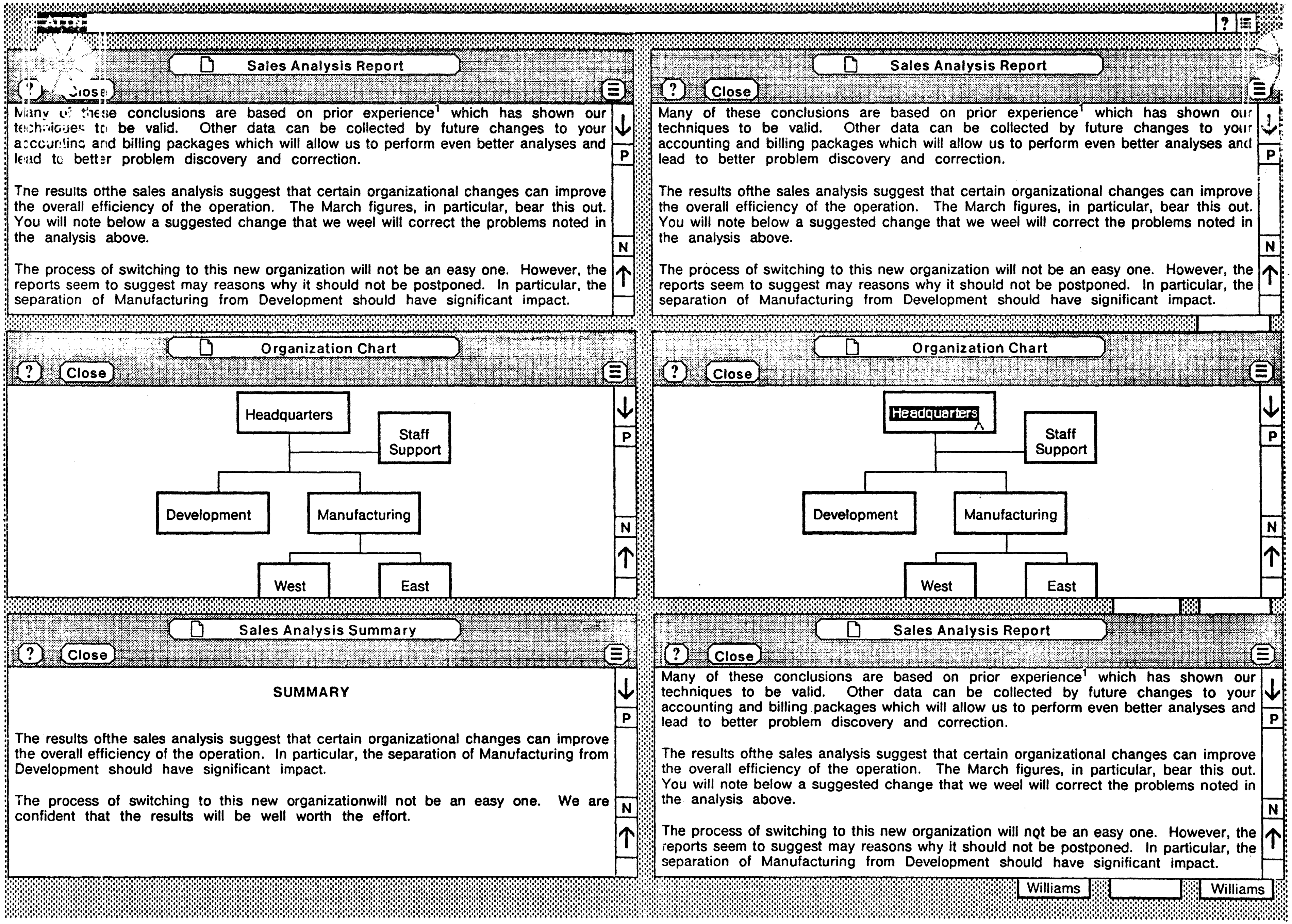


The results ofthe sales analysis suggest that certain organizational changes can improve the overall efficiency of the operation. The March figures, in particular, bear this out. You will note below a suggested change that we weel will correct the problems noted in the analysis above.

The process of switching to this new organization will not be an easy one. However, the reports seem to suggest may reasons why it should not be postponed. In particular, the separation of Manufacturing from Development should have significant impact.

Also, we feel strongly that merging East and West Coast Development will help. As we have suggested in past reports there has always been

Figure DTP-3 -- Positioning of two document windows open at same time.



Many of these conclusions are based on prior experience[1] which has shown our techniques to be valid. Other data can be collected by future changes to your accounting and billing packages which will allow us to perform even better analyses and lead to better problem discovery and correction.

The results ofthe sales analysis suggest that certain organizational changes can improve the overall efficiency of the operation. The March figures, in particular, bear this out. You will note below a suggested change that we weel will correct the problems noted in the analysis above.

The process of switching to this new organization will not be an easy one. However, the reports seem to suggest may reasons why it should not be postponed. In particular, the separation of Manufacturing from Development should have significant impact.

Many of these conclusions are based on prior experience[1] which has shown our techniques to be valid. Other data can be collected by future changes to your accounting and billing packages which will allow us to perform even better analyses and lead to better problem discovery and correction.

The results ofthe sales analysis suggest that certain organizational changes can improve the overall efficiency of the operation. The March figures, in particular, bear this out. You will note below a suggested change that we weel will correct the problems noted in the analysis above.

The process of switching to this new organization will not be an easy one. However, the reports seem to suggest may reasons why it should not be postponed. In particular, the separation of Manufacturing from Development should have significant impact.

SUMMARY

The results ofthe sales analysis suggest that certain organizational changes can improve the overall efficiency of the operation. In particular, the separation of Manufacturing from Development should have significant impact.

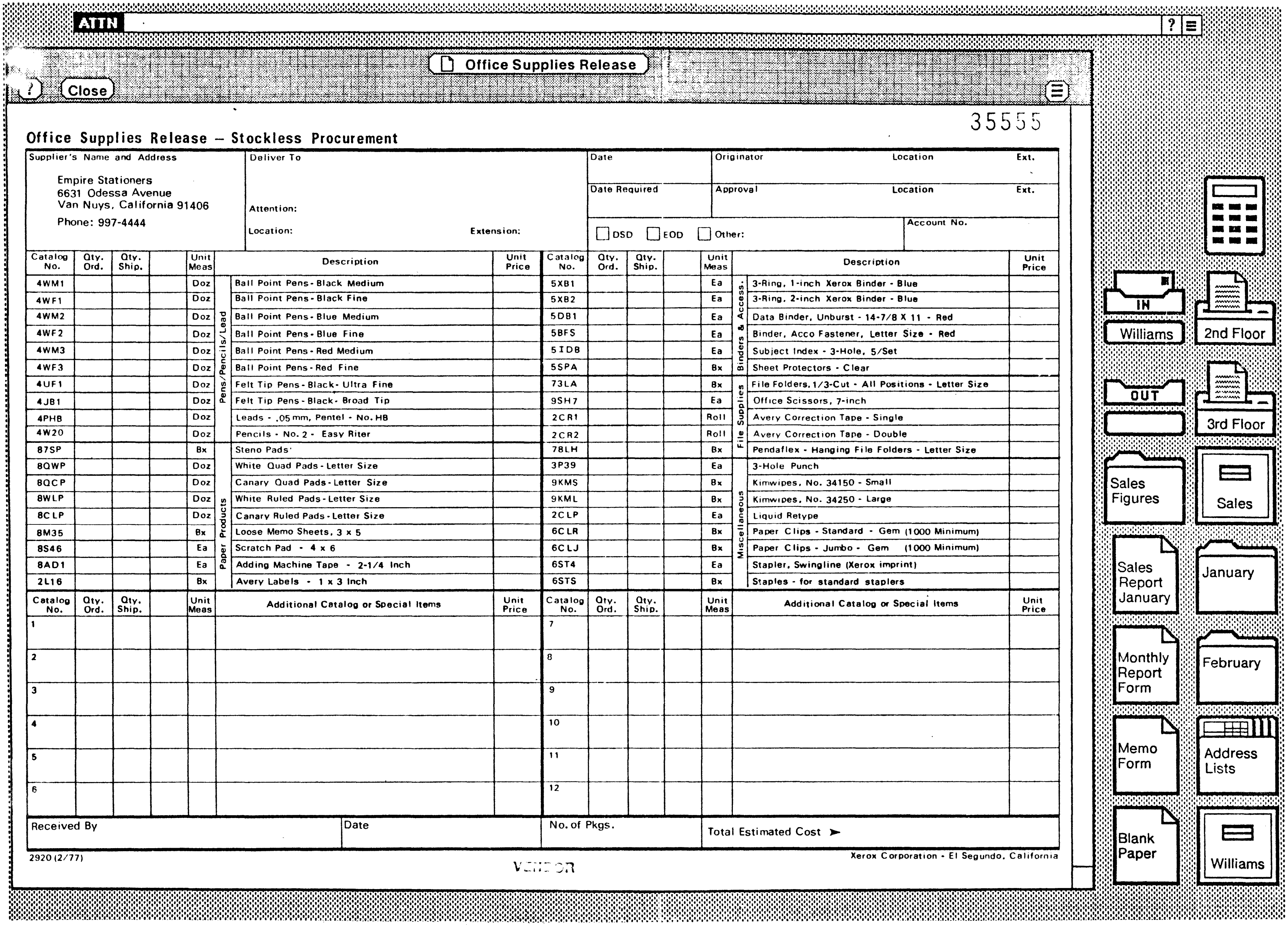
The process of switching to this new organizationwill not be an easy one. We are confident that the results will be well worth the effort.

Many of these conclusions are based on prior experience[1] which has shown our techniques to be valid. Other data can be collected by future changes to your accounting and billing packages which will allow us to perform even better analyses and lead to better problem discovery and correction.

The results ofthe sales analysis suggest that certain organizational changes can improve the overall efficiency of the operation. The March figures, in particular, bear this out. You will note below a suggested change that we weel will correct the problems noted in the analysis above.

The process of switching to this new organization will not be an easy one. However, the reports seem to suggest may reasons why it should not be postponed. In particular, the separation of Manufacturing from Development should have significant impact.

Figure DTP-4 -- Desktop with six document windows open.



35555

Office Supplies Release – Stockless Procurement

Supplier's Name and Address

Empire Stationers
6631 Odessa Avenue
Van Nuys, California 91406
Phone: 997-4444

Deliver To

Attention:

Location: Extension:

Date

Originator Location Ext.

Date Required

Approval Location Ext.

□ DSD □ EOD □ Other:

Account No.

Catalog No.	Qty. Ord.	Qty. Ship.		Unit Meas		Description	Unit Price	Catalog No.	Qty. Ord.	Qty. Ship.		Unit Meas		Description	Unit Price
4WM1				Doz	Pens/Pencils/Lead	Ball Point Pens - Black Medium		5XB1				Ea	Binders & Access.	3-Ring, 1-inch Xerox Binder - Blue	
4WF1				Doz		Ball Point Pens - Black Fine		5XB2				Ea		3-Ring, 2-inch Xerox Binder - Blue	
4WM2				Doz		Ball Point Pens - Blue Medium		5DB1				Ea		Data Binder, Unburst - 14-7/8 X 11 - Red	
4WF2				Doz		Ball Point Pens - Blue Fine		5BFS				Ea		Binder, Acco Fastener, Letter Size - Red	
4WM3				Doz		Ball Point Pens - Red Medium		5IDB				Ea		Subject Index - 3-Hole, 5/Set	
4WF3				Doz		Ball Point Pens - Red Fine		5SPA				Bx		Sheet Protectors - Clear	
4UF1				Doz		Felt Tip Pens - Black - Ultra Fine		73LA				Bx	File Supplies	File Folders, 1/3-Cut - All Positions - Letter Size	
4JB1				Doz		Felt Tip Pens - Black - Broad Tip		9SH7				Ea		Office Scissors, 7-inch	
4PHB				Doz		Leads - .05 mm, Pentel - No. HB		2CR1				Roll		Avery Correction Tape - Single	
4W20				Doz		Pencils - No. 2 - Easy Riter		2CR2				Roll		Avery Correction Tape - Double	
87SP				Bx	Paper Products	Steno Pads'		78LH				Bx		Pendaflex - Hanging File Folders - Letter Size	
8QWP				Doz		White Quad Pads - Letter Size		3P39				Ea	Miscellaneous	3-Hole Punch	
8QCP				Doz		Canary Quad Pads - Letter Size		9KMS				Bx		Kimwipes, No. 34150 - Small	
8WLP				Doz		White Ruled Pads - Letter Size		9KML				Bx		Kimwipes, No. 34250 - Large	
8CLP				Doz		Canary Ruled Pads - Letter Size		2CLP				Ea		Liquid Retype	
8M35				Bx		Loose Memo Sheets, 3 x 5		6CLR				Bx		Paper Clips - Standard - Gem (1000 Minimum)	
8S46				Ea		Scratch Pad - 4 x 6		6CLJ				Bx		Paper Clips - Jumbo - Gem (1000 Minimum)	
8AD1				Ea		Adding Machine Tape - 2-1/4 Inch		6ST4				Ea		Stapler, Swingline (Xerox imprint)	
2L16				Bx		Avery Labels - 1 x 3 Inch		6STS				Bx		Staples - for standard staplers	

Catalog No.	Qty. Ord.	Qty. Ship.		Unit Meas	Additional Catalog or Special Items	Unit Price	Catalog No.	Qty. Ord.	Qty. Ship.		Unit Meas	Additional Catalog or Special Items	Unit Price
1							7						
2							8						
3							9						
4							10						
5							11						
6							12						

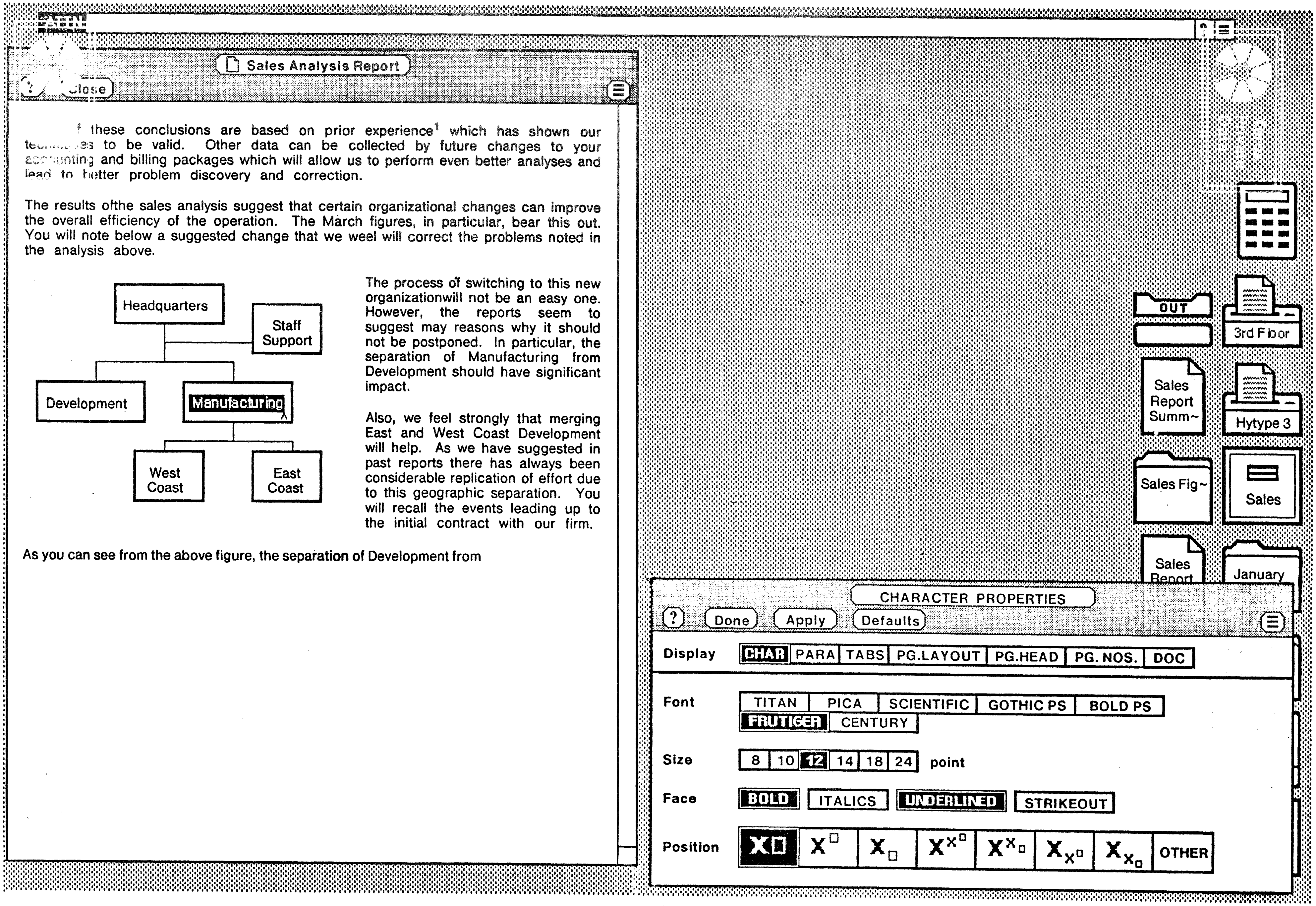
Received By	Date	No. of Pkgs.	Total Estimated Cost ➤

2920 (2/77)

Xerox Corporation - El Segundo, California

VENDOR

Figure DTP-5 -- Document window for landscape page



[illegible] f these conclusions are based on prior experience[1] which has shown our techniques to be valid. Other data can be collected by future changes to your accounting and billing packages which will allow us to perform even better analyses and lead to better problem discovery and correction.

The results ofthe sales analysis suggest that certain organizational changes can improve the overall efficiency of the operation. The March figures, in particular, bear this out. You will note below a suggested change that we weel will correct the problems noted in the analysis above.

The process of switching to this new organizationwill not be an easy one. However, the reports seem to suggest may reasons why it should not be postponed. In particular, the separation of Manufacturing from Development should have significant impact.

Also, we feel strongly that merging East and West Coast Development will help. As we have suggested in past reports there has always been considerable replication of effort due to this geographic separation. You will recall the events leading up to the initial contract with our firm.

As you can see from the above figure, the separation of Development from

CHARACTER PROPERTIES

? Done Apply Defaults

Display: **CHAR** PARA TABS PG.LAYOUT PG.HEAD PG. NOS. DOC

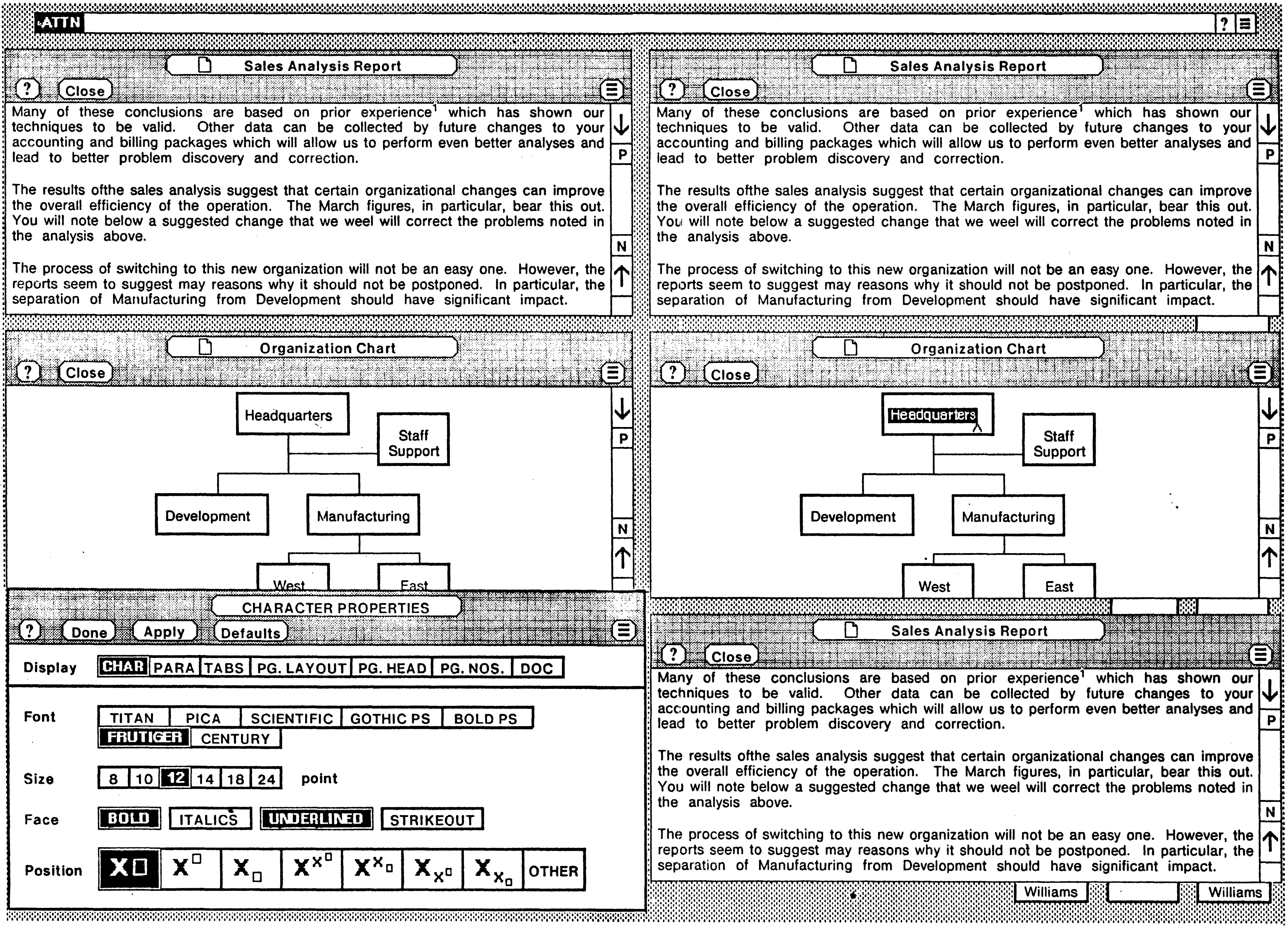
Font: TITAN PICA SCIENTIFIC GOTHIC PS BOLD PS **FRUTIGER** CENTURY

Size: 8 10 **12** 14 18 24 point

Face: **BOLD** ITALICS **UNDERLINED** STRIKEOUT

Position: **X□** $X^{□}$ $X_{□}$ $X^{X^{□}}$ $X^{X_{□}}$ $X_{X^{□}}$ $X_{X_{□}}$ OTHER

Figure DTP-6 -- Property sheet positioned on right when selection on left



Many of these conclusions are based on prior experience[1] which has shown our techniques to be valid. Other data can be collected by future changes to your accounting and billing packages which will allow us to perform even better analyses and lead to better problem discovery and correction.

The results ofthe sales analysis suggest that certain organizational changes can improve the overall efficiency of the operation. The March figures, in particular, bear this out. You will note below a suggested change that we weel will correct the problems noted in the analysis above.

The process of switching to this new organization will not be an easy one. However, the reports seem to suggest may reasons why it should not be postponed. In particular, the separation of Manufacturing from Development should have significant impact.

Many of these conclusions are based on prior experience[1] which has shown our techniques to be valid. Other data can be collected by future changes to your accounting and billing packages which will allow us to perform even better analyses and lead to better problem discovery and correction.

The results ofthe sales analysis suggest that certain organizational changes can improve the overall efficiency of the operation. The March figures, in particular, bear this out. You will note below a suggested change that we weel will correct the problems noted in the analysis above.

The process of switching to this new organization will not be an easy one. However, the reports seem to suggest may reasons why it should not be postponed. In particular, the separation of Manufacturing from Development should have significant impact.

Many of these conclusions are based on prior experience[1] which has shown our techniques to be valid. Other data can be collected by future changes to your accounting and billing packages which will allow us to perform even better analyses and lead to better problem discovery and correction.

The results ofthe sales analysis suggest that certain organizational changes can improve the overall efficiency of the operation. The March figures, in particular, bear this out. You will note below a suggested change that we weel will correct the problems noted in the analysis above.

The process of switching to this new organization will not be an easy one. However, the reports seem to suggest may reasons why it should not be postponed. In particular, the separation of Manufacturing from Development should have significant impact.

CHARACTER PROPERTIES

? Done Apply Defaults

Display [✓ CHAR] PARA TABS PG. LAYOUT PG. HEAD PG. NOS. DOC

Font TITAN PICA SCIENTIFIC GOTHIC PS BOLD PS [✓ FRUTIGER] CENTURY

Size 8 10 [✓ 12] 14 18 24 point

Face [✓ BOLD] ITALICS [✓ UNDERLINED] STRIKEOUT

Position [✓ X□] $X^{\square}$ $X_{\square}$ $X^{X^{\square}}$ $X^{X_{\square}}$ $X_{X^{\square}}$ $X_{X_{\square}}$ OTHER

Figure DTP-7 -- Property sheet displayed on left when selection on right

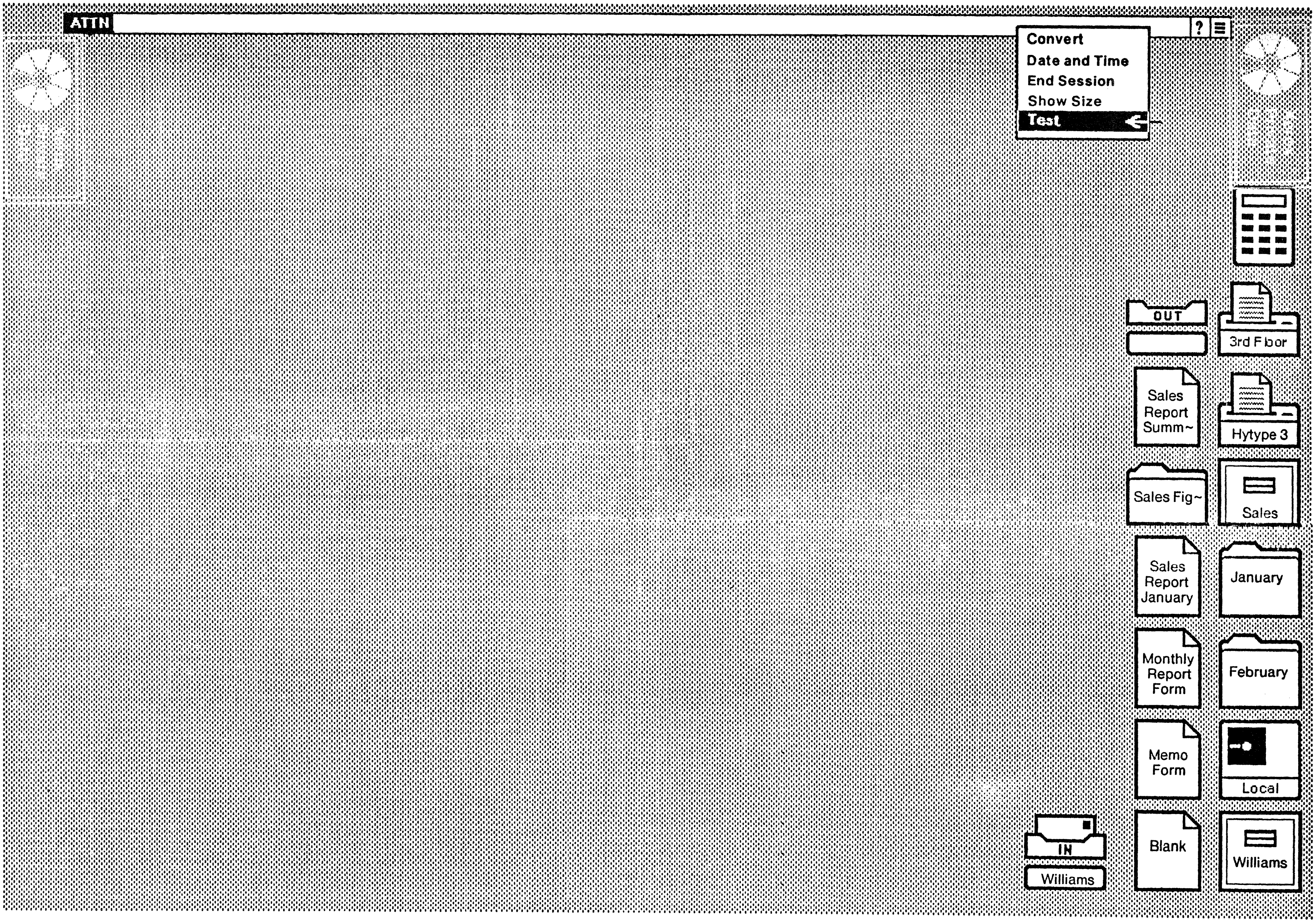
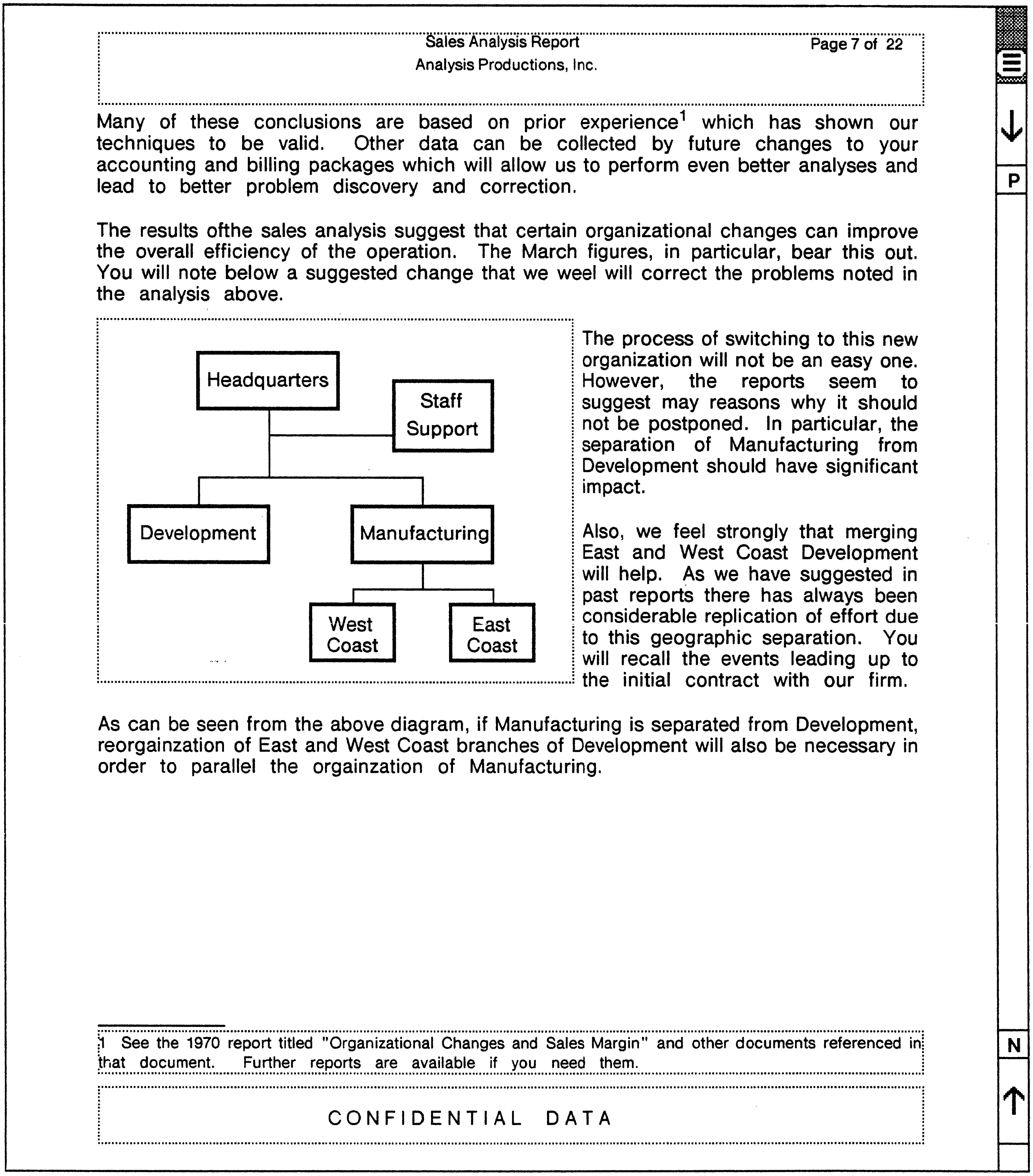


Figure DTP-8 -- Invoking a Desktop Transient Menu Command



Sales Analysis Report

Page 7 of 22

Analysis Productions, Inc.

Many of these conclusions are based on prior experience[1] which has shown our techniques to be valid. Other data can be collected by future changes to your accounting and billing packages which will allow us to perform even better analyses and lead to better problem discovery and correction.

The results ofthe sales analysis suggest that certain organizational changes can improve the overall efficiency of the operation. The March figures, in particular, bear this out. You will note below a suggested change that we weel will correct the problems noted in the analysis above.

The process of switching to this new organization will not be an easy one. However, the reports seem to suggest may reasons why it should not be postponed. In particular, the separation of Manufacturing from Development should have significant impact.

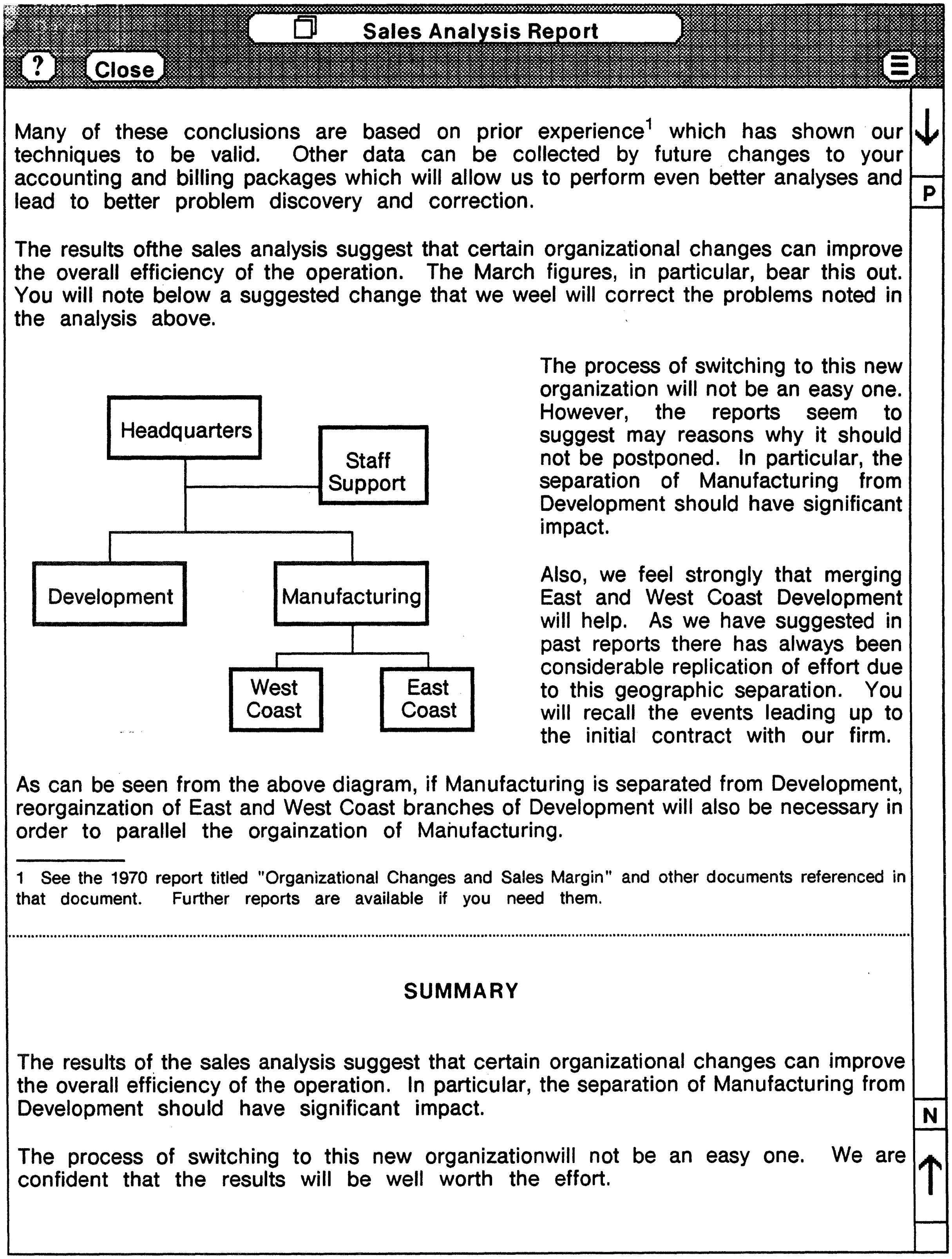
Also, we feel strongly that merging East and West Coast Development will help. As we have suggested in past reports there has always been considerable replication of effort due to this geographic separation. You will recall the events leading up to the initial contract with our firm.

As can be seen from the above diagram, if Manufacturing is separated from Development, reorgainzation of East and West Coast branches of Development will also be necessary in order to parallel the orgainzation of Manufacturing.

[1] See the 1970 report titled "Organizational Changes and Sales Margin" and other documents referenced in that document. Further reports are available if you need them.

CONFIDENTIAL DATA

Figure DOC-1 -- Document window with Show BOUNDARIES and MARGINS set.



Many of these conclusions are based on prior experience[1] which has shown our techniques to be valid. Other data can be collected by future changes to your accounting and billing packages which will allow us to perform even better analyses and lead to better problem discovery and correction.

The results ofthe sales analysis suggest that certain organizational changes can improve the overall efficiency of the operation. The March figures, in particular, bear this out. You will note below a suggested change that we weel will correct the problems noted in the analysis above.

The process of switching to this new organization will not be an easy one. However, the reports seem to suggest may reasons why it should not be postponed. In particular, the separation of Manufacturing from Development should have significant impact.

Also, we feel strongly that merging East and West Coast Development will help. As we have suggested in past reports there has always been considerable replication of effort due to this geographic separation. You will recall the events leading up to the initial contract with our firm.

As can be seen from the above diagram, if Manufacturing is separated from Development, reorgainzation of East and West Coast branches of Development will also be necessary in order to parallel the orgainzation of Manufacturing.

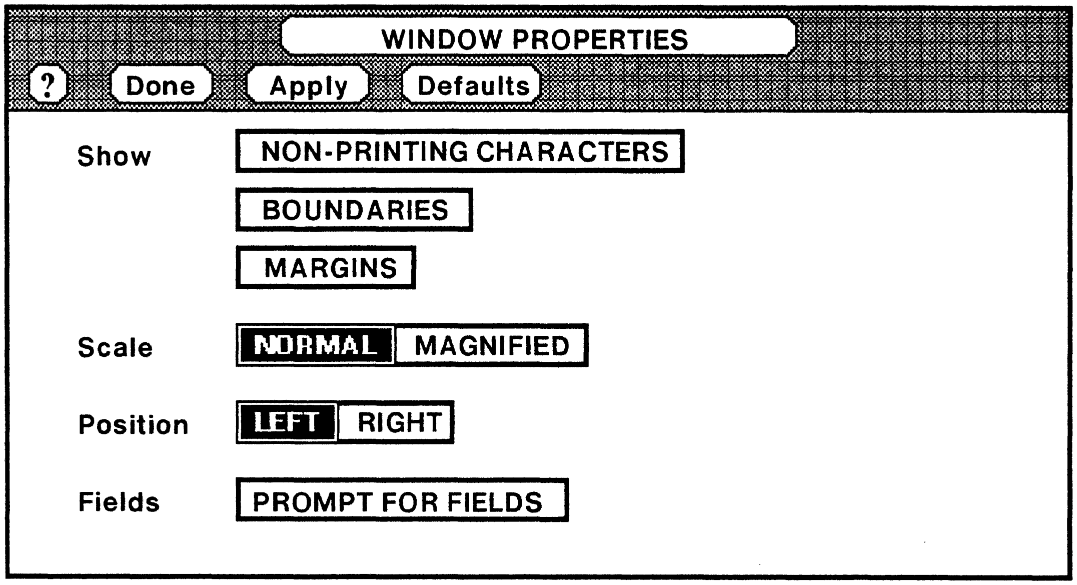
[1] See the 1970 report titled "Organizational Changes and Sales Margin" and other documents referenced in that document. Further reports are available if you need them.

SUMMARY

The results of the sales analysis suggest that certain organizational changes can improve the overall efficiency of the operation. In particular, the separation of Manufacturing from Development should have significant impact.

The process of switching to this new organizationwill not be an easy one. We are confident that the results will be well worth the effort.

Figure DOC-2 -- Document window showing page break.



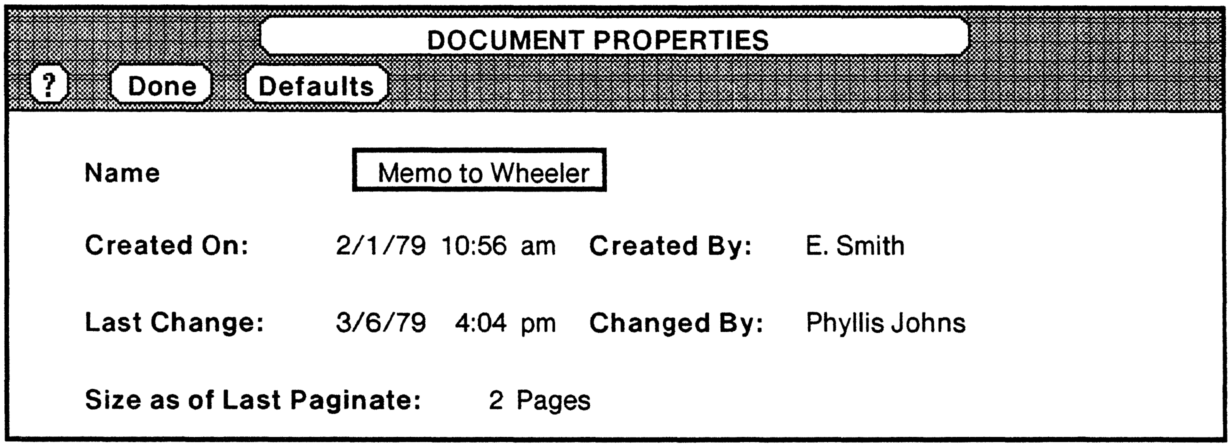
Show: NON-PRINTING CHARACTERS; BOUNDARIES; MARGINS

Scale: [✓ NORMAL] MAGNIFIED

Position: [✓ LEFT] RIGHT

Fields: PROMPT FOR FIELDS

Figure DOC-3 -- Window Property Sheet



Name	Memo to Wheeler		
Created On:	2/1/79 10:56 am	Created By:	E. Smith
Last Change:	3/6/79 4:04 pm	Changed By:	Phyllis Johns

Size as of Last Paginate: 2 Pages

Figure DOC-4 -- Document Property Sheet

(THIS PAGE INTENTIONALLY BLANK)

8. TEXT EDITING

The features described in this section allow the user to enter and edit text characters. These operations apply to all text within documents, as well as to any other text the user needs to supply or modify, such as a document name. This section is closely related to *Formatting and Layout*. While this section describes operations to create and modify text characters, *Formatting and Layout* describes how appearance properties are assigned and modified, thus determining how the text is rendered on the display or in hardcopy form.

USER OVERVIEW

Star provides a single set of text editing functions. These apply whenever the user needs to enter or modify text, primarily the contents of documents. This approach provides a consistent, easy-to-learn interface to the system.

All text editing operations are based on a *selection* composed of a *sequence of characters*. This selection is the operand of editing commands. When the user enters text from the keyboard, it is placed immediately after the selection. Initially the selection itself possesses an insertion caret at its right end. As soon as the user begins typing, the selection disappears, and the caret remains to mark the place where insertion occurs. Entering text is the default editing operation. It does not require an explicit user command.

Because the selection is integral to editing operations, the system provides optimizations for selecting words, sentences and paragraphs, as well as characters. The length of the selection ranges from a single character up to the entire contents of the document.

Commands are provided to move, copy or delete a character sequence, or to expand a character sequence to a predefined, arbitrary sequence of characters. Other commands search for a specific string or make global substitutions of one string for another. In addition, the user may include all or part of other documents (e.g. boilerplate) into the document being edited.

DETAILED SPECIFICATION

OBJECTS

Text Character

A *text character* is a letter, digit, symbol or special character that can be entered from the keyboard. Text characters are the basic elements used to form the entities that comprise documents. They have many special appearance properties associated with them. These properties are described in *Formatting and Layout*.

The standard American Star keyboard provides the following text characters, as indicated on the keycap labels.

Letters

abcdefghijklmnopqrstuvwxyz
ABCDEFGHIJKLMNOPQRSTUVWXYZ

Digits

0123456789

Symbols

@ # $ % <cent sign> & * + = / ( ) [ ] { } < > “ ” ‘ ’ . , ; : ? !
- (hyphen, both breaking and non-breaking)

The fonts that are to be supported in Star-1 include *Fruitiger* and *Century Schoolbook* for ROS printers and a TBD set of fonts available on impact print wheels.

The *Keyboard Sets* addendum to this section describes how additional character sets, fonts and functions may be accessed via the KEYBOARD key. This facility permits the total set of characters available in Star to be expanded indefinitely.

All "character" keys on the Star keyboard repeat if held down. Most function keys do not repeat. Exceptions are NEW LINE, NEW PARAGRAPH, BACKSPACE, BACKWORD and NEXT.

Special Character

A *special character* is a character other than the ones above which can be entered from the keyboard. Special characters are non-printing (except hyphens) and have various formatting properties. Many of them are entered via the KEYBOARD key as described later in this section. Except as noted, they are not displayed unless they are selected or unless Show Non-Printing Characters is on in the in the Set Window option sheet (see Figure DOC-3). Figure TXT-1 shows how special characters are displayed.

Space - The space character is used to delimit words. It is normally displayed as a horizontal area of white space between text characters. It may be variable in width. It is not displayed when selected.

Required Space - This character is similar to space, and is normally displayed as a horizontal area of white space. It is treated like a letter and does not delimit words (for selection and line breaks). It does not vary in width but is fixed at the width of a normal (non-justified) blank. It is entered via the KEYBOARD key. It is not displayed when selected.

Tab - The tab character is used to position text at a fixed horizontal location. It is normally displayed as a variable-width, horizontal area of white space between text characters. It is not displayed when selected.

Overstrike - The overstrike character is a non-erasing backspace, which allows two or more characters to occupy the same position. It is entered via the KEYBOARD key. The next character typed is placed in the same horizontal position as the character before the overstrike. Multiple characters can be placed in the same location with successive overstrikes. However, overstrike affects only the previous character, and entering overstrike several times in succession is the same as doing it once. The two or more characters that are overstuck are selected as a single character unit.

For proportionately spaced (or differing) fonts, the first character is placed in the line of text. Subsequent "overstruck" characters are placed on top, such that their right edges coincide.

(Note: Overstrike is not used for accents, in general. For European installations special keyboards will be provided that have some keys which act as "dead keys". Dead keys exist on European typewriters today. They type an accent mark but do not advance the character position. Star's dead keys will have the same effect from the user's point of view. Should users of American Star need to type an accent, they will either have to use overstrike or switch to a different keyboard interpretation via the KEYBOARD key (see *Keyboard Sets*).

New Line (NL) (Carrier Return) - Normally text is displayed across a line until the right margin is reached and automatically continues on the next line at a word boundary. This character forces a new line when text is rendered. When displayed, it is at the left end of the line it begins. New line is entered with the NEW LINE key.

New Paragraph (NP) - This character is used to start a new paragraph of text. Paragraphs are significant as editing operands and as units with special properties. A new paragraph character is entered with the NEW PARAGRAPH key. When entered, it assumes the properties of the preceding paragraph. When displayed it appears just to the left of the first normal character in the paragraph.

Page Format Character (PFC) - This character carries page formatting information such as absolute margins, headings, etc. The formatting information in the page format character affects the rendering of every page that follows the character. The page format character is entered via the KEYBOARD key. A Page Format character is always accomanied by a New Paragraph character. When a Page Format character is displayed, it appears just to the left of its paragraph's first normal character.

Start-of-Document (SOD) - This character is the first character in every document. It is provided automatically by the system. There is only one per document. It cannot be entered or deleted by the user, and is never printed on hardcopy. If the user attempts to delete it, an error message is displayed. The SOD is followed by a Page Format character and a New Paragraph character. It may also be assigned text properties. Thus, even an empty document contains complete formatting information, so that text may be added without specifying this information.

Stop Character - This character causes printing on a character printer to pause for operator action. It is used for special purposes defined by the user. It is not necessary for font changes; character printers will automatically stop in those cases. It is entered via the KEYBOARD key.

Hyphen - A normal hyphen has a number of interpretations: as a character in a word (e.g. "intra-document"), as a minus sign ("x-3"), or (usually in pairs) as a dash. A single hyphen does not delimit words. When rendering text, however, a single hyphen, surrounded by letters, is used as a temporary word break at the end of a line. In this case the hyphen remains with the first part of the word. It is entered with the hyphen key.

Required Hyphen - This hyphen is like a normal hyphen but cannot be used as a temporary word break by the rendering algorithm (e.g. "x-3"). It is entered with the upper-case hyphen key.

Discretionary Hyphen - This hyphen is entered by the user to indicate potential points of hyphenation, normally for long words. The rendering algorithm uses these hyphens to assist in aligning and justifying lines of text. It is entered via the KEYBOARD key. It is never printed unless it is used to break a line, when it is printed as a hyphen.

Word

A *word* is a convenient aggregation of text characters. The definition of "word" is given in *Keyboard Sets,* following. Special facilities are provided for selecting words.

Sentence

A *sentence* is a sequence of words and symbols bounded by sentence separators. A sentence separator is:

1) any Star character which is a period, question mark or exclamation point, when it is followed by:
    a) one or more spaces,
    b) any Star character which is a single or double neutral or right quote,
    c) any Star character which is a right parenthesis or other right bracket,
    d) any combination of a, b and c;

2) a Japanese period followed by:
    a) no spaces,
    b) any combination of single or double neutral or right quotes, right parentheses or other right brackets, without spaces;

3) a new line or new paragraph character;

4) the start or end of the frame, for text inside a frame;

5) the start or end of the document.

If a sentence ends with a separator of type 1, then it includes the trailing separator. Special facilities are provided for selecting sentences.

Paragraph

A *paragraph* is a sequence of text characters, aggregated as words and sentences. A paragraph is bounded by new paragraph characters or their equivalents (Start-of-Document and Page Format); a document's last paragraph is bounded by the end of that document. Paragraphs provide a convenient aggregation of words and sentences. Paragraphs have many special properties associated with them, which affect their rendering. Special facilities are provided for selecting paragraphs.

Text Selection

A *text selection* is a sequence of text and special characters that serves as the operand for text editing operations. The selection is normally displayed in reverse video, together with a caret at its right end.

Caret

A *caret* is always shown at the right end of every text selection. A caret may also exist by itself during text entry from the keyboard, or immediately following a Delete. There is never more than one caret present at a time. The caret is repositioned when the user makes a new selection. The caret has alternate shapes, as depicted in Figure TXT-1, to indicate some text formatting properties for subsequently typed text.

ACTIONS

Make a Selection

A selection is necessary for most text editing operations. The selection may be made in units of characters, words, sentences, or paragraphs, as described below. In all cases, when a selection is made, it is highlighted, and any previous selection is deselected.

Select Character - To select a single character, the user points (with the mouse cursor) at the desired character on the display and presses and releases the SELECT mouse button. If SELECT is held down and the cursor moved, the selection changes from character to character.

Select Word - To select a single word, the user points to any character in the desired word and presses and releases the SELECT mouse button twice. (This is called "double clicking.") The second click must occur within the character that was selected by the first click. The word selected will include an adjacent space, subject to the following rule:

> Category 1: If the word is textual (a sequence of letters, digits, hyphens, required spaces, overstrikes, and dead deys), and has a *trailing* space, the space is selected along with the word.
>
> Category 2: If the word is textual, and has no trailing space, but has a *leading* space, the space is selected along with the word.
>
> Category 3: If the word is textual, and has no trailing or leading space, then only the word itself is selected.
>
> Category 4: If the word is non-textual (e.g. several spaces in a row), then only the word itself is selected.

Select Sentence - To select a single sentence, the user points to any character in the desired sentence and presses and releases the SELECT mouse button three times. Each successive click must be within the existing selection. After each click, the highlighted entity expands, from a character to a word, to the sentence.

The sequence of words comprising the sentence will be selected, together with the trailing punctuation mark and the adjacent spacing, subject to the following rule:

> Category 1: If the punctuation mark at the end of the sentence is follwed by one or more spaces, the spaces are selected along with the sentence.
>
> Category 2: If the punctuation mark at the end of the sentence is not followed by a space, and the first character of the sentence is preceded by one or more spaces, then these preceding spaces are selected along with the sentence.
>
> Category 3: If the sentence has no spaces surrounding it, then just the sentence is selected.

(Note: The definition of sentence above is not intended to handle all possible sentences, but to provide a means of selecting typical sentences. If the sentence selected by the system is not what the user desires, he must select it as a sequence of characters or words.)

Select Paragraph - To select a single paragraph, the user points to any character in the desired paragraph and presses and releases the SELECT mouse button four times. Each successive click must be within the existing selection.

Select Everything - If the user multiply clicks SELECT five times, the contents of the entire document is selected, and the message "Everything in the document is selected" is displayed. If the user clicks SELECT a sixth time, the selection returns to the character level.

Adjust Selection

A selection can be adjusted to incorporate more or fewer characters. After making a selection, the user points to where he wants the new endpoint to appear, and presses and releases the ADJUST button. The nearer end of the selection adjusts to the location of the cursor when the ADJUST button is depressed. The user may, if desired, hold the ADJUST button down while adjusting, and move the cursor from one place to another. While the ADJUST button is down, the selection will change to reflect the location of the cursor.

When adjusting a selection (other than "everything"), the endpoint moves in the same units that were used to make the original selection. Thus, if the user is adjusting a word selection (one made by double clicking), the endpoint will always jump to a word boundary. When adjusting a sentence selection, the endpoint will always jump to a sentence boundary.

If the user desires to adjust a word, sentence, or paragraph selection in terms of single characters rather than larger units, he may press the ADJUST button over either end character of the selection, move to the desired character, then release the button.

Specific details of adjusting depend on the nature of the selection, as follows.

Character selection - To select a sequence of characters, the user first selects a single character, then moves the mouse cursor to some other character (before or after the first character selected) and presses and releases the ADJUST button.

Word selection - To select a sequence of words, the user first selects a single word, then moves the mouse cursor to some other word and presses and releases the ADJUST button. The selection as a whole includes its adjacent space, as described above. A contiguous sequence of the *same* symbol or special character is treated as a word for selection purposes (it can be selected by double clicking SELECT), even though it is not a "word" in the English language. For example, three periods in a row or two spaces in a row would be treated as a word.

Sentence selection - To select a sequence of sentences, the user first selects a single sentence, then moves the mouse cursor to some other sentence and presses and releases the ADJUST button. The extended sentence selection as a whole includes surrounding spaces according to the above rule.

Paragraph selection - To select a sequence of paragraphs, the user first selects a single paragraph, then moves the mouse cursor to some other paragraph and presses and releases the ADJUST button.

Enter Text Characters

To add text to a document, the user merely types the characters on the keyboard. If a selection is present, the first character typed is inserted immediately after the selection but in front of the caret. The selection highlighting disappears. (If there is no text selection or caret, an error message is displayed, and any typed-in characters are discarded.) When the user finishes entering characters at a given point, he makes a selection somewhere else in the document, and resumes typing.

A special exception is made for entering text in front of the first character of a line or in front of the first character in a (non-empty) field. When the user places the cursor in the left half of the first visible character of a line or form field or in unoccupied margin space to the left of the first character in an ordinary document and clicks SELECT, the system places an insertion caret to the left of the character.

A special optimization is provided for selecting a blank field within a form: whenever a blank field exists within a *locked* document, the user may insert text within the field by first clicking SELECT anywhere within the frame. This action causes a caret to be placed within the field. It is not necessary for the special field characters to be visible to invoke this function.

As a rule, the default properties of a character entered from the keyboard are obtained from the character immediately to its left. An exception is that after a Delete the caret has the properties of the first character in the deleted selection; this permits the *replace* operation to be simply a Delete followed by an Insert. Each character assumes the appearance properties of the caret as it is entered. Some of these settings are reflected in the shape of the caret. At any point, the user may change the properties of the caret. (Properties may also be changed after the characters are entered.)

The user types continuously, without regard to line breaks. Star automatically starts a new line on a word boundary when the right margin is reached (automatic carrier return). If the user enters a word longer than the width of the paragraph, the system automatically breaks the "word," but does not hyphenate it; such lines may not be properly justified.

The system continuously reformats characters as the user enters text. Thus, at any point, the user sees all the effects of his changes. Line breaks are reformatted, and all paragraph properties are applied. (If the user types ahead of the system, reformatting may happen in bursts, since the system will process typed-ahead characters before reformatting.)

Unusual Conditions - If the caret is out of the window and the user starts typing, the document automatically scrolls to show the caret. When this happens, the page containing the caret is displayed, normally, with the top of that page at the top of the document window. In cases where the top of the page and the caret cannot be viewed simultaneously, the displayed page is positioned so that the caret appears 25% of the way down the window. [This is referred to as the Positioning Rule in subsequent paragraphs.]

If, in the process of entering characters, the user fills the last line in the window, the contents of the window automatically scroll upward. The document is positioned so that the line containing the caret remains at the bottom of the window. For pages wider than the window, if the insertion caret reaches the right edge, the contents are automatically scrolled, horizontally to the left, one inch.

When entering text at a centered, right-flush or decimal aligned tab, characters propagate to the left. If they expand to the point that they overlay preceding characters, the system automatically starts a new line with the preceding TAB character.

Enter Character with the KEYBOARD Key

Less frequently used characters are entered by means of the KEYBOARD key. By pressing KEYBOARD, the user can temporarily change the meaning of the keys on the keyboard, thus gaining access to additional character sets. Special objects such as frames are also entered in this manner. These objects are placed in the document at the location of the caret, the same as text characters.

When KEYBOARD is pressed, The window header is overlaid with a pictorial representation of the property keys located in the top row of the keyboard (see Figure TXT-2a). However, instead of their normal labels, the displayed keys show the keyboard set alternatives that may be chosen, either by pressing the corresponding physical property key, or by selecting the displayed key with the mouse.

As described further in *Keyboard Sets*, the user may reassign the entire keyboard to one of the other sets, either temporarily or (semi) permanently. Figure TXT-2b shows one form of "Special" keyboard set that may be employed in this way.

Insert Time and Date

The user may insert the time and/or date via the KEYBOARD key. They are in the same format described in *Desktop*. They assume the font and face properties of the caret. If the caret is in a non-Roman font, the standard font is used. A leading and/or trailing space will accompany the date/time under the same rules that govern move/copy word, described later, in "Move Text."

Delete Text

To delete a sequence of characters, the user selects them and presses the DELETE key. After a deletion, a caret remains at the position the deleted text had occupied. The caret assumes the properties of the first character of the deleted text. The display reflects correct line breaks and paragraph formatting.

The appearance properties of the remaining characters are unchanged. However, a long deletion may span several paragraphs and cause parts of two paragraphs to merge. In this event, since the new paragraph character holds the paragraph's properties, the second half (from the last paragraph) assumes the paragraph properties of the first paragraph.

If the selection was entirely out of the window when the DELETE key was pushed, a warning message is displayed. The document scrolls automatically according to the Positioning Rule. If the user has been entering characters and presses DELETE, no deletion takes place; the user must explicitly select the text to be deleted.

Backspace and Backword

Star provides an additional key to expedite deletion of characters and words. BACKSPACE (←) deletes the character preceding the caret. The displayed text performs exactly as with the DELETE key. BACKWORD (SHIFT ←) works like BACKSPACE except that the word (and trailing delimiters) preceding the caret are deleted.

If the user holds down the BACKSPACE (or BACKWORD) key, characters (words) are deleted continuously until either the key is released or until the caret reaches an object that may be difficult to reconstruct. When the repetition is suspended due to reaching such a barrier, lifting the key and repressing it will resume the repetition.

The barrier objects are New Paragraph characters, Page Format characters, Frames, and Equations.

Replace Text

There is no explicit replace command in Star. The user deletes the text he wishes to replace and types the replacement text.

Move Text

The user can move text from one point to another in the same document or another document. To do this he selects what he wants to move, depresses and releases the Move key, and indicates the destination with either mouse button, as described in *System Overview*. When the user depresses a mouse button to indicate the destination, the original selection is highlighted by a dotted underline, and the system response depends upon the type of the original selection:

> *Character Selection:* The destination is shown as a selected character and a caret (a caret by itself at left margins). As the cursor is moved around with the button held down, the destination character selection changes appropriately. When the button is released, the text to be moved is inserted at the position of the caret.
>
> *Word Selection:* The destination is shown as a selected word, with the standard trailing caret. If the destination word is textual, it does not include surrounding spaces. As the cursor is moved around with the button held down, the destination word selection changes appropriately. When the button is released, the text to be moved is inserted after the destination word according to the following rules:
>
>> 1) If the destination word is non-textual, the text to be moved is always inserted after the destination, with no other changes made.

2a) If the destination word is textual and the text to be moved is a Category 1 word selection, its trailing space is stripped off, and inserted between the destination word and the text to be moved.

2b) If the destination word is textual and the text to be moved is a Category 2 word selection, the text to be moved is simply inserted after the destination word.

2c) If the destination word is textual and the text to be moved is a Category 3 word selection, a space is created and placed between the destination word selection and the text to be moved.

2d) If the destination word is textual and the text to be moved is a Category 4 word selection, the text to be moved is simply inserted after the destination.

Sentence Selection: The destination is shown as a selected sentence, with the standard trailing caret. The destination sentence does not include surrounding spaces. As the cursor is moved around with the button held down, the destination sentence selection changes appropriately. When the button is released, the text to be moved is inserted after the destination sentence according to the following rules:

1) If the sentence to be moved is a Category 1 selection, its trailing spaces are stripped off, and inserted between the destination sentence and the sentence to be moved.

2) If the sentence to be moved is a Category 2 selection, the sentence to be moved is simply inserted after the destination sentence.

3) If the sentence to be moved is a Category 3 selection, two spaces are created and placed between the destination sentence and the sentence to be moved.

Paragraph Selection: The destination is shown as a selected paragraph, with the standard trailing caret. The highlighting includes the pre- and post-leading. As the cursor is moved around with the button held down, the destination paragraph changes appropriately. When the button is released, the text to be moved is inserted after the destination paragraph.

Everything Selection: The destination is shown as a selected paragraph, just as if a paragraph were being moved.

The appearance properties of all characters remain unchanged. However, because the moved text remains selected, it can easily be assigned the appearance properties of adjacent text with the Copy Properties command.

The moved text may create new paragraphs from parts of old paragraphs at both the source and destination. As with delete, the new paragraph character retains its paragraph properties and affects the rendering of all characters between it and the next new paragraph character.

If the user attempts to move a selection into itself, nothing happens.

Copy Text

The Copy command works exactly like Move except that the selected text is not deleted at the source. The command is invoked with the COPY key on the keyboard. Following the copy operation, the new copy of the text (at the destination) is selected.

Text may be copied inside itself.

If the user has been inserting characters and wants to replicate them, he can do so by making a new selection and pressing AGAIN.

Search

The Search command allows a user to make a selection by searching for sequences of characters and/or text properties. To invoke it, the user presses the FIND key. Star displays an option sheet containing search options (see Figure TXT-3). The parameters remain set from the last time FIND was pressed. By pressing the FIND key a second time the user can continue the previous search. Alternatively, the user can change any of the parameters on the option sheet before starting the search. The user can search for (1) a specific character sequence ignoring properties, (2) any character sequence with a specific set of properties, or (3) a specific character sequence with specific properties. Case is compared unless IGNORE CASE is on. If IGNORE CASE is on, upper and lower case are considered to be the same.

A user sets the In parameter to bound the search. If ENTIRE DOCUMENT is chosen, the search proceeds from the beginning to the end of the document. If REST OF DOCUMENT is chosen, the search proceeds from the current position of the caret forward to the end of the document. If CURRENT SELECTION is chosen, the search is only inside the current selection from beginning to end. The Search command examines *all text* in the specified range, including text inside frames.

Search for Characters only - If a user wishes to search for a string of characters, irrespective of properties, he chooses TEXT in the By Matching parameter and enters the text in the Search For parameter. He then invokes Start (or presses the FIND key again) and the search proceeds. In the process of searching, only characters are compared. All properties are ignored; "chair" and "***chair***" match. This is expected to be the normal search mode for most users.

Search for Properties only - If a user wishes to search for any string of characters having specific properties, he chooses PROPERTIES in the By Matching parameter. In Star-1 only character properties may be searched for; later releases may allow paragraph and page properties as well. The user enters in the Search For parameter any characters that have the properties he wishes to locate. When a property sheet is requested for this text, an additional state (indicated by graying the rectangle for state parameters) is added to each state and choice parameter to indicate a "Don't Care" condition for that property. If a property is either on or off, then Search will require the corresponding property to be on or off. If the Don't Care state is chosen, that property is ignored in the search. In the process of searching, the actual characters are ignored; only properties are compared.

The following rules apply:

> All state and choice parameters (in the Character property sheet in Star-1) must match unless set to Don't Care.
>
> All empty text parameters are ignored. All non-empty ones must match.

Search selects the entire string of contiguous characters that have the specified properties.

If all characters in the Search For parameter do not have the same properties, the properties of the first character are used, and a warning message is displayed.

Search for Text and Properties - If a user wishes to search for a specific string with specific properties, he chooses BOTH in the By Matching parameter. He enters the desired string in the Search For parameter with the appropriate properties. Search compares both the characters and properties as described above.

Operation - While Star is searching, the message "Searching..." is displayed. If a match is found, the text is selected and displayed according to the Positioning Rule. When completed (i.e. the entire range has been searched), Search restores the selection and display to their states before the Search began.

After a match has been found, a Search can be continued by hitting the AGAIN key or by reinvoking the Search command without changing any of the parameters. The Search will continue where it left off.

Stopping a Search - A user may terminate an ongoing Search by pressing the STOP key. The selection and display are restored to their states before the Search began.

Ellipsis Search - When searching for text or text and properties, the Search For parameter can include a single "matches anything" character. This ellipsis character is entered by means of the KEYBOARD key and is displayed as "...". Thus, the string "abc...def" means any sequence starting with "abc" followed by any string (possibly null) followed by "def".

An ellipsis search can match strings that cross paragraph boundaries. The matching process is restarted each time the characters preceding the ellipsis are found. For example, given the Search For parameter "abc...def", only "abcdef" would be selected in the string "abcabcdef". If the ellipsis is the first or last character in the Search For parameter, it matches a single character.

Unusual Conditions - The Search command operates in the window containing the selection or caret. If there is no selection or caret, an error message is displayed when FIND is pressed. Similarly, if the user chooses CURRENT SELECTION and there is only a caret, an error message is displayed when Start is invoked or when FIND is pressed for the second time.

Substitute

Substitute is used to change a number of occurrences of a sequence of characters and/or properties to a different sequence of characters and/or properties. This command is invoked with the FIND key, as an alternate to the Search command. The user specifies what to search for, as described for the Search command, then turns on the CHANGE IT option to indicate that anything found is to be changed. This causes three additional options to be displayed (see Figure TXT-4) which control the substitution to be made. The three options are Change To, By Altering, and CONFIRM EACH CHANGE. Change To contains the replacement text. By Altering controls what is replaced: TEXT, PROPERTIES or BOTH. CONFIRM EACH CHANGE gives the user the option of confirming or rejecting each replacement. As with Search, Substitute examines *all text* in the specified range, including text inside frames.

The following descriptions assume that the user has already specified the Search For, By Matching, and In parameters as he would for a normal Search, and has also turned on CHANGE IT.

> **Substitute Characters only** -- If the user wishes to substitute one string of characters for another, irrespective of properties, he chooses TEXT in the By Altering parameter. He fills in the Change To parameter with the replacement text and invokes Start (or presses the FIND key again). The system searches for the Search For text as described above. Subject to confirmation, described below, all matching text in the specified range is deleted and replaced by the Change To contents. The replacement text takes on the properties of the first character of the replaced text, as though the user had deleted the old text and typed the new. If IGNORE CASE is on and the first character of the replaced text was upper case, the first character of the replacement text is also set to upper case.
>
> **Substitute Properties only** - If the user wishes to substitute one set of properties for another set of properties on *any* text, he chooses PROPERTIES in the By Altering parameter. In Star-1 only character properties may be altered; later releases may allow paragraph and page properties as well. The user enters in the Change To parameter any characters that have the properties he wishes to assign. When a property sheet is requested for this text, an additional state (indicated by graying the rectangle for state parameters) is added to each state and choice parameter to indicate a "Don't Care" condition for that property. If a property is either on or off, then Substitute will set the corresponding property to be on or off in the text found. If the Don't Care state is chosen, that property is not changed.

The process of searching proceeds exactly as described for Search. When a match is found, all properties from Change To that are not set to "Don't Care" replace the properties in the string found. Substitute changes the entire string of contiguous characters that have the specified properties.

If all characters in the Change To parameter do not have the same properties, the properties of the first character are used and a warning message is displayed. (Recall that this is also true of Search For.)

Substitute Text and Properties - If a user wishes to substitute a string of text with one set of properties for another string with a different set of properties, he chooses BOTH in the By Altering parameter. He enters the desired string in the Change To parameter with the appropriate properties. Substitute searches for both the characters and properties in Search For as described above. When a match is found, the matched string is replaced by Change To. All properties in Change To that are set to "Don't Care" are set from the first character of the replaced text. If IGNORE CASE is on and the first character of the text to be replaced is upper case, then the first character of the replacement text is made upper case.

Operation - When Star is substituting, the message "Substituting..." is displayed. If the user did not request confirmation of each change, the substitution runs to completion with no updating of the display. When completed (i.e. the entire range has been searched), Substitute restores the selection and display to their states before the Substitute began (but not undoing any changes, of course). A message is displayed indicating the number of substitutions made.

If the user requests confirmation, as each potential substitution is found, it is selected and displayed on the screen according to the Positioning Rule. The message "Make this change?" and a choice parameter with the choices YES and NO are displayed in the Message Area. The user chooses either YES or NO. Either choice causes the substitution to continue.

Stopping a Substitute - A user may terminate an ongoing Substitute by pressing the STOP key. Substitutions made up to that point remain in effect. The selection and display are restored to their states before the Substitute began.

Ellipsis Substitute - As described in the Search command, a user may include a single ellipsis character in each of the Search For and Change To parameters. (If an ellipsis appears in Change To but not in Search For, an error message is displayed.) In Change To, the ellipsis is replaced by the characters matched by the ellipsis in Search For, if any. Thus, substituting "abc...def" for "123...456" would change "123xxx456" to "abcxxxdef".

Unusual Conditions - Like Search, the Substitute command operates in the window containing the selection or caret. If there is no selection or caret, an error message is displayed when FIND is pressed. Similarly, if the user chooses CURRENT SELECTION and there is only a caret, an error message is displayed when Start is invoked or when FIND is pressed for the second time.

ABBREVIATION EXPANSION

The abbreviation expansion features allow the user to include previously entered text (or any valid component of a document) by "name." This serves a variety of functions, from allowing the user to define a shorthand abbreviation for a commonly typed name or phrase, to the inclusion of "standard" paragraphs from "libraries" of information. Abbreviation expansion can be invoked any place the user enters text.

OBJECTS

Abbreviation

An *abbreviation* is a text string with which a replacement, or "expansion," text string has been associated. Each abbreviation/expansion pair is represented by a named field. The name of the field is the abbreviation; the contents of the field is the expansion. Abbreviations and their expansions must obey the normal field constraints (see *Field Definition*). In particular, the abbreviation (field name) must be unique in its document, and it can be no more than 100 characters in length. The expansion can be of any length and may contain anything that is legal in a document: text characters, formatting characters, fields, footnotes, equations and frames. The contents may have any legal properties.

Abbreviations Document

An *abbreviations document* is a standard Star document containing abbreviations. It can be edited, filed, mailed, printed, etc. It may contain anything, but abbreviation expansion examines only its fields and their contents.

When the system adds/deletes abbreviations (as described below), it operates in a standard format that enhances readability. Each expansion is in one or more separate paragraphs. The first line is the abbreviation. (This is redundant - it merely enhances readability.) The remainder of the paragraphs are in a field containing the expansion. When the user directly edits an abbreviations document he may do whatever he wishes, but it is recommended that he follow the system's format.

Abbreviations Folder

Every user has a folder named "Abbreviations" associated with his Desktop. This folder resides in the Directory (see *Directories*). It cannot be deleted. When expanding abbreviations, Star uses the abbreviations in the documents in the Abbreviations folder. When a new user is added, this folder contains a single document, "Standard Abbreviations" with some useful abbreviation/expansion pairs.

Thus, a (useful) abbreviations document is merely one containing fields. To be used in abbreviation expansion it is placed in the Abbreviations Folder. Whenever an abbreviations document is added to the Abbreviations Folder, Star ensures that the visible text preceding each abbreviation matches the name of the field containing that abbreviation. The user is thereby assured of the correspondence between internal and external representations of the abbreviation.

ACTIONS

Define Abbreviation

A user can define new abbreviations "on the fly" or by explicitly editing an abbreviations document.

When entering text, the user can define an expansion without opening a second document window to edit an abbreviations document directly. This is done by pressing the DEFINE key. The Define Expansion option sheet (see Figure TXT-5) is displayed. This can be done any time commands are permitted. If a sequence of characters is selected (or some text has just been entered) before the DEFINE key is pressed, the characters are copied into the Expansion parameter; otherwise it is empty. The caret is placed in the Abbreviation parameter. The user enters the Abbreviation and Expansion and invokes Start (or presses DEFINE again). As in the Search For and Change To parameters of Search and Substitute, text in the Expansion parameter has an additional "Don't Care" state for each of its text properties. The system adds the expansion to the *first* document in the Abbreviations folder. If the abbreviation already exists in that document, a warning message is displayed, and the user is asked to confirm replacing the existing definition. If the Abbreviations folder is empty when DEFINE is pressed, an error message is displayed.

A user can create a separate set of expansions (for example, a set of standard paragraphs for creating legal documents) in a document other than Default Abbreviations by directly editing an abbreviations document. He can enter an abbreviation by inserting a field, placing the expansion inside the field, and naming the field with the abbreviation (see *Field Definition* and *Field Fill In*). The user is prevented from having duplicates by the requirement that field names be unique.

> [NOTE: We can provide a template in the Default Abbreviations document which the user can invoke to insert a template for a new abbreviation/expansion. This would assist users who directly edit abbreviations documents in conforming to the system formatting requirements.]

When the user requests that an abbreviation be expanded (see below), the system searches the documents in the Abbreviations folder from front to back. Thus, the user can affect the expansion process by ordering the documents in this folder; for example, the Default Abbreviations document need not be first.

Delete Abbreviation

The Define option sheet enables a user to Delete an abbreviation. Entering the name of an abbreviation in the Abbreviation parameter and specifying a null expansion causes that abbreviation and its expansion to be deleted. The user may also delete an abbreviation by editing the appropriate abbreviations document.

Expand Abbreviation

When entering text the user can cause an abbreviation (a character sequence) to be expanded to a character sequence (including special objects) defined in one of the abbreviations documents in the Abbreviations folder. This is done by pressing the EXPAND key. The word immediately to the left of the caret or the last word of the character selection (not including trailing blanks or special characters) is assumed to be an instance of an abbreviation. The searching for an abbreviation is done regardless of case - "chair" and "Chair" are the same. The instance is deleted and replaced by the pre-defined string of characters, which are automatically selected.

The replacement of the abbreviation by the expansion follows exactly the same rules as in Substitute Text and Properties above.

Unusual Conditions - If the abbreviation is not found in any of the documents in the Abbreviations folder, an error message is displayed, and the document is not changed.

[NOTE: This is a very straightforward implementation of Abbreviation Expansion. It relies on the use of a document's internal symbol table (where field names are stored) to speed the search for the abbreviation. If this performance is not adequate, other enhancements will be added. For a user with ~100 small expansions (~50 characters each) it is expected that inserting the expansion will take about .1 - .5 seconds more than a COPY of the same amount of text. In general the amount of time to expand an abbreviation will be affected by the length of the document in which the abbreviation is found. This suggests that short abbreviations should be grouped in one document, long ones in another, since the user will probably expect the delay when a long expansion is being inserted. (The search time will also be "hidden" by the longer COPY and display update times.)

Also note that the "primary" user interface to any abbreviations document is through the DEFINE option sheet and placing the desired document at the front of the Abbreviations folder. Direct editing allows more general access. This is a low-risk approach in that only one interface (DEFINE) is built on top of standard document features. This will suffice for Star-1. If it is proven inadequate, another specialized set of routines can be added to facilitate directly editing an abbreviations document.]

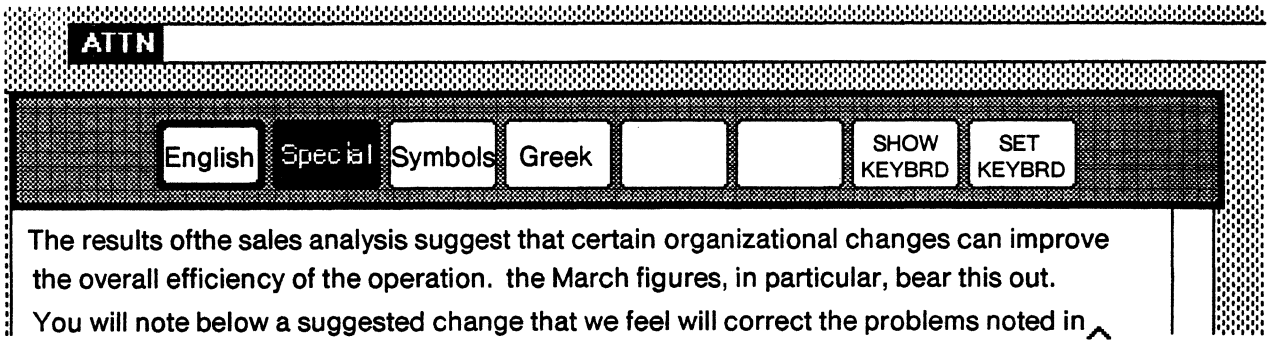
SPACE	·
Non-breaking space	ˏ
Discretionary Hyphen	•
TAB	→
PARA-TAB	⇛
NEW LINE	↲
NEW PARAGRAPH	↵
Page Format Character	▦
Start of Document	▸
Stop Character	●

(when displayed)

Normal Caret	∧
Underlining Caret	⩟
Boldface Caret	▲
Italic Caret	𝛬
Strike-out Caret	∧̶

(or a combination of the above)

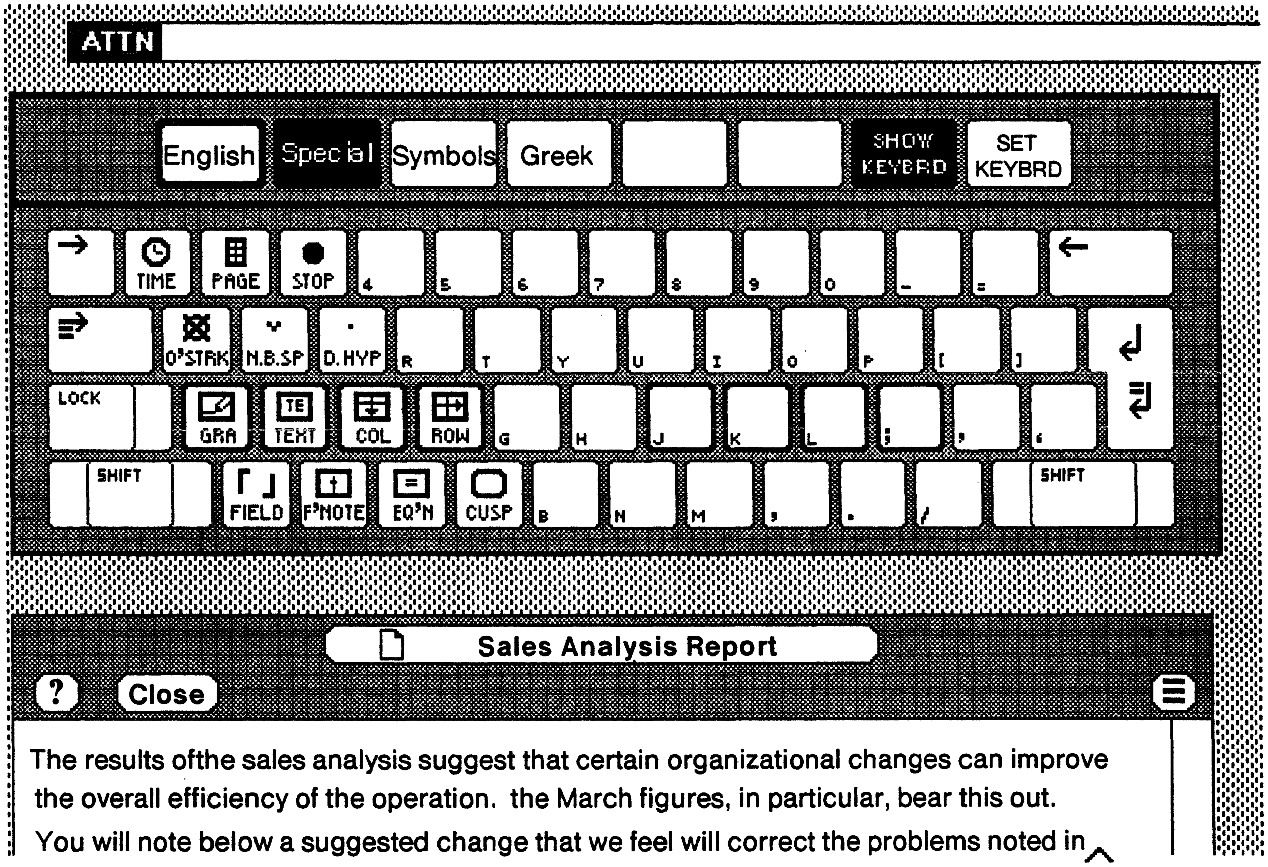
Figure TXT-1 -- Display of Special Characters and Insertion Caret



The results ofthe sales analysis suggest that certain organizational changes can improve the overall efficiency of the operation. the March figures, in particular, bear this out. You will note below a suggested change that we feel will correct the problems noted in^

Figure TXT-2a

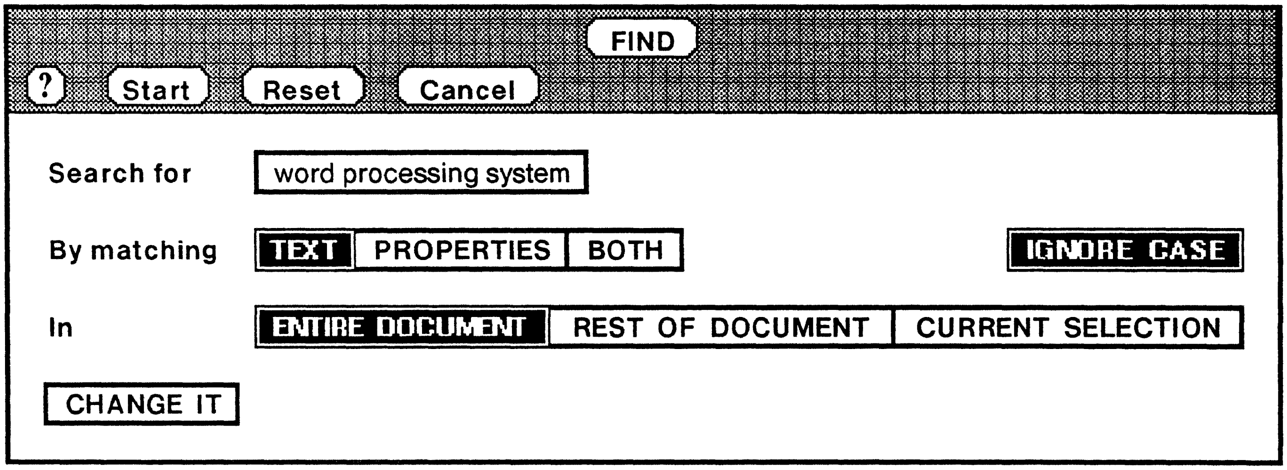
With the KEYBOARD key held down, the window header is overlaid with keyboard settings and alternatives (shown here, English is the permanent interpretation). The alternatives can be chosen by selecting with the mouse or pressing the corresponding property key (Center, Bold, etc.). Special is shown being chosen.



The results ofthe sales analysis suggest that certain organizational changes can improve the overall efficiency of the operation. the March figures, in particular, bear this out. You will note below a suggested change that we feel will correct the problems noted in^

Figure TXT-2b

After choosing an alternative keyboard set, the key assignments can be shown by selecting Show Keyboard with the mouse or pressing the corresponding property key. The special characters available depend on context. The text context is shown. In Search and Filter parameters, other Special characters are added to the keyboard shown. Other contexts for special characters are Equations and Field property sheets.



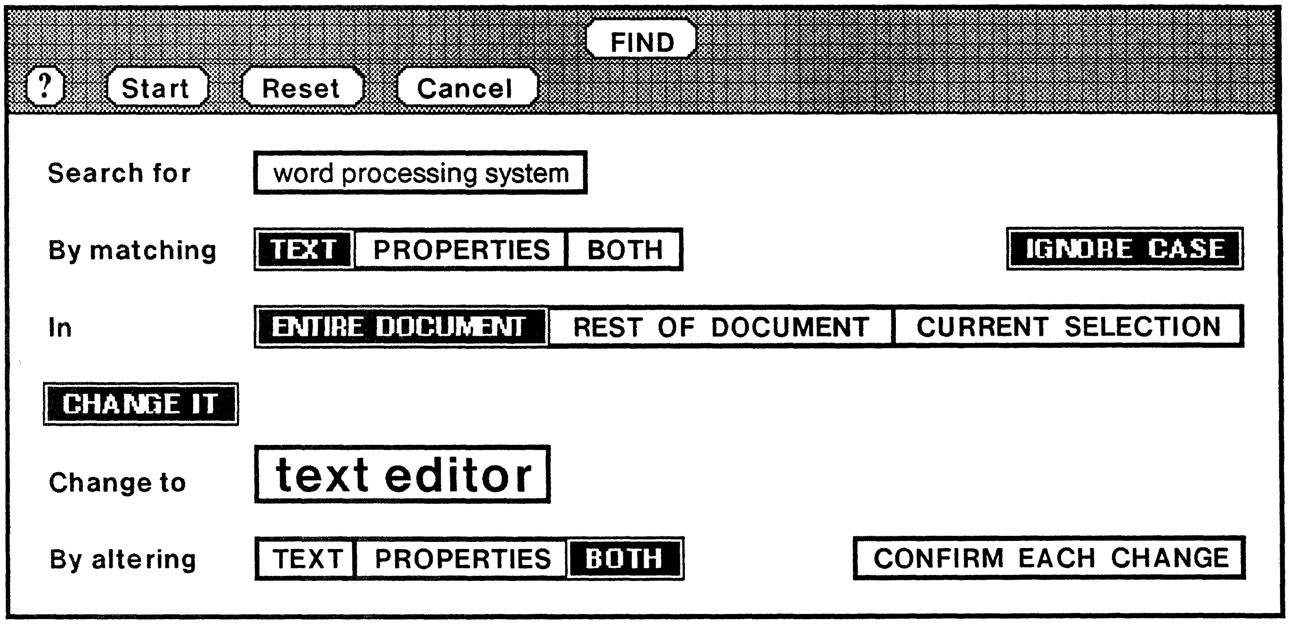
Search for: word processing system

By matching: [✓ TEXT] PROPERTIES BOTH [✓ IGNORE CASE]

In: [✓ ENTIRE DOCUMENT] REST OF DOCUMENT CURRENT SELECTION

CHANGE IT

Figure TXT-3 -- Find option sheet showing Search options



Search for: word processing system

By matching: [✓ TEXT] PROPERTIES BOTH [✓ IGNORE CASE]

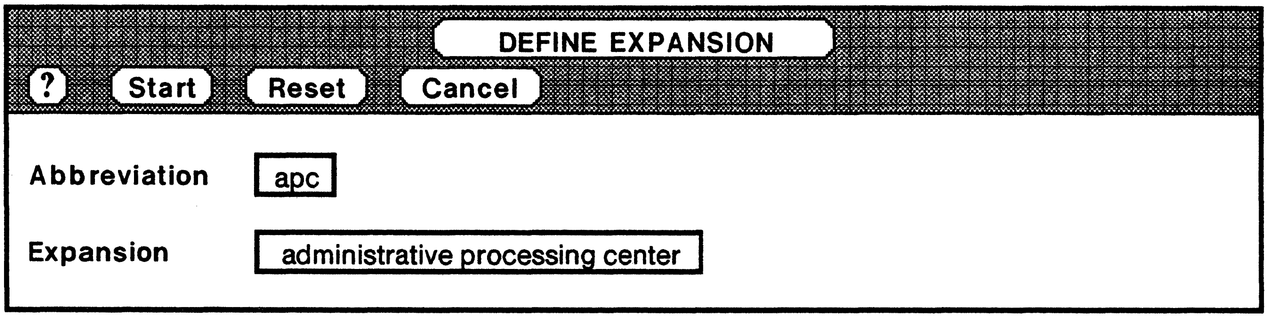
In: [✓ ENTIRE DOCUMENT] REST OF DOCUMENT CURRENT SELECTION

[✓ CHANGE IT]

Change to: text editor

By altering: TEXT PROPERTIES [✓ BOTH] CONFIRM EACH CHANGE

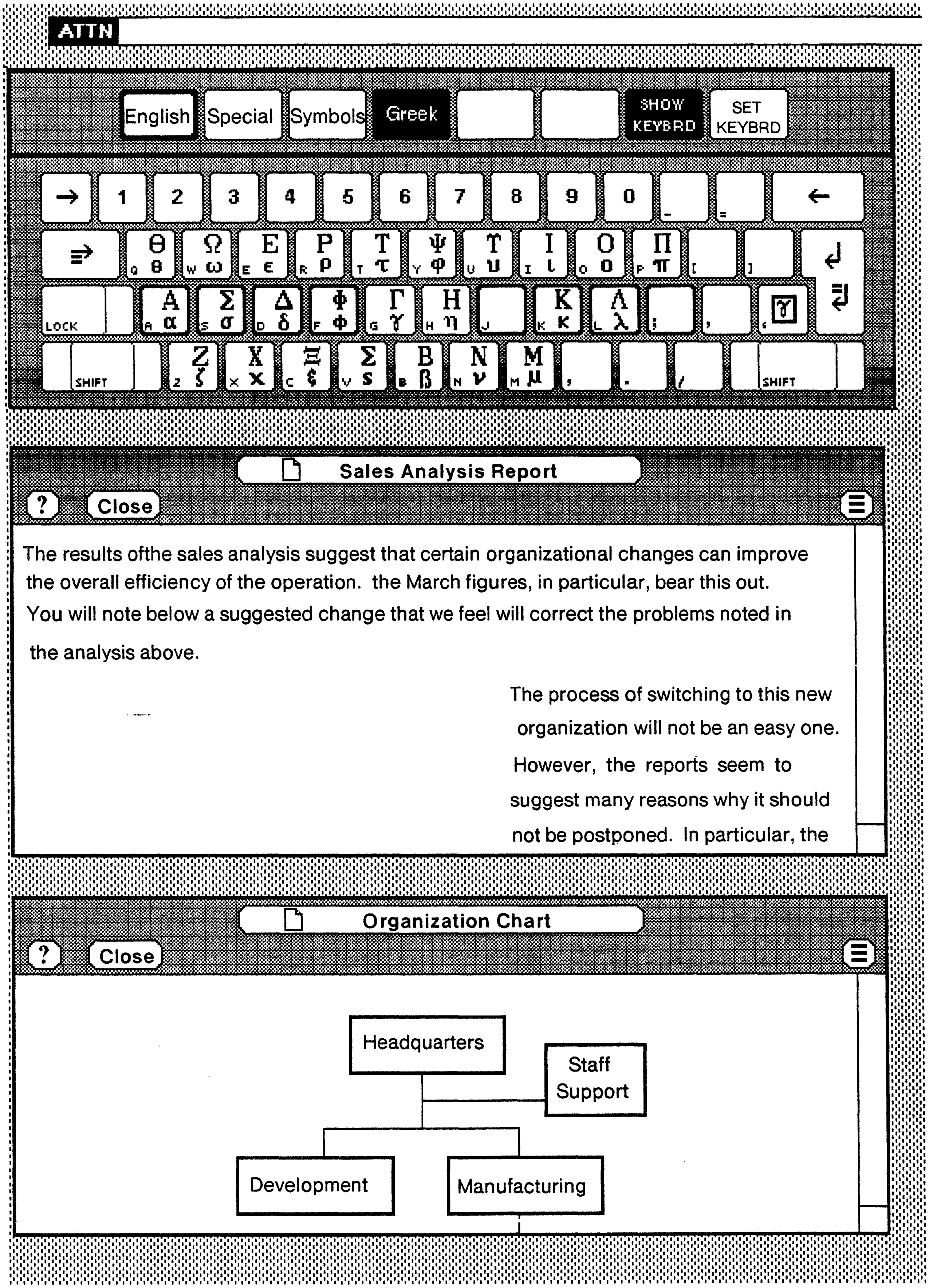
Figure TXT-4 -- Find option sheet showing both Search and Substitute options



Abbreviation apc

Expansion administrative processing center

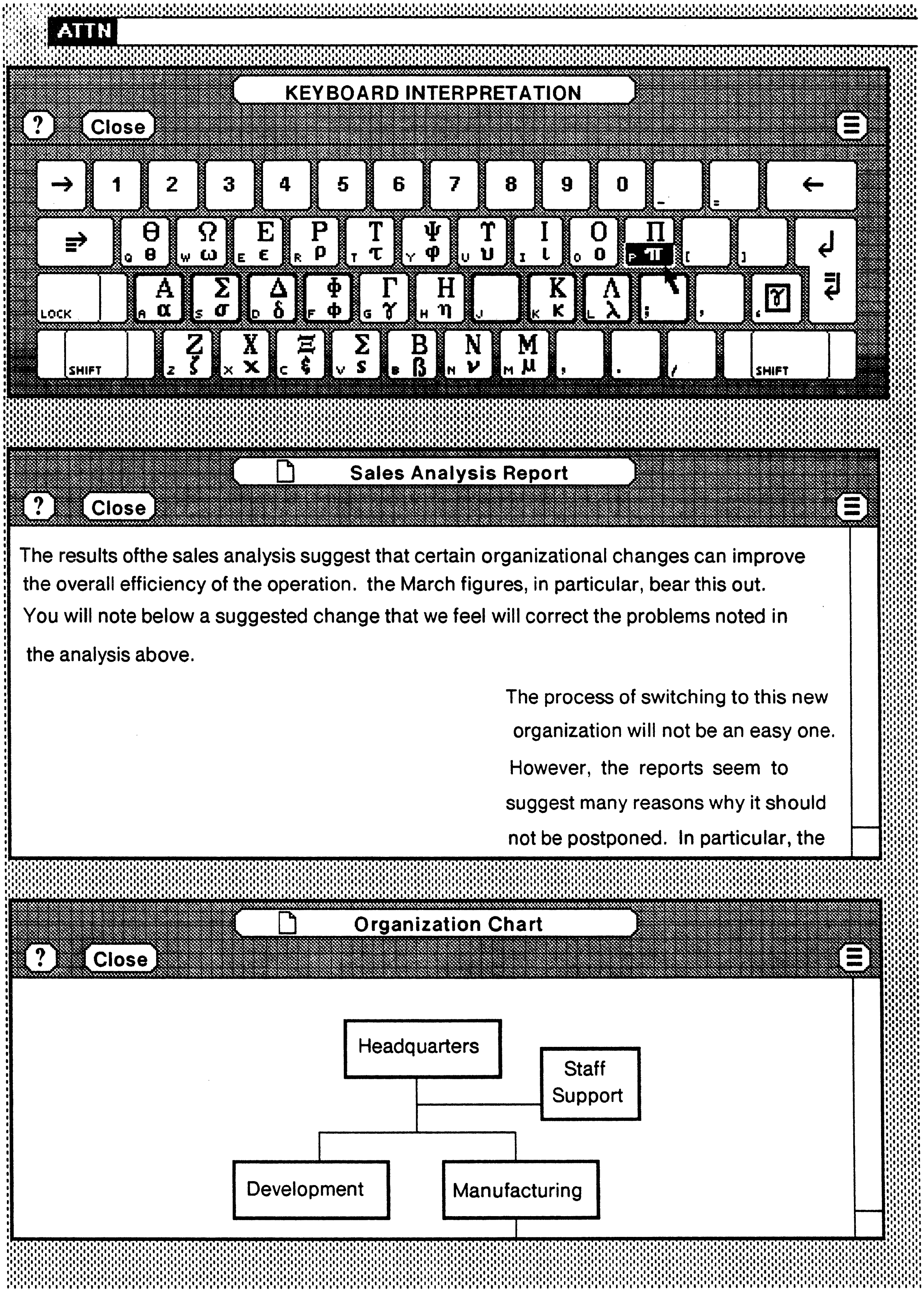
Figure TXT-5 -- Option sheet for defining an abbreviation and its expansion


ATTN
English
Special
Symbols
Greek
SHOW KEYBRD
SET KEYBRD
→ 1 2 3 4 5 6 7 8 9 0 _ = ←
⇛ Θ θ Ω ω Ε ε Ρ ρ Τ τ Ψ φ Υ υ Ι ι Ο ο Π π [ ] ↲
LOCK Α α Σ σ Δ δ Φ φ Γ γ Η η Κ κ Λ λ ; ,
SHIFT Ζ ζ Χ χ Ξ ξ Σ ς Β β Ν ν Μ μ , . / SHIFT
Sales Analysis Report
?
Close

The results ofthe sales analysis suggest that certain organizational changes can improve the overall efficiency of the operation. the March figures, in particular, bear this out. You will note below a suggested change that we feel will correct the problems noted in the analysis above.

The process of switching to this new organization will not be an easy one. However, the reports seem to suggest many reasons why it should not be postponed. In particular, the

Figure KBD-3 -- KEYBOARD Key Depressed -- Keyboard Window Open, Showing Greek Keyboard Interpretation



The results ofthe sales analysis suggest that certain organizational changes can improve the overall efficiency of the operation. the March figures, in particular, bear this out. You will note below a suggested change that we feel will correct the problems noted in the analysis above.

The process of switching to this new organization will not be an easy one. However, the reports seem to suggest many reasons why it should not be postponed. In particular, the

Figure KBD-4 -- KEYBOARD Key Released with Keyboard Window Open and Keyboard Set Permanently to Greek -- Keyboard Window Remains Open Cursor is Shown Poised to Mouse-Select the Character π

9. FORMATTING AND LAYOUT

Star renders characters and frames (e.g. tables and graphics) according to their associated *formatting and layout properties*. Users may display and alter these properties to cause a document to be rendered in the desired format.

In general, an object's properties control how it is printed and displayed. *Formatting* properties control the appearance of text objects. For example, characters may be boldface or italicized. *Layout* properties control both the positions of text objects, frames and special page objects on pages and the appearance of pages as a whole.

Every document has a set of initial character, paragraph, page, and document properties. Simple, direct mechanisms allow the user to view and modify properties.

A set of rules, collectively called the *rendering process*, displays objects based on their properties (see *The Rendering Process*). Thus, Star integrates document editing, formatting, and layout so that the user may perform them in any order.

USER OVERVIEW

Formatting and layout properties are displayed via property sheets and the Carriage. They are apparent from the appearance of the caret and of the document itself. Properties are modified using property sheets, the Carriage and property keys.

Properties are grouped according to the types of objects they affect and, in some cases, the functions they perform. The following objects have associated formatting and layout properties: *documents*, *sequences of pages*, *paragraphs*, *characters*, and *frames*. Page-specific objects such as headings and footings have properties, but they are associated with a page format character and are also affected by its properties.

Formatting and Layout properties involve the following property sheets.

> **Document Formatting Properties** - document wide properties such as widow control, revision bars, etc.
>
> **Page Layout Properties** - settings for page size, page margins, number of columns and new page control.
>
> **Page Headings and Footings** - properties to request headings and footings.
>
> **Numbering** - properties for page, line, and footnote numbering.

Paragraph Properties - paragraph indentation, alignment and spacing.

Tab Settings - settings for tab locations and tab alignment.

Character Properties - font selection, face setting, super/sub-scripting.

Frame Properties - settings that determine the size of the frame and where it is placed on the page. These properties are described in *Frames*.

An auxiliary menu for specifying distance units is associated with each property sheet having distance-related parameters. This allows the user to operate in the distance units most appropriate to his job, e.g. inches, millimeters, points, micas.

Formatting and layout properties for a sequence of characters, paragraphs, or pages can be copied from another object of the same type with the Copy Properties command (see *System Overview)*. The property keys provide a quick way to alter the most frequently changed character and paragraph properties. The Carriage is a horizontal scale that provides a simple, visual means of setting paragraph margins and tab stops.

Several *special characters* (see *Text Editing*) play major roles in formatting and layout. NEW LINE forces a line break. NEW PARAGRAPH carries paragraph properties (such as paragraph margins and tab settings) for the following paragraph. The PAGE FORMAT character carries properties (such as page margins, headings, number of columns, etc.) that affect the characteristics of successive pages. Entering a PAGE FORMAT character also inserts a NEW PARAGRAPH character. The START OF DOCUMENT character establishes default properties for the document; these are the properties used by the Default property sheet command and the DEFAULTS key on the keyboard. There is also a fixed PAGE FORMAT and NEW PARAGRAPH character at the beginning of each document.

DETAILED SPECIFICATION

OBJECTS

Property

A user can change the setting of any formatting or layout *property* at any time while the document is displayed in a document window. Properties are classified according to the type of objects they affect. In terms of formatting and layout this involves *characters, paragraphs, pages,* and the entire *document*. For example, a character property (Face) can cause text to be displayed with any combination of italics, boldface, underlined, or strikeout faces.

Properties are permanently associated with all entities in a document. This makes it possible to transfer documents between users without altering their appearance.

Property Sheet

Each formatting and layout *property sheet* contains a Display parameter and the parameters particular to the type of property sheet. The Display parameter can be used to access a different formatting and layout property sheet. For example, when altering character properties, the user might want to alter the properties of the paragraph in which the affected characters reside.

Each property sheet related to formatting and layout is described below. Detailed descriptions of all properties appear later under "ACTIONS".

Character Property Sheet - The *Character property sheet* (Figure FMT-1) contains information that controls how individual characters are rendered. All selectable characters defined in *Text Editing* have character properties. This includes characters such as SPACE, NEW LINE and NEW PARAGRAPH. A user can display the Character property sheet by selecting a character, word or sentence and pressing the PROPERTIES key, or by choosing CHARACTER as a Display option.

Paragraph Property Sheet - The *Paragraph property sheet* (Figure FMT-2) contains information that controls how individual paragraphs are rendered. Although tab settings are associated with individual paragraphs, they are specified via a separate property sheet (described below). A user can display the Paragraph property sheet by selecting an entire paragraph or a single NEW PARAGRAPH character and pressing the PROPERTIES key, or by choosing PARAGRAPH as a Display option.

Tab Settings Property Sheet - The *Tab Settings property sheet* (Figure FMT-3) is used to set tab stops for a paragraph. Star allows up to 100 tab stops to be defined (more if the design permits). A tab may be left flush, right flush, centered, or decimal-aligned. A user can display the Tab Settings property sheet by choosing TABS as a Display option.

Page Layout Property Sheet - The *Page Layout property sheet* (Figure FMT-4) contains miscellaneous information used to control the appearance of a sequence of pages. A user can display the Page Layout property sheet by selecting a Page Format Character and pressing the PROPERTIES key, or by choosing PAGE LAYOUT as a Display option.

Page Headings and Footings Property Sheet - The *Page Headings and Footings property sheet* (Figure FMT-6) is used to specify the positioning and format of page headings and footings. The actual text is entered directly into the top and bottom page margins, using normal editing operations. (The top and bottom margins are only visible when Show MARGINS is turned on in the Set Window option sheet.) A user can display the Page Headings and Footings property sheet by choosing HEADINGS as a Display option.

Numbering Property Sheet - The *Numbering property sheet* (Figure FMT-7) controls the numbering of pages, lines and footnotes. A user can display the Numbering property sheet by choosing NUMBERING as a Display option.

Document Formatting Property Sheet - The *Document Formatting property sheet* (Figure FMT-8) contains miscellaneous information that controls the formatting of the document as a whole, such as whether or not to collect revision marks. A user can display the Document Formatting property sheet by selecting either the entire document or the START OF DOCUMENT character and pressing the PROPERTIES key, or by choosing DOCUMENT as a Display option.

Units Transient Menu

Every property sheet that contains distance-related properties has a *Units auxiliary menu.* It is used to specify the units (inches, millimeters, points, micas, 10-pitch, 12-pitch) in which property settings are expressed. The units in effect may differ from property sheet to property sheet. This means, for example, that the user can express paragraph properties in points, and page layout properties in inches (see Figures FMT-2 through FMT-5). A Units transient menu is accessed as described in *System Overview*.

The Caret

Property sheets normally display properties for the *current selection*. When the user is entering text from the keyboard, there is no selection -- just a *caret*. The caret possesses a complete set of character properties, and therefore has an associated Character property sheet. When first placed by the user (see *Text Editing*), the caret inherits the character properties of the nearest preceding character. Subsequently, the caret's properties may be altered via property keys or by displaying and modifying its property sheet. Thus, the user may set character properties before entering the text to which they will be applied.

The shape of the caret reflects the state of its character Face parameters: BOLD, ITALICS, UNDERLINED, STRIKEOUT. (See Figure TXT-1.)

Property Keys

Property keys permit the user to change certain character and paragraph properties with a single keystroke, without recourse to property sheets. The BOLD, ITALICS, UNDERLINE, SUPERSCRIPT, SUBSCRIPT and LARGER/SMALLER keys affect character properties. The CENTER key affects paragraph properties. If the SHIFT key is held down while a property key is pushed, the effect of the property key is reversed. For example SHIFT BOLD means "not bold."

Property keys affecting character properties are applied to the current selection (if there is one) and to the caret. When applied to a selection, they affect every character in the selection. When applied to the caret, they affect the appearance of the caret and subsequently entered text. Property keys that affect paragraph properties affect either every paragraph that intersects the selection, or the single paragraph containing the caret, if there is no selection.

BOLD, ITALICS and UNDERLINE keys - These keys set/reset the BOLD, ITALICS and UNDERLINED properties.

SUPERSCRIPT and SUBSCRIPT **keys** - These keys alter the Position and Size properties of either the caret or all characters in the current selection. They raise or lower, respectively, affected characters from their current baseline and reduce the size. If smaller fonts are not available, the font size is not reduced. SUPERSCRIPT followed by type-in followed by SUBSCRIPT will subscript the superscript (i.e. lower the position and further shrink the font size). SUPERSCRIPT followed by SHIFT+SUPERSCRIPT will return the character position to the preceeding offset and size.

Each character in a document has an associated offset from its baseline. SUPERSCRIPT and SUBSCRIPT shift characters (an amount dependent on font size) relative to the baseline. SHIFT+SUPERSCRIPT will not move a character lower than the baseline; SHIFT+SUBSCRIPT will not move a character higher than the baseline. They are used for returning to the baseline.

LARGER/SMALLER **key** - This key changes the size of characters to the next larger or smaller size, respectively, available in the current font. If a smaller/larger size is not available, an error message is displayed.

CENTER **key** - This key affects the paragraph containing the caret and all paragraphs intersecting the selection, if there is one. CENTER causes all affected paragraphs to be centered between the current paragraph margins. SHIFT CENTER (uncenter) makes them left flush.

PARA-TAB **key** - This key inserts a special non-printing character which: 1) acts as a normal tab character on the current line (moving subsequent text to be left- or right-flush, decimal aligned or centered depending on the type of the next tab stop) and 2) moves the left margin of all subsequent lines in the paragraph to that tab stop (i.e. effectively making that tab stop a left-flush type). Normally the first character in an indented paragraph will be a PARA-TAB thus nesting the entire paragraph an amount determined by the first tab stop.

DEFAULTS **key** - This key sets all character properties for the selection/caret to their default settings, as determined by the START OF DOCUMENT character.

Carriage

The Carriage (see Figure FMT-9) provides a graphical means of setting tab stops and margins. It consists of a horizontal scale whose divisions represent positions at which tabs and margins may be set. Above the scale are two gray bands which represent the Left and Right paragraph margins. The downward-pointing arrows represent tab settings.

The smallest point on the scale (i.e. 0) corresponds to the left page (or column) margin. The largest point on the scale corresponds to the right page (or column) margin. For pages wider than the display, only a portion of the scale can be displayed (see "Show Carriage" for details). The scale is expressed in the units currently chosen on the Paragraph Units auxiliary menu.

The Carriage acts like a property sheet. However, the position of the carriage differs from that of property sheets (see "Show Carriage"). Its menu commands (Done, Apply and Defaults) function exactly like the corresponding commands on a property sheet menu.

NEW PARAGRAPH Character

The NEW PARAGRAPH character carries the paragraph properties of the paragraph that follows it. When copying paragraph properties with the Copy Properties command, a NEW PARAGRAPH character (when visible) can be used as the source for properties (see "Copy Properties"). When a NEW PARAGRAPH character is inserted into a document it has all of its properties automatically set from the previous NEW PARAGRAPH character in the document.

PAGE FORMAT Character

The PAGE FORMAT character carries page layout and formatting properties. When copying page layout properties with the Same As command, a PAGE FORMAT character (when visible) can be used as the source for properties (see "Copy Properties"). When a PAGE FORMAT character is inserted into a document it has all of its properties automatically set from the previous PAGE FORMAT character in the document. Entering a PAGE FORMAT character also inserts an associated NEW PARAGRAPH character.

START OF DOCUMENT Character

The START OF DOCUMENT character establishes default properties for the document. There is also a fixed PAGE FORMAT character and a NEW PARAGRAPH character at the beginning of each document. By changing these, a user can set up his own "styles" of documents.

Newly-entered NEW PARAGRAPH and PAGE FORMAT characters inherit paragraph or page properties from the immediately preceding NEW PARAGRAPH or PAGE FORMAT character, respectively. Thus, the characters at the beginning of documents serve as the source of properties in empty documents. By creating a number of blank documents and assigning different formatting and layout properties to their START OF DOCUMENT characters, a user can create a "library" of blank documents with standard, pre-assigned properties.

Unknown Character

The Unknown Character is represented as a black rectangle of arbitrary dimensions. Whenever a character is from an unkown font or character set. it is displayed and printed as the Unknown Character. Star attempts to preserve the line break positions from information stored in the file. However, if the user edits text containing the Unknown Character, then line break information may not be computed correctly.

ACTIONS

Show Properties

There are two ways to view the properties associated with an object: (1) *by inspection*, and (2) *by displaying property sheets (including the Carriage).*

Inspection can be used for readily discernable properties, such as paragraph alignment and character face. When entering text, the character face properties to be assigned to the next-typed character may be ascertained by examining the caret.

If one or more complete paragraphs or a single NEW PARAGRAPH character is selected when PROPERTIES (PROP'S) is pressed, a Paragraph property sheet is displayed. If a single PAGE FORMAT character is selected, a Page Layout property sheet is displayed. If everything in a document or the START OF DOCUMENT character is selected, the Document Formatting property sheet is displayed. If a frame, footnote marker, or footnote body is selected, its property sheet is displayed. If any other selection exists, or if only the caret is present, a Character property sheet is displayed.

Other property sheets may be displayed by selecting a choice on the property sheet Display parameter. This causes the currently displayed property sheet to be replaced by a property sheet of the type specified.

Change Properties

Formatting and layout properties can be modified in three ways: (1) by altering settings in property sheets, (2) by using the property keys, or (3) by using the Carriage.

Property keys are provided to facilitate the modification of certain properties involved in the rapid entering of text. The Carriage provides a more visual, direct means of setting certain paragraph properties.

Because property sheets are linked via Display options, a displayed property sheet might not apply solely to the objects in the selection (or solely to the caret). For example, if the user makes a character selection, presses PROPERTIES, and switches to the Paragraph property sheet, changes will affect *all paragraphs intersecting the selection.* Similarly, if CENTER is pressed when there is a character selection (or just a caret), all paragraphs intersecting the selection (or the paragraph containing the caret) will be affected.

The changes made to a Property Sheet are applied to the current selection when the Apply or the Done command is invoked in the property sheet menu.

The following rules define the object(s) to which a property sheet, the Carriage or a property key applies. (The caret is treated as a single character selection.)

> **Character property sheets or character-related property keys** affect all characters (text characters and special characters) in the selection.

Paragraph and Tab Settings property sheets, the CENTER **key, and the Carriage** affect all paragraphs intersecting the selection.

Page Layout, Page Headings and Footings, and Numbering property sheets affect the PAGE FORMAT character that precedes the selection (unless the first character in the selection is a PAGE FORMAT character) and all PAGE FORMAT characters contained in the selection. Indirectly, all pages controlled by the affected PAGE FORMAT characters are also affected.

Document Formatting property sheets affect the entire document.

Set Properties to Defaults

The Defaults command causes all property settings in a property sheet to be set to their default values. For each property discussed in this section, the default value is taken from the current setting of the same property of the START OF DOCUMENT character for the document. New default settings can be established by modifying the properties of the START OF DOCUMENT character.

Show Carriage

The Carriage (see Figure FMT-9) is displayed by pushing the MARGINS key. The Carriage cannot be displayed when a property sheet is already present. (If so, the error message "Cannot display Carriage while property sheet is displayed" appears.) The carriage is always positioned above the first character of the selection. Its width extends the width of the column or page (or frame) containing the first character of the selection.

The Carriage shows the properties of the first selected paragraph. Changes to the Carriage are applied to all paragraphs in the selection.

If the document is wider than the window in which it is displayed (e.g. a landscape page), the Carriage reflects only the portion of the page shown in the document window. Some tab stops and margin settings may not be visible. If the window is scrolled horizontally while the Carriage is displayed, the Carriage scale is updated appropriately.

Change Carriage Settings

When the Carriage is displayed, paragraph margins and tab stops may be set with the mouse. Tab stops are set by moving or copying the appropriate tab symbol to the desired point on the scale. Paragraph margins are set by moving the end of the appropriate grey band to the desired point on the scale. Tabs and margins can be placed only at marked points on the scale. Changes in the Carriage are reflected in the selected paragraphs when the Done or Apply menu command is invoked.

Symbols are moved, copied, and deleted in the same manner as graphic symbols (c.f. *Graphics*). Margins and tabs are converted into offsets from the page/column margins for display on Paragraph and Tab Settings property sheets. This enables Star to preserve the relative indentation of paragraphs if the page margins are subsequently changed.

> **Set Paragraph Margin** - Each of the margin bands contains a small, selectable control point at its innermost end. To set a margin, the user selects the control point with the mouse, presses MOVE, points to the destination and presses SELECT. The black band moves with the mouse until SELECT is released. The black bands are constrained in movement such that a paragraph can never be narrower than 1 inch.
>
> **Set Tab Stop** - Tab stops are set by copying or moving a tab symbol to the desired point on the scale. Tab symbols can be selected with the SELECT mouse button. The normal Move, Copy and Delete commands can then be used. The tab symbols above the scale (in the Carriage menu area) may be copied, but not moved or deleted. This insures that a full set of tab symbols is always available. Tab stops are removed by selecting the associated tab symbol on the carriage and hitting the DELETE key.
>
> The user may not place two tab symbols at the same point. A tab symbol may be outside the paragraph margins, but such tab stops cannot be reached. A maximum of 100 tab settings may be defined. If the user tries to create additional tab settings, the error message "You cannot have more than 100 tab stops." is displayed.

Carriage settings can be set to their default values with the Defaults menu command. The defaults are taken from the START OF DOCUMENT character.

Copy Properties

A sequence of like objects can be assigned the properties of another object of the same type using the SAME key as described in *System Overview*. The legal sources and destinations for this command are:

> **Character-related properties** may be copied to a sequence of characters from any "ordinary" character.
>
> **Paragraph-related properties** (those in the Paragraph and Tab Settings property sheets) may be copied to a paragraph or multiple-paragagaph selection or to (if visible) a NEW PARAGRAPH character from any point in a paragraph or from (if visible) a NEW PARAGRAPH character or the START OF DOCUMENT character.
>
> **Page-related properties** (those in the Page Layout, Page Headings and Footings, and Numbering property sheets) may be copied to (if visible) a PAGE FORMAT character from (if visible) any PAGE FORMAT character or from (if visible) the START OF DOCUMENT character or from any point in the document.

This process is simplified by constraining the type of selection made after Copy Properties is invoked to be of the same type as that of the initial selection, just as for Move or Copy. For example, if a user selects a paragraph, presses SAME, and then selects another piece of text, the entire paragraph containing that other text will be highlighted to indicate the source of properties to be copied. The determination of exactly which object(s) receive the copied properties is as specified above under "Change Properties."

Expand Properties

Abbreviations can be defined (with the DEFINE key) to permanently associate commonly used sets of properties with short identifiers. Those abbreviations can later be expanded (with the EXPAND key) when the user wishes to place the properties into a document. (This is discussed in more detail in *Text Editing*.) For example, the user can define an abbreviation named "ch" to "hold" a set of properties and a text pattern that defines the format of a chapter heading. The expansion for "ch" might be:

Chapter X
Chapter Heading Goes Here

To start a new chapter, the user would enter "ch" and press the EXPAND key. The "ch" would be replaced by the above expansion. The user would then alter the text of the expansion (e.g. "**Chapter X**" might be replaced by "**Chapter 7**"). Paragraph and character properties would already be correctly assigned.

Specify Units

Distance-related properties are expressed in the units currently chosen in the Units transient menu. They are set for a particular property sheet by selecting the auxiliary menu symbol in the property sheet header, and then selecting the desired Units choice in the menu that appears. The choices are (Figure FMT-5): USE INCHES, USE MILLIMETERS, USE POINTS, USE MICAS, USE 10-PITCH, and USE 12-PITCH. Distances may be specified using decimal notation (see Figure FMT-4). For fractional notation, the integer (if any) must be followed by one or more spaces, the numerator, a "/" and the denominator (e.g. "8 1/2").

One point equals 1/72 inch. One mica equals 1/2540 inch. 10-PITCH and 12-PITCH reflect standard typewriter scales having either 10 or 12 characters per inch. These are most commonly used with fixed-pitch fonts. Horizontal measurements for these choices are expressed in "spaces" (see Figure FMT-3), and vertical spacing is expressed in "lines" (see Figure FMT-2). One line equals 1/6 inch.

Set Character Properties

Character properties are specified by displaying the Character property sheet (Figure FMT-1) and setting the appropriate parameters. Some character properties may also be set via the keyboard, as described under "Property Keys."

The following parameters and options appear on the Character property sheet:

Font - Font is a choice parameter, which sets the type of font used to display characters. Typical choices are CENTURY, FRUTIGER, TITAN, PICA, SCIENTIFIC, and BOLD-PS.

As an optimization, the FONT key may be used to set character fonts. FONT is a shift key like the KEYBOARD key. While FONT is pressed, the font virtual property keys appear at the top of the screen, as described in *Keyboard Sets*. When the user presses one of the reassigned property keys shown, the corresponding font and size are set. If there is a selection, the characters in the selection are also converted to that font and size.

Size - Size is a choice parameter, which sets the size of the current font. The choices depend on the font selected. If the selected font is available in only one size, this parameter is shown as information only. As an optimization, the LARGER/SMALLER key may be used to set character sizes.

Face - State parameters are provided to modify the "face" of the current selection. The parameters are BOLD, ITALICS, UNDERLINE and STRIKEOUT (if all are available for the specified font and size). (STRIKEOUT causes a horizontal line to be drawn through the characters.) Any combination of faces may be in effect at one time. If the BOLD or ITALICS parameter is set for a font for which the option does not exist, the displayed image is not changed.

As an optimization, the BOLD, ITALICS and UNDERLINE keys can be used to set those attributes.

Position - This parameter sets the offset of a character from the document's baseline (and sometimes changes its displayed size). It is a choice parameter, with seven symbols representing the choices: normal, superscript, subscript, super-superscript, sub-superscript, super-subscript, sub-subscript, plus the choice OTHER. If OTHER is chosen, a text parameter Offset is displayed. Offset is expressed in terms of the current character units. Negative offsets lower the character relative to the baseline; positive offsets raise it.

The displayed (and printed) size and offset of a super- or sub-scripted character depend on the chacter's Size property. For example, a character with Size 18 points and Position SUPERSCRIPT might be displayed in a 12 point font with an offset of +6 points.

As an optimization, the SUPERSCRIPT and SUBSCRIPT keys may be used to change the Position property. A character can be superscripted or subscripted no more than twice. Further changes must be made to Position (OTHER) and Size independently.

Set Paragraph Properties

Paragraph properties are specified on the Paragraph property sheet (Figure FMT-2) and on the Tab Settings property sheet (Figure FMT-3). Paragraph properties and tab settings are carried by NEW PARAGRAPH characters.

Paragraph property sheet

Alignment - Alignment is specified as a combination choice and state parameter, with choices FLUSH LEFT, CENTERED and FLUSH RIGHT and state JUSTIFIED. FLUSH LEFT gives a ragged right edge. FLUSH RIGHT gives a ragged left edge. JUSTIFIED gives flush, non-ragged left and right edges (except for the last line of a paragraph, which is flush-left, centered, or flush-right, depending on the choice parameter setting). CENTERED centers all lines if justification is off; it centers only the lines narrower than the paragraph margins if justification is on. All alignment is relative to the paragraph margins (including indentation) currently in effect for the paragraph.

As an optimization, the CENTER keys may be used to alter paragraph alignment. SHIFT CENTER sets the alignment to left flush.

For text inside a frame, the Paragraph Property Sheet adds a FLUSH TOP/CENTER/BOTTOM parameter to the usual FLUSH LEFT/CENTER/RIGHT. This enables a user to control vertical positioning of text in a frame..

Hyphenation - Star can automatically hyphenate any English text. (It uses Knuth's TEX algorithm with a small exception dictionary.) A state parameter, USE HYPHENATION, determines whether Star uses hyphenation in the current paragraph. Text is automatically hyphenated as it is typed if USE HYPHENATION is on. If USE HYPHENATION is off, then no attempt to hyphenate words (except the normal use of hyphens and discretionary hyphens) will be made.

[The system has built-in algorithms for determining the parameters for hyphenation (hot zone, etc.). We consider this a major improvement over current systems which burden the user with this decision. If this proves wrong, hyphenation parameters can be made a user-settable parameter in future releases.]

Additional, optional points of hyphenation may be inserted into words as Discretionary Hyphens (see *Text Editing*).

Paragraph Indentation - The left and right paragraph indentation from the page margins can be set independently. Two text parameters enable the user to specify values in the current units. The Left parameter is added to the left page margin value to determine the effective left margin for the paragraph. The Right parameter is added to the right page margin value to determine an effective right margin for the paragraph. (Maintaining paragraph margins as indentations from the page margins, rather than as absolute positions on the page, enables paragraphs to retain their logical structure, even if page margins change.) Any combination of Left and Right values that would cause the Right margin to occur within one inch of the Left margin is not allowed, and an error message is displayed.

As an optimization, the Carriage can be used to display and alter paragraph indentation (as well as tab settings). Also, during typing the ParaTab key can be used to indent all subsequent lines in the same paragraph; this does not change the setting of the Left Indent on the paragraph property sheet.

Spacing - Three parameters allow the user to specify the spacing between lines within a paragraph, the space that appears before a paragraph, and the space that appears after a paragraph. These appear as choice parameters.

> **Between Lines** - This determines the amount of white space left between lines. The choices are SINGLE, 1 1/2, DOUBLE, TRIPLE and OTHER. These refer to single-line, one-and-a-half-line, double-line, and triple-line spacing, respectively. Line spacing

applies only to lines within the same paragraph.

SINGLE is always 1/6 inch or 12 points independent of font choice. For larger fonts (e.g. a 12 point font size), the user will want to use a larger value of Line Spacing; for example 1 1/2 (i.e. 18 points), would provide 6 points of space between 12 point high characters. More precise control of line spacing can be achieved with OTHER.

If OTHER is chosen a text parameter is displayed which allows the user to specify the space between lines in the current units. If the line spacing is insufficient to accomodate the characters on a line, the tops of the characters are chopped off.

Before Paragraph - This determines the amount of white space left above a paragraph. It has the same choices as described above. Before-paragraph spacing for the first paragraph on a page is not rendered.

After Paragraph - This determines the amount of white space left below a paragraph. It has the same choices as described above. After-paragraph spacing for the last paragraph on a page is not rendered.

Two paragraphs are separated by the sum of the after-paragraph spacing of the first and the before-paragraph spacing of the second.

None of these parameters may be negative. If they are, the error message "You cannot request negative spacing--it will cause characters to overlap." is displayed.

Same Page as Next Paragraph - This state parameter prevents Star from rendering the paragraph on the current page unless the first two lines of the next paragraph will also fit on the same page, subject to Widow/Orphan control. For example, this feature could be used to ensure that the major heading, sub-heading, and first line of a chapter all appear on the same page.

It is possible that so many successive paragraphs have SAME PAGE on that it is impossible to render them on any page. In such a case, Star renders paragraphs as if SAME PAGE had not been requested.

(NOTE: When a NEW PARAGRAPH character is entered. it does not inherit the SAME PAGE property of the preceding paragraph.)

Tab Settings property sheet

There are four types of tabs: left flush, right flush, centered and decimal-aligned. Text typed after a left-flush tab extends to the right, starting at the tab position. Text typed after a right-flush tab is placed immediately to the left of the tab position. Text typed after a centered tab is centered about the tab position. Text typed after a decimal-aligned tab is positioned as described under "Set Column Properties" in the section *Tables*.

Note: In European installations, a comma assumes the role of the period in decimal-aligned tabs.

When the caret is positioned such that there are no tab settings to the right which are less than the Right paragraph margin, pressing the TAB key causes the caret to be positioned at the first tab stop on the next line.

The following parameters appear on the Tab Settings property sheet:

> **Position** - Up to 100 tab positions may be set by entering values into the Position text parameter. Only non-negative values are allowed. Position is expressed in the current units and is interpreted as an offset from the current left page margin. Position values that would cause a tab stop to occur before the Left paragraph margin or after the Right paragraph margin are allowed, but the user will not be able to "tab" to these stops.
>
> **Tab Type** - Tab types are specified by selecting the appropriate pictorial choice from the adjacent Type parameter. The symbols are the same as those used on the Carriage. This property sheet reflects tabs set either with the property sheet or with the Carriage. If the selection includes paragraphs with different tab settings, only the settings for the first paragraph are shown but any changes are applied to all selected paragraphs.

When displayed initially, the Position/Type pairs are placed in ascending order according to the Position values. The user may enter tab positions in any order, however. There is always one blank entry at the end of the list. When the user fills in the Position text parameter of the blank entry, the default tab type is set automatically, and another blank entry is generated.

Changes in tab settings are not reflected in the document until the user invokes Done or Apply. If the user deletes the contents of a Position parameter and invokes Done or Apply, the tab setting is deleted.

Set Page Layout and Formatting Properties

Page layout and formatting properties are specified on the Page Layout (Figure FMT-4), Page Headings and Footings (Figure FMT-6), and Numbering (Figure FMT-7) property sheets associated with the affected PAGE FORMAT characters.

Page Layout property sheet

Page Control - A choice parameter displays the following choices: CONTINUE, NEW PAGE, NEW LEFT PAGE, NEW RIGHT PAGE and NEW COLUMN. CONTINUE causes Star to continue placing text in the same page and column. NEW PAGE causes a new page to be started. NEW LEFT PAGE and NEW RIGHT PAGE operate like NEW PAGE, but they leave an intervening blank page if necessary to advance to the next "left" or "right" page, respectively. (For a definition of "left" and "right" page, see the description of the First Page Binding Position property below.) NEW COLUMN causes a new column to be started; this is the same as NEW PAGE if there is only one column.

Page Size - The page size is specified via three choice parameters--**8.5x11**, **8.5x14**, **11x8.5**, **14x8.5** and OTHER. If the latter is chosen, two text parameters, Width and Height, are also displayed. The values entered must be positive.

If the Page Size parameter is different from the size previously in effect and Page Control is set to CONTINUE or NEW COLUMN, a new page is started automatically.

Margins - The Top, Bottom, Left and Right page margins are specified via text parameters as distances from the corresponding edge of the page. An additional indentation can be specified for the Binding margin (or "gutter"). All page margins are expressed in the current units.

Only positive values for which the sum of the Top and Bottom margins does not exceed the page Height minus one inch, and the sum of the Left, Right and Binding margins does not exceed the page Width minus 1 inch are valid. Left and Right page margin values that would invalidate the paragraph margins for any paragraphs controlled by the affected PAGE FORMAT characters(s) are invalid. If either of these cases occurs, the error message "The left/right page margins do not leave enough room on the page (for existing paragraphs)." is displayed.

The Binding margin is added to the left margin of "right" pages and to the right margin of "left" pages. The page width is effectively reduced by the size of the Binding margin. The Left, Right and Binding margin values determine the usable width of the page. Headings, footings and page numbers are entered into the top and bottom page margins.

Number of Columns - This text parameter specifies the number of columns of text to appear on a page. Only an integer value can be specified. If a value greater than 1 is chosen, the spacing between columns is specified by a text parameter (see below). All columns on a page have the same width.

If the user changes the number of columns in mid-page, any multiple columns appearing on the previous portion of the page are automatically balanced in length. At pagination time, Star attempts to balance columns at the end of the document or whenever a page ends.

Spacing Between Columns - This parameter is displayed only if Number of Columns is greater than one. The value entered specifies the distance by which columns are to be separated. It is expressed in the current units. Only non-negative values that would permit all columns to fit on the page are valid. If too large a number is entered, the error message "The requested inter-column spacing does not leave enough room for existing paragraphs." is displayed. All columns have the same inter-column spacing. If this parameter is not specified, it is set by default as a function of the number of columns and the page size.

Note that, at the user's option, line numbers and revision markers may appear within the inter-column spacing. Inter-column spacing is split in half. One half acts as a right margin (for the column to its left) while the other half acts as a left margin (for the column to its right). If the applicable margin is too small to accomodate a given line number or revision marker, that item is not rendered.

Page Headings and Footings property sheet

Parameters on the Page Headings and Footings property sheet specify whether or not page *headings* and *footings* are to exist. *The text comprising these entities is not specified in the property sheet* but is entered directly into an accurately sized representation of the top and bottom margins (when Show Margins is turned on in Set Window option sheet). Figure DOC-1 shows an example.

Heading - This state parameter allows insertion of text in the top page margin.

Headings Start On - This parameter determines whether the new heading will be printed on the page containing the page format character or just on subsequent pages. (The first page in a chapter frequently does not have a heading.)

Footing - This state parameter allows insertion of text in the bottom page margin. There is no Starting With parameter for footings, they start on the current page.

Left/Right Pages Different - This parameter indicates whether the "left" and "right" page headings and footings are to be independently positioned and specified. It causes separate heading and footing frames for "left" and "right" pages to be made available. (If different headings but identical footings are required, the user can copy text entered in the left footing into the right footing on the next page.)

The position of headings is controlled by aligning the paragraph(s) in the heading area, e.g. LEFT FLUSH, JUSTIFIED, CENTERED, RIGHT FLUSH on a Paragraph Property Sheet. The page margins define the width and height of the heading area. The Paragraph property sheet for text inside the heading area adds a FLUSH TOP/CENTER/BOTTOM parameter in addition to the usual FLUSH LEFT/CENTER/RIGHT.

(The heading and footing position parameters are initially set to CENTERED.)

Enter Headings and Footings - As stated above, page headings and footings are entered *directly* into the appropriate page margins. This is only possible when a document is displayed with Show MARGINS turned on in the Set Window option sheet (otherwise top and bottom margins are not visible). For a freshly entered PAGE FORMAT character, Star provides default headings. The text of each default heading is initialized from the preceding PAGE FORMAT character that has HEADING on. The user may instead select HEADING, LEFT HEADING or RIGHT HEADING--depending on page properties and the nature of the page being rendered. Similarly, Star provides the following default footings: FOOTING, LEFT FOOTING or RIGHT FOOTING. The user may retain the default setting, or alter it with a new PAGE FORMAT character.

The user may edit headings and footings in the usual manner; their text is treated like ordinary text in a document. Headings and footings are more than mere text, however. They exist in special areas, the *Top Page Margin* and the *Bottom Page Margin.* The areas' sides are constrained to the left and right page margins and cannot be changed. The content of the areas each consist of one or more paragraphs. When one of these areas is modified, all corresponding page headings or footings are affected. In effect, the alteration occurs simultaneously on *every* applicable page, whether before or after the page at which the alteration takes place. (Of course, the alteration doesn't affect pages outside the scope of the current PAGE FORMAT character.)

Numbering property sheet

The Numbering property sheet controls the numbering of various pages and lines within the sequence of pages controlled by the associated PAGE FORMAT character.

Page Numbering - Star can automatically generate page numbers which the user can surround with an optional text pattern. The Page Numbering choice parameter has three choices: NONE, CONTINUE and RESTART. If NONE is chosen, no page numbering will appear. If CONTINUE is chosen, page numbers will continue from the point at which they left off in the previous page sequence. If RESTART is chosen, a page number will appear, but pages will be numbered beginning with the number (or letter) specified in the Restart With text parameter, below. If RESTART is chosen, the following parameters also appear:

> **Start With** - When first displayed, it is set to the first number (or letter) in the numbering sequence chosen for the Format parameter.
>
> **# Format** - This choice parameter specifies the format to be used for system-generated page number. The choices are integers ("1,2,3"), lower-case letters ("a,b,c"), upper-case letters ("A,B,C"), lower-case Roman numerals ("i,ii,iii"), and upper-case Roman numerals ("I,II,III"). (Letter sequences double after the 26th page: "aa" follows "z.")
>
> **Pattern** - This text parameter allows the user to specify surrounding text. The # symbol is always present and cannot be deleted, moved or copied. Its character properties can be changed (e.g. BOLD, ITALIC) and arbitrary text can be inserted in front of or behind it to construct the entire page number pattern. For example, if the Format parameter was set to "1,2,3", then the pattern appearing on page number one would be "1". The user could add to the pattern, for example, "Page IV.# of IV.22". On page 1 the page number would read "Page IV.1 of IV.22"; on page 2 of the page sequence, the pattern "Page IV.2 of IV.22" would automatically be generated.
>
> **Margin** - This choice parameter has two choices: TOP and BOTTOM and determines where the page number is placed.
>
> **Position** - This choice parameter specifies where the page number pattern is to appear within the selected page margin. The choices are LEFT, CENTERED, RIGHT and

OUTER. OUTER is equivalent to RIGHT on "right" pages and LEFT on "left" pages. (Note that the page number pattern can potentially overlay the heading or footing, since the position of the page number is independent of heading and footing positions.)

Examples of the types of numbering patterns that are generated with the above number formats appear below. In the examples, the character " # " stands for the system-generated page number.

The pattern:	with format:	prints as:
#	integer	1, 2, 3, 4, ...
(#)	lower-case letter	(a), (b), (c), (d), ...
-#-	lower-case Roman	-i-, -ii-, -iii-, -iv-, ...
1.#	integer	1.1, 1.2, 1.3, 1.4, ...
XXIV-#	upper-case Roman	XXIV-I, XXIV-II, XXIV-III, ...
Page # of 22	integer	Page 1 of 22, Page 2 of 22, ...

To view the page numbers Show MARGINS must be turned on in the Set Window option sheet.

Line Numbering - This parameter has two choices: NONE and START. If START is chosen, lines are numbered beginning with the specified interval (see below) on each page.

All line numbers are rendered as integers. If START is chosen, the following parameters are also displayed:

Interval - This text parameter indicates the intervals at which line numbers are to be generated. Only positive integers may be entered. Only those line numbers that are integral multiples of the Interval parameter are printed.

Line numbering starts fresh at each page break. Numbering continues from one column to the next.

Editing affects line numbers in the following way: *Any* editing of a numbered line eradicates its line number. This includes property changes, insertions, deletions, substitutions and moves. Furthermore, any reformatting which causes a line to gain characters or break at a new point also eradicates the line number. For example, the effects of editing the first line in a paragraph may ripple through some or all of its lines; such lines become unnumbered. If a numbered line is copied, it remains numbered (at the source, not the destination). Line numbering is regenerated during pagination.

Position - This parameter has choices LEFT, RIGHT, OUTER and BOTH. These specify which margins are to contain line numbers. For multi-column pages, the inter-column spacing must also be considered. The space between two columns is divided in half. The left half serves as the "right margin" for the column to its left, while the right half serves as the "left margin" for the column to its right.

Note: If a given line number won't fit in its designated margin, the number isn't rendered.

The LEFT choice causes line numbers to appear in the left margin, while RIGHT causes line numbers to appear in the right margin. BOTH causes line numbers to appear in both margins. OUTER is equivalent to RIGHT on "right" pages and LEFT on "left" pages.

Footnote Numbering - Footnotes are described in detail in *Footnotes.*

Pattern - Four options for footnote marker symbols are available: asterisks, daggers, double daggers, and arabic numerals.

Start Option - This parameter displays three choices: CONTINUE, START and START AT PAGE BREAK. CONTINUE causes footnote numbering to continue from the point where it left off in the previous page sequence. START causes footnote numbering to restart with a specific number, and an additional parameter, Start With, appears. START AT PAGE BREAK is like START except that it does not take effect until the next page break.

Start With - This text parameter appears if the START or START AT PAGE BREAK option is chosen. It enables the user to enter a positive integer with which footnote numbering will start. It must be a positive integer.

Set Document Options

Document options are specified on the Document Formatting property sheet (Figure FMT-8). The parameters are:

Revision Marks - Revision marks will be generated automatically by Star, based on the following parameters.

Collect - This state parameter directs Star to keep track of changes to the document as it is edited. Collected revision data is somewhat akin to a character property. If text is inserted, each new character gets the revision "property." (A Copy or Global Replace is treated as an Insert at the destination. A Move is treated as a Delete at the source and an Insert at the destination.) When revision marks are rendered, they appear as vertical bars in the applicable margin. (They will be displayed only if Show MARGINS is turned on the Set Window option sheet.) All lines containing characters with the revision "property" are so marked. The position of these marks is controlled by the Position parameter. The COLLECT parameter can be turned off while insignificant changes are being made and turned back on when important changes are in process.

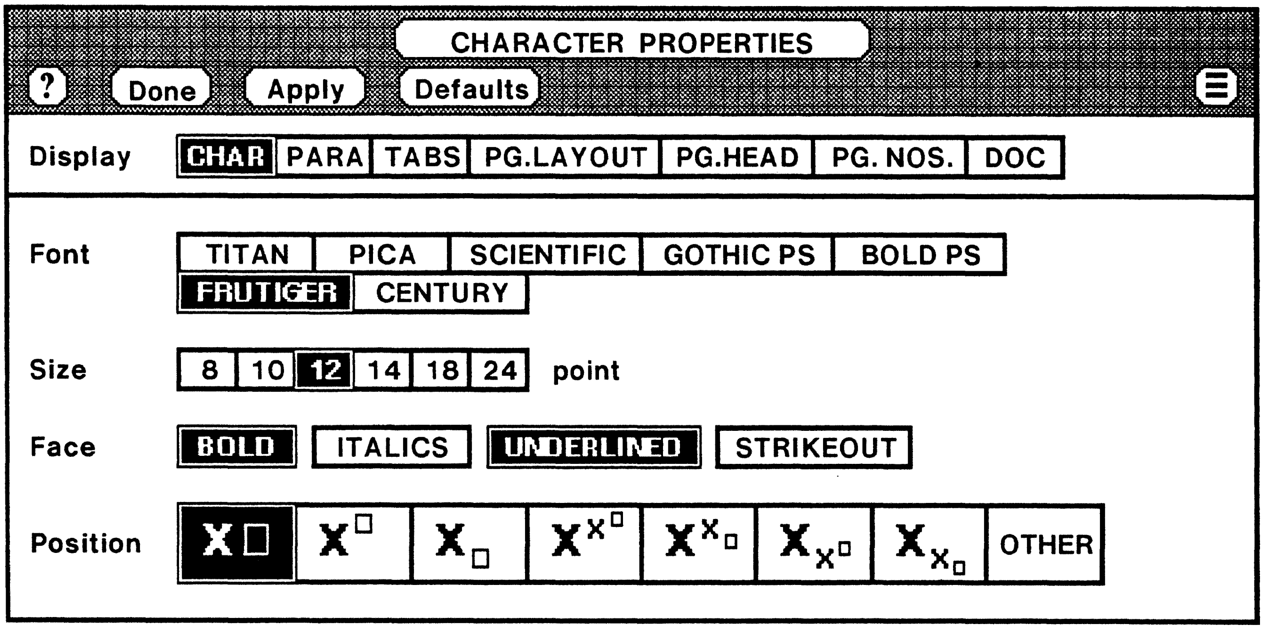
Position - This choice parameter indicates where revision marks should appear when the document is rendered: LEFT, RIGHT, BOTH or OUTER. If OUTER is chosen, the marks will appear at the left on "left" pages and at the right on "right" pages.

Revision marks are only collected for changes to the text in the main document scroll. No revision marks are generated for changes inside frames.

The Document transient menu contains the command Delete Revision Marks. This command causes all revision marks for the selection to be deleted.

First Page Binding Position - This choice parameter specifies whether the first page in the document is to be considered a "left" or "right" page with respect to the bound version of the document. The choices are LEFT and RIGHT. Whether or not a given page in the document is considered to be a "left" or "right" page depends on the setting of this parameter and on the physical position of the page, relative to the beginning of the document (i.e. whether it is the first page, second page, etc.). The document cover page is not considered when determining whether a page is a "left" or "right" page.

Content except fields: Locked - This is a state parameter indicating whether the document is locked (see *Field Definition*). If a document is locked, only the contents of fields in the document are allowed to be modified.



CHARACTER PROPERTIES

Display [✓ CHAR] PARA TABS PG.LAYOUT PG.HEAD PG. NOS. DOC

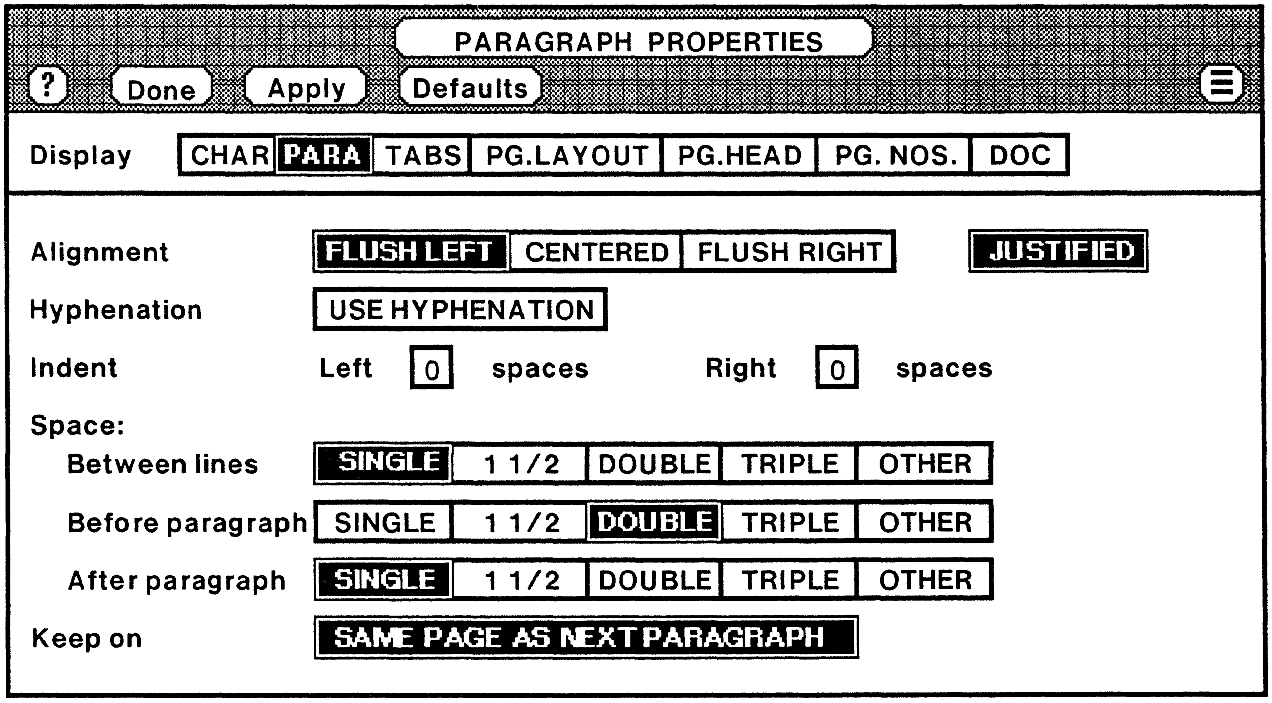
Font TITAN PICA SCIENTIFIC GOTHIC PS BOLD PS [✓ FRUTIGER] CENTURY

Size 8 10 [✓ 12] 14 18 24 point

Face [✓ BOLD] ITALICS [✓ UNDERLINED] STRIKEOUT

Position [✓ X□] $X^{\square}$ $X_{\square}$ $X^{X^{\square}}$ $X^{X_{\square}}$ $X_{X^{\square}}$ $X_{X_{\square}}$ OTHER

Figure FMT-1 -- Character property sheet



PARAGRAPH PROPERTIES

Display CHAR [✓ PARA] TABS PG.LAYOUT PG.HEAD PG. NOS. DOC

Alignment [✓ FLUSH LEFT] CENTERED FLUSH RIGHT [✓ JUSTIFIED]

Hyphenation USE HYPHENATION

Indent Left 0 spaces Right 0 spaces

Space:

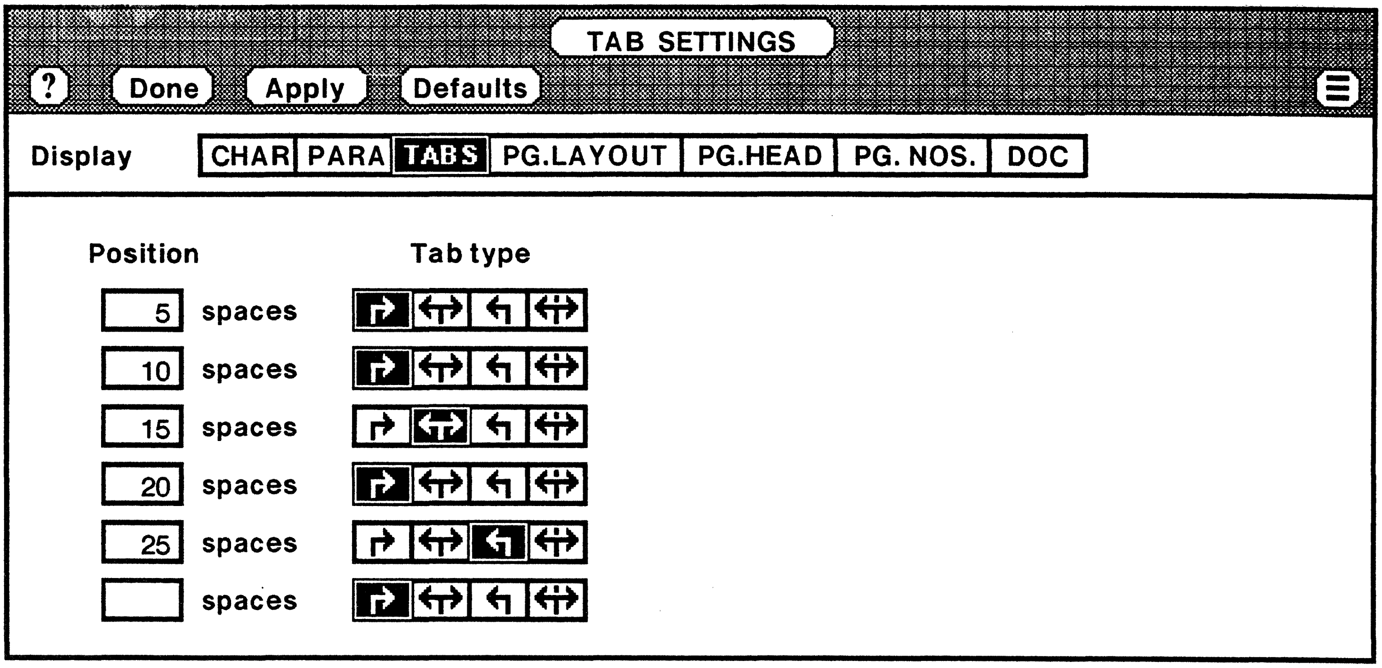
Between lines [✓ SINGLE] 1 1/2 DOUBLE TRIPLE OTHER

Before paragraph SINGLE 1 1/2 [✓ DOUBLE] TRIPLE OTHER

After paragraph [✓ SINGLE] 1 1/2 DOUBLE TRIPLE OTHER

Keep on [✓ SAME PAGE AS NEXT PARAGRAPH]

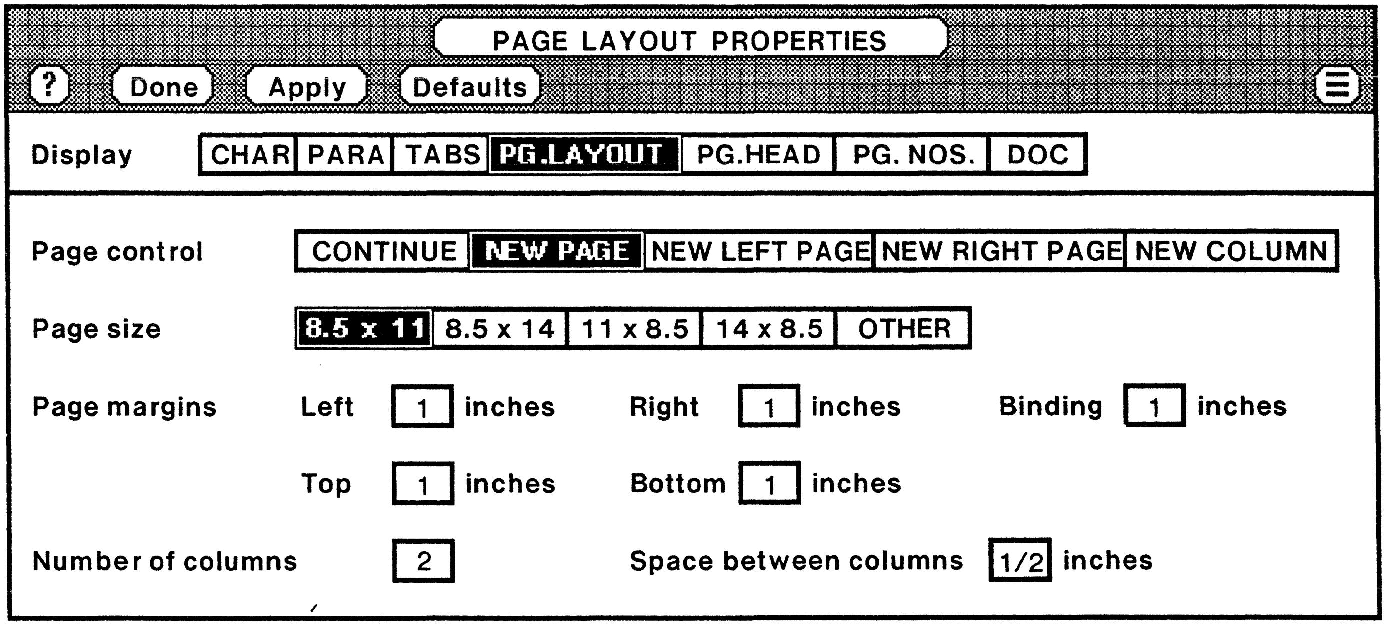
Figure FMT-2 -- Paragraph property sheet



Display CHAR PARA [✓ TABS] PG.LAYOUT PG.HEAD PG. NOS. DOC

Position		Tab type			
5	spaces	[✓ ↱]	↔	↰	⇹
10	spaces	[✓ ↱]	↔	↰	⇹
15	spaces	↱	[✓ ↔]	↰	⇹
20	spaces	[✓ ↱]	↔	↰	⇹
25	spaces	↱	↔	[✓ ↰]	⇹
	spaces	[✓ ↱]	↔	↰	⇹

Figure FMT-3 -- Tab Settings property sheet



Display CHAR PARA TABS [✓ PG.LAYOUT] PG.HEAD PG. NOS. DOC

Page control CONTINUE [✓ NEW PAGE] NEW LEFT PAGE NEW RIGHT PAGE NEW COLUMN

Page size [✓ 8.5 x 11] 8.5 x 14 11 x 8.5 14 x 8.5 OTHER

Page margins Left 1 inches Right 1 inches Binding 1 inches

Top 1 inches Bottom 1 inches

Number of columns 2 Space between columns 1/2 inches

Figure FMT-4 -- Page Layout property sheet

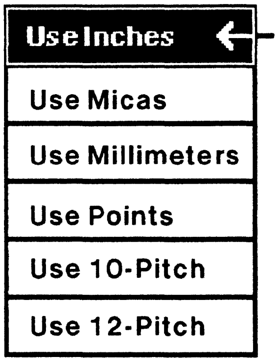
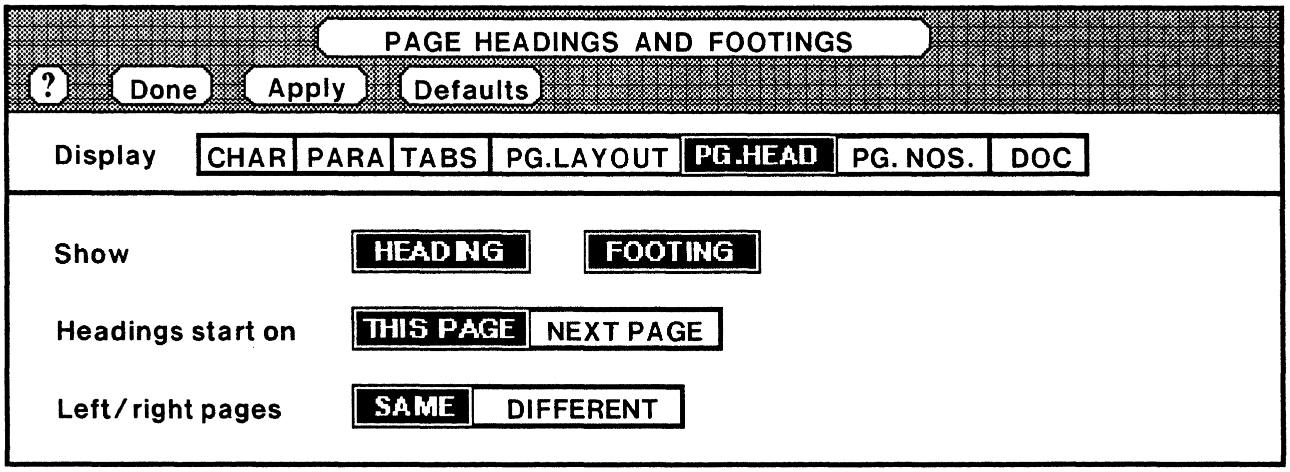


Figure FMT-5 -- Text Units transient menu



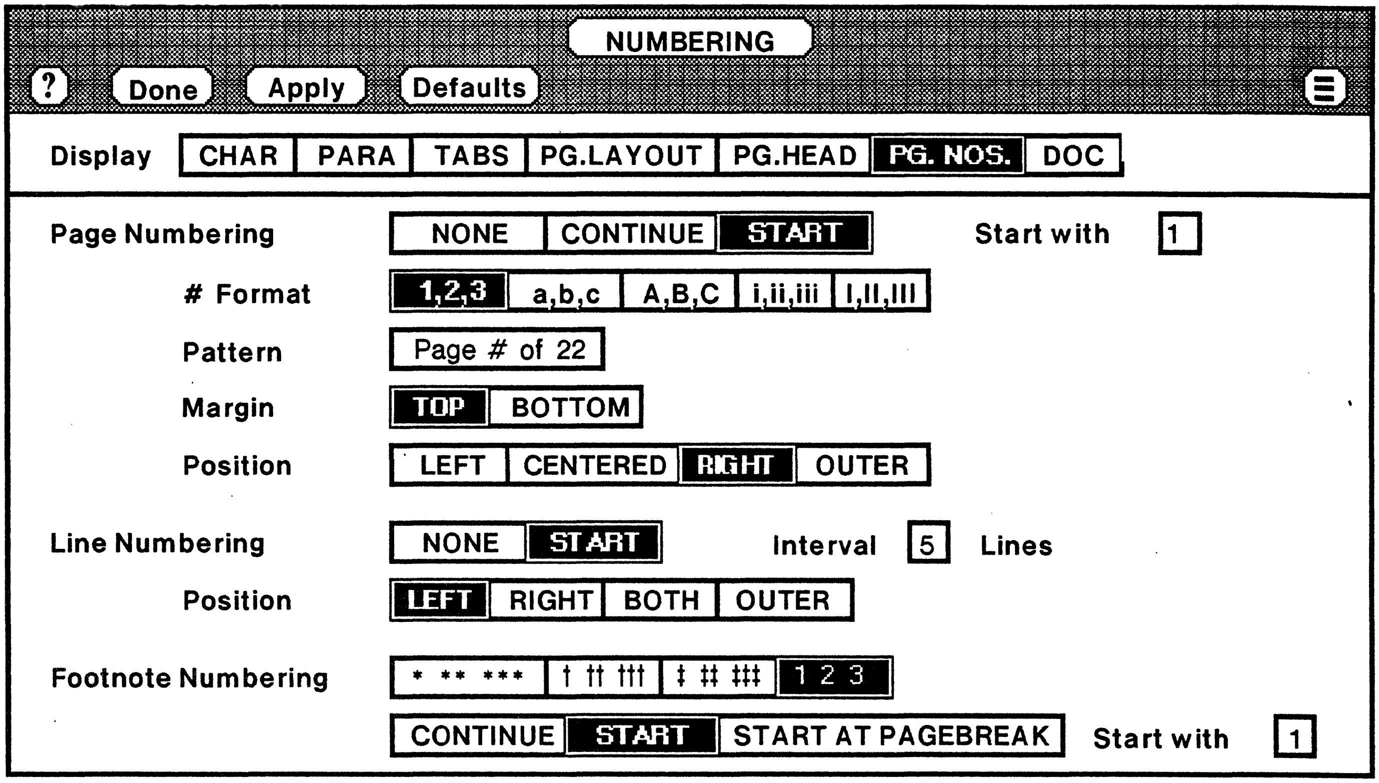
Display: CHAR | PARA | TABS | PG.LAYOUT | **PG.HEAD** | PG. NOS. | DOC

Show: **HEADING** **FOOTING**

Headings start on: **THIS PAGE** | NEXT PAGE

Left/right pages: **SAME** | DIFFERENT

Figure FMT-6 -- Page Headings and Footings property sheet



Display: CHAR | PARA | TABS | PG.LAYOUT | PG.HEAD | [✓ PG. NOS.] | DOC

Page Numbering: NONE | CONTINUE | [✓ START] Start with 1

Format: [✓ 1,2,3] | a,b,c | A,B,C | i,ii,iii | I,II,III

Pattern: Page # of 22

Margin: [✓ TOP] | BOTTOM

Position: LEFT | CENTERED | [✓ RIGHT] | OUTER

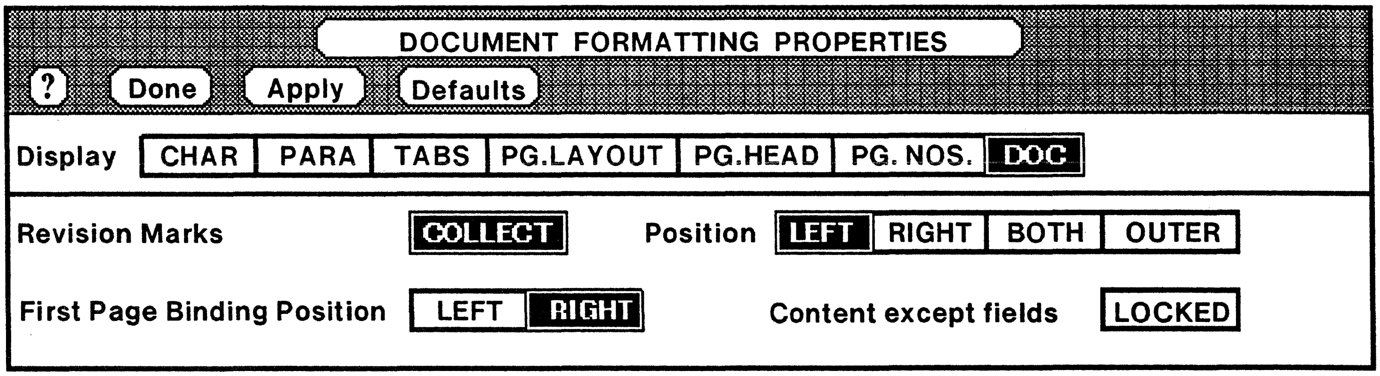
Line Numbering: NONE | [✓ START] Interval 5 Lines

Position: [✓ LEFT] | RIGHT | BOTH | OUTER

Footnote Numbering: * ** *** | † †† ††† | ‡ ‡‡ ‡‡‡ | [✓ 1 2 3]

CONTINUE | [✓ START] | START AT PAGEBREAK Start with 1

Figure FMT-7 -- Numbering property sheet



Display: CHAR | PARA | TABS | PG.LAYOUT | PG.HEAD | PG. NOS. | [✓ DOC]

Revision Marks: [✓ COLLECT] Position: [✓ LEFT] | RIGHT | BOTH | OUTER

First Page Binding Position: LEFT | [✓ RIGHT] Content except fields: LOCKED

Figure FMT-8 -- Document Formatting property sheet

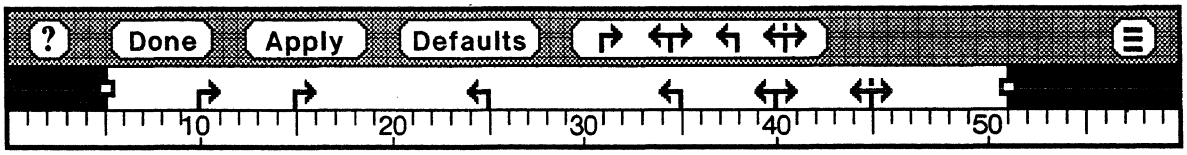


Figure FMT-9 -- The Carriage

10. FRAMES

The features described in *Text Editing* are adequate for the preparation of many documents. However, some types of documents contain other elements besides text. One of the principle means in Star for introducing these other elements into documents is the "frame."

USER OVERVIEW

As described in *Documents*, a Star document is composed of text characters, formatting characters, fields, footnotes, equations and frames. This section describes frames as objects and briefly touches on their contents. Other sections such as *Tables* and *Graphics* describe the contents of frames in detail.

A frame is a rectangular area in a document. It reserves "white space" into which the ordinary text of a document cannot intrude. (The text is said to "run around" the frame area.) Frames provide the means for spatially intermingling text, graphics, tables and other objects. A document page containing a frame is shown in in Figure FRA-1; the organization chart shown is composed of lines and text frames inside a graphics frame.

There are five types of frames: text frames, graphics frames, table frames, form field frames, and button frames. The types are distinct and different. Frames cannot be converted from one type to another; a frame is always a given type during its entire existence. A *text frame* can contain anything that is legal in a document; it is like an embedded document. A *graphics frame* can contain graphic objects such as lines and bar charts (see *Graphics*). A *table frame* is composed of table entries and ruling lines (see *Tables*). A *form field frame* is the combination of a field and a text frame (see *Field Definition*). A *button frame* can contain a Cusp program and can be used to invoke that program (see *Cusp*).

A frame may be placed in text, or it may be placed in a graphics frame. Frames may be nested inside text and graphics frames to any desired level.

A frame residing in text, either at the outer level of a document or in a text frame, is called an *anchored* frame. Its position is associated with an *anchor* that lies between two text characters. Star may reposition a frame (during pagination) if its anchor moves during text editing. The intent is that the frame stay as close as possible to the "designated" text -- i.e., the text surrounding the anchor. Two anchored frames may not overlap unless one is completely contained within the other; i.e. the outer frame is a text frame containing the other frame, and its anchor.

A frame residing in a graphics frame is called an *embedded* frame. Its position is fixed relative to the containing frame, as with graphic objects. Embedded frames may overlap arbitrarily. All frames are opaque, and obscure anything that lie "underneath" them.

DETAILED SPECIFICATION

OBJECTS

Frame

A frame is a rectangular area in a document that is used to reserve a region of (initially white) space. The minimum frame size is 1/8" by 1/8"; this ensures that it is always large enough to be selected. A frame can be up to ~1000" wide and ~1000" tall. Separate frame properties allow independent control over the width and height. The size of most frames may be fixed or variable, at the user's option; graphics frames always have fixed sizes. Variable size means that the frame can grow or shrink as required by its contents. Anchored frames are constrained by the enclosing page/column margins; normally a frame may not exceed the column width. Embedded frames may grow without bound; if they become too large for their containing frame, they "slide under" the edge of the containing frame.

All frames are named. If the user does not specify a name, a default name is assigned.

A frame has a border with eight "control points" (see *Graphics*), one at each corner and one in the middle of each side, just like graphic rectangles. Frames are selected, moved, copied, deleted, stretched and magnified exactly like graphic rectangles. Frame borders may be visible or invisible. Visible borders are available in a variety of styles and widths. Invisible borders are normally not displayed. However, they appear as dotted lines when (a) Show BOUNDARIES is turned on in the Window option sheet, (b) the frame is selected, (c) anything inside the frame is selected and the frame is the smallest one containing the selection, or (d) a mouse button is depressed inside the frame.

Text Frame

A *text frame* can contain anything that a document can, including text characters, formatting characters, fields, footnotes, equations and other frames. Frames inside a text frame are *anchored* (see below); text runs around them as at the document level. A text frame is like a miniature page. It has edges and margins similar to page edges and page margins; paragraph leading and indenting is added to those margins, as with pages. Footnotes are placed at the bottom of the page containing the reference, as always. Page Format Characters are ignored.

Graphics Frame

A *graphics frame* can contain any graphic transfer symbols (point, line, rectangle, triangle, ellipse) and/or graphic constructions (bar chart, pie chart, curve graph). It can also contain other frames, which are *embedded* (see below). The objects in a graphics frame may be placed arbitrarily within the frame, subject to grid constraints. Once placed, the objects do not change position unless the user explicitly moves them.

Graphics frames have a square *grid* which facilitates alignment and spacing of graphic objects. The user can turn the grid on and off, change its spacing, and change its visual appearance. When the grid is on, guiding point destinations during graphic Move, Copy, Stretch and Magnify commands are constrained to fall on grid points (see *Graphics*). When the grid is off, a destination may be any screen dot. The grid is never printed when a frame is rendered on paper.

Table Frame

A *table frame* (or "table" for short) consists of an array of fields divided into rows and columns, with ruling lines between the rows and columns and around the table as a whole (see *Tables*). It provides a way to enter and change tabular information without using tabs.

Form Field Frame

A *form field frame* (or "form field" for short) consists of an ordinary field (see *Field Definition*) joined with a text frame. It provides a way to insert a rectangular field in a document (without using tables). The field and frame are indivisible; they cannot be separated from one another. Neither can be selected unless both are. Anything inserted in the frame is always placed inside the field.

Button Frame

A *button frame* (or "button" for short) consists of a name, graphic lines surrounding the name, and an internal Cusp program (see *Cusp*). It provides a convenient and powerful interface to Cusp, particularly when buttons are included in forms.

Anchored Frame

An *anchored frame* is a frame that resides in text, either at the outside level of a document or inside a text frame. Any type of frame can be anchored; they are treated the same as far as positioning is concerned.

Anchored frames have a specific horizontal position: flush left, centered or flush right. They may also have a specific vertical position: flush top, centered or flush bottom, or they may "float" up and down with their anchors as text is edited. An anchored frame is always shown in its correct horizontal position, but its vertical position may not be shown correctly until the document is paginated.

A frame can be inserted in text and later moved or copied. The surrounding text is always reformatted to make room for the frame. This is accomplished as follows (see *The Rendering Process*):

> Once the user has positioned a frame in text, the system continues to place characters after the anchor until the line is full. For left-flush or right-flush frames that are less than the column width, subsequent text is placed in a narrower "column" between the frame's edge and the column's margin. This is called text "run around." (Text will not run around a frame unless there are at least two inches between the frame and the margin.) After the bottom of the frame, text is again

placed in full-width lines (unless another frame is encountered).

Star always attempts to place an anchored frame completely on one page. If the frame is longer than the page, Star will split it over pages when the document is rendered onto paper. Text and form field frames are split at line boundaries. Graphics and button frames are split wherever the bottom of the page occurs. Table frames are split at row boundaries if possible; otherwise, at line boundaries. In every case, the remainder of the frame is continued on the following pages. When a table is split over pages, a user may request that column headings be printed at the top of each continuation page. (Internally Star always keeps and displays frames in one piece. This may make the page containing a frame be a "long" page. Pagination never splits frames.) [This design for handling frames longer than a page is our current best guess. Actual implementation may alter the model.]

Embedded Frame

An *embedded frame* is a frame that resides in a graphics frame. Any type of frame can be embedded; they are treated the same as far as positioning is concerned. Embedded frames may overlap other objects including other frames in the graphics frame; there are no restrictions on overlapping. All frames are opaque and hide anything they overlap.

Embedded frames have a fixed position in the enclosing graphics frame. Unlike anchored frames, embedded frames do not move as surrounding objects are edited. Embedded frames also obey the graphics frame's grid.

Embedded text frames may contain anchored frames.

Frame Property Sheet (see Figures FRA-3 and FRA-4)

All frames have the following properties.

Name (text) - All frames are named. The defaults are "Frame#" (where # is a positive integer representing the frame's chronological order in the document) for text and graphics frames, "Table#" for table frames, "Field#" for form field frames, and "Button#" for button frames. The name cannot be empty. Users may change the name, but frame names must adhere to the same restrictions as field names (see *Field Definition*). In particular, they must be unique within a document.

Width, Height (text and choice) - Two text parameters control the size of a frame, each of which has an associated choice parameter, FIXED/VARIABLE. Graphics frames are always FIXED in both height and width. The defaults (for frames entered via the KEYBOARD key) are 1" x 1" VARIABLE for text, form field and button frames, 2" x 2" FIXED for graphics frames, and 2" x 2" VARIABLE for table frames. A user may alter the Width or Height text parameters only if the associated choice parameter is set to FIXED. When the Width or Height is VARIABLE, the text parameter is information-only. The FIXED/VARIABLE parameters control whether the frame automatically expands or shrinks horizontally and/or vertically to accomodate objects added to or removed from the frame. See "Change Frame Size" below.

Margins (text) - Four parameters control the left, right, top and bottom margins outside the frame. The defaults (for frames entered via the KEYBOARD key) are uniform 1/4" margins for text, form field and button frames, and 1/4" left, right and top margins and 1/2" bottom margin for graphics and table frames. The margins determine an area of white space around the border into which surrounding text may not intrude; this is most useful for anchored frames.

> NOTE: The *size* of a frame (as controlled by a user during Stretch or Magnify or when editing the Width or Height) is the usable space inside the border. The *frame size plus the margins* is the total space occupied by the frame.

Caption (state) - Two parameters, TOP and BOTTOM, control whether or not a caption can be entered in the top and/or bottom margin. The defaults are none for text, form field and button frames, and BOTTOM for graphics and table frames. Captions are similar to page headings and footings (see *Formatting and Layout*); a user types into them just as he does with headings and footings. Anything can be put in them that can be put in a text frame. They always occupy the full width between the left and right margins and the full height of the top or bottom margin. See Figure FRA-2.

Border Style, Border Width (choice) - These control the appearance of a frame's border. The choices in each are exactly the same as for graphic rectangles. The default is invisible (Style = NONE). When visible, the border is rendered on both the display and hardcopy.

Alignment (choice) - The user may specify the horizontal and vertical position of an anchored frame on a page. The horizontal choices are FLUSH LEFT, CENTERED, and FLUSH RIGHT; the vertical choices are FLUSH TOP, CENTERED, FLUSH BOTTOM, and FLOATING. The defaults are CENTERED and FLOATING. When a frame is embedded in a graphics frame, the Alignment property disappears.

Text frame properties

In addition to the basic properties shared by all frames, text frames have a ruling property.

Interior (state) - Turning RULED on causes ruling lines to be drawn in the frame. The distance between ruling lines is determined by the line height of the first contained paragraph. The default is no ruling.

Graphics frame properties

In addition to the basic properties shared by all frames, graphics frames have properties to control their grids.

Grid: ON/OFF, **Grid Style, Grid Spacing** (choice) - These control whether the frame's grid is on or off, and if on, its appearance on the screen. The grid Style and Size are present only if the grid is ON. The grid may be displayed with a dot or plus sign at grid points or by placing tick marks along the frame border. The available grid spacings are 4, 8, 12, 16 and 32 screen dots. The default is ON with a TICK grid at 8 unit spacing. The grid is never printed on hardcopy, even if it is visible on the screen.

Users can also turn a frame's grid on and off with the graphics soft keys (see *Graphics*).

Table properties

Tables have the general frame properties, plus separate Table and Column Headings property sheets (see *Tables*). Users can switch between the sheets via a Display parameter having options TABLE, FRAME and COLUMN HEADINGS.

Form Field properties

Form fields have the general frame properties, plus a separate Field property sheet (see *Field Definition*). Users can switch between the sheets via a Display parameter having options FIELD and FRAME.

Button properties

Buttons have the general frame properties, plus a separate Program property sheet (see *Cusp*). Users can switch between the sheets via a Display parameter having options PROGRAM and FRAME.

Frame Anchor

A frame's *anchor* lies between two text or formatting characters. Every frame residing in text has an anchor. The anchor determines on which page of the document the frame appears, as described in *The Rendering Process*.

ACTIONS

Create Frame

There are several ways to create a new frame in a document.

(1) A user can insert a frame with the KEYBOARD key as described in *Keyboard Sets*. In the Special keyboard set there are entries for a text frame, a graphics frame, two table frames, and a button frame. If the frame is inserted in text, it is immediately positioned as specified by its Alignment property, except that correct vertical positioning may not happen until the next time the document is paginated. The frame's anchor is inserted preceding the caret. The caret remains where it was (after the anchor), ready for further typing.

(2) A user can insert a frame by moving or copying it from somewhere else. (See "Move Frame" below.) There is an instance of every type of frame in the basic system

transfer sheet, to serve as a convenient source for Move and Copy when working in graphics. This is the typical source of form field frames. Users can also create transfer sheets containing their own predefined frames.

(3) A user can insert a frame into text via abbreviation expansion. He may define an abbreviation whose expansion contains one or more frames (and possibly associated text). These frames may have their own user-specified properties.

(4) A user can insert a frame into text by Substituting a string containing one or more frames (and possibly associated text) for a string that does not.

Select Frame

A user selects a frame by pointing to its border and clicking the SELECT button. The cursor must be in close proximity to the border, as described in *Graphics*. In the case of an anchored frame, the cursor must also be inside the border; otherwise it might be ambiguous as to whether the frame or a nearby character was being designated. If the frame border is invisible, it becomes dotted when a mouse button is depressed while the cursor is inside the frame; this aids in selecting frames with invisible borders. When selected, the rectangular area of the frame is inverted, all eight control points are highlighted, and the graphics soft keys are displayed.

Multiple embedded frames can be selected just like multiple graphic rectangles. The user selects one as described above, then points to another one and presses the ADJUST button. The second frame is added to the selection. Any number of frames can be so added. Clicking an already selected frame with ADJUST removes that frame from the selection. (See *Graphics* for further details on multiple selection.)

Except for table frames, a frame cannot be selected by multiple clicking in its contents. Nor does selecting an anchor cause an anchored frame to be selected. The user may include an anchored frame *in a text selection* by selecting text before or after the frame and extending the selection beyond the frame. (If Show Properties is invoked, the character property sheet appears displaying the properties of the first text character in the selection.) However the frame may now be moved, copied or deleted along with the selected text. Multiple anchored frames cannot be selected except as part of an extended text selection.

If a user types when a frame is selected, Star issues an error message and discards the characters.

Change Frame Size

A user can change the size of a frame in the following ways.

(1) He can show its properties and edit the Width or Height property. Each is editable only if the associated choice parameter is set to FIXED.

(2) He can invoke the Stretch command (the STRETCH soft key) or the Magnify command (the MAGNIFY soft key) to graphically change the size. The Stretch and Magnify commands work as described in *Graphics*. They always set the frame's width and/or height to FIXED, depending on what was changed. These commands do not change a frame's margins, only the area inside the border.

(3) He can edit the contents of the frame. This includes typing into a text frame and moving/copying into a graphics frame. If the frame is specified to have a VARIABLE width and/or height, the variable dimension always maintains the *minimum* size necessary to display all of its contents, if possible.

Changes in size are constrained as follows.

> The basic rule is: If a frame tries to get larger, for any of the reasons listed above, it is permitted to do so until it encounters an edge of its container. Its container may be another frame, a table entry, or a column in a page in a document. The frame then requests that its container expand. If the container can do so, the frame continues expanding. If the container cannot do so and the frame chooses to expand, it "slides under" the edge of its container.
>
> In order to expand, a container may have to request that *its container* expand, and so on up through the levels of nesting.
>
> A frame cannot grow in width by request of its contents if its width is FIXED. It cannot grow in height by request of its contents if its height is FIXED. If both dimensions are VARIABLE, a frame will expand horizontally as far as possible, then expand vertically if still more room is needed.
>
> Page column widths never expand upon the request of contained frames. They change only when a user alters the properties of a Page Format character.
>
> If a frame cannot grow large enough to hold its contents, *the contents are truncated*. They "slip under" the edge of the frame. They are retained but not (entirely) displayed until the frame is made larger.

During a size change, the grid in a graphics frame and the graphic objects inside it remain positioned relative to a corner that was not moved.

Changing a frame's size does not affect its Alignment nor its anchor. However, its position on the display and on the page may change as a result of its new size. For example, it may no longer fit on the page.

Move Frame

A user can move a frame by selecting it and pressing the MOVE key, as described in *System Overview*. When the user presses a mouse button to indicate a destination inside a graphics frame, the outline of a rectangle the size of the frame appears and moves with the mouse cursor, exactly like a graphic rectangle. When the mouse button is released, the outline of the rectangle disappears, and the frame is deleted from its original location and placed at the last position of the rectangle.

If the destination is text, a caret appears indicating where the anchor will be placed. The frame is placed "on" the following line and repositions itself horizontally to satisfy its property sheet settings when the mouse button is released; vertical repositioning is done at pagination. If a frame and its anchor are moved together as part of an extended *text selection*, the anchor remains the same. If a frame but not its anchor is moved as part of an extended text selection, the anchor is on the line preceding the frame. (Moving a frame as part of a text selection proceeds as if only text were being moved, as described in *Text Editing*.)

If a frame is moved into a graphics frame, the operation proceeds as described in *Graphics.* For multiply-embedded frames (i.e. a frame in a frame in a frame), the destination is the smallest legal frame enclosing the cursor when the mouse button is released. The moved frame assumes a fixed location with respect to the enclosing frame, and it's Alignment parameter disappears.

If the destination frame is not large enough to hold the moved frame, the destination frame will expand if possible; otherwise, it will clip (retain but not display) the excess contents. See "Change Frame Size" above.

Copy Frame

Copy works like Move except that the original frame remains unchanged where it was, and the new frame is assigned a default name.

Delete Frame

A user deletes a frame by selecting it and pressing the DELETE key.

Delete Anchor

If a user deletes the characters surrounding a frame's anchor, the anchor is left behind. A frame in text always has an anchor.

Show/Change Frame Properties

A user can display a frame's property sheet by selecting it and pressing the PROPERTIES key. The properties are changed in the normal way.

Show/Change Default Properties in a Text or Form Field Frame

The default paragraph properties are set by making a selection in the first paragraph in the frame and setting them in the normal fashion. Default text properties may be set for a form field frame by selecting the first field character (with Show NON-PRINTING CHARACTERS on) and setting properties in the normal fashion.

Copy Frame Properties

A user can copy the properties of one frame to another by selecting the frame to be changed, pressing the SAME key, and selecting the source frame. All the properties that the two frames have in common are copied except Name. Note that if Copy Properties is applied to frames of different types, some properties are ignored.

Insert Text in Text Frame

A user may enter text into a text frame in any way that is legal in a document: type it, move it, copy it, expand abbreviations, etc. All the normal text editing operations apply. He may also enter objects such as fields and equations, just as at the outer level of a document. If a text frame cannot expand large enough to hold its contents, it will retain but not display the excess contents, and a warning message is issued. See "Change Frame Size" above.

A text frame has many of the same characteristics as a document. In an empty text frame, a user may point inside the frame and click SELECT. This causes a caret to be placed in the frame at the position specified by the frame's first paragraph properties, e.g. top-left corner, top-center, etc. Characters typed are entered with the frame's default properties; a user can alter their properties in the normal way.

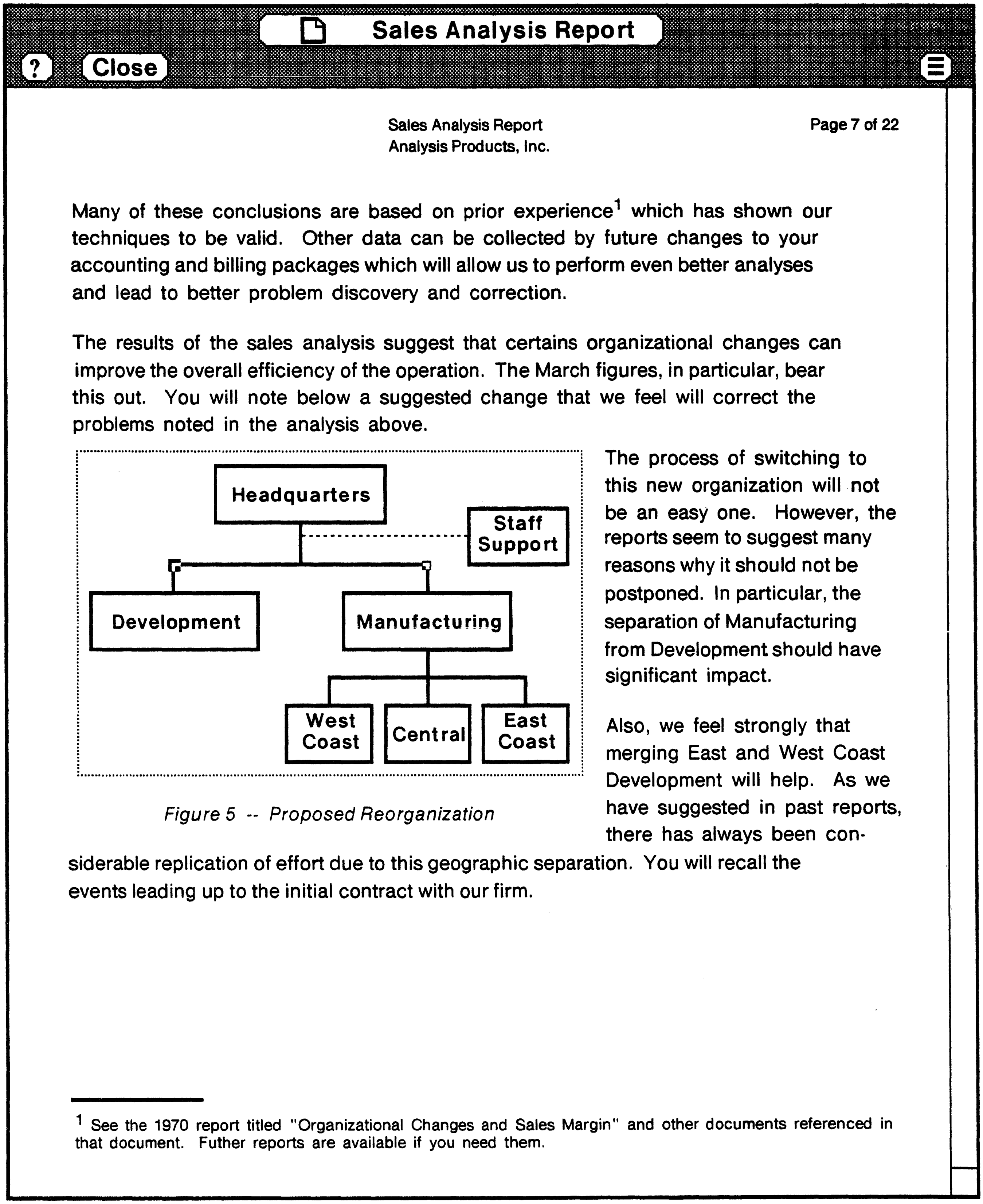
Insert Graphics in Graphics Frame

Graphics objects may be placed in a graphics frame as described in *Graphics.*

Search/Substitute in Frames

The Search and Substitute commands are described in *Text Editing.* During a search, the character contents of frames are searched if the frame is in the range of the Search. A Search may be limited to the interior of a frame by selecting just the frame or some part of its contents and choosing the CURRENT SELECTION Search option. Substitute works in a similar fashion.

In Substitute, the Change To parameter may contain one or more frames. However, the Search For parameter (what to substitute for) may not. If it does, an error message is displayed when Start is invoked. (This may seem arbitrary, but it prevents the user from accidentally deleting all frames in a document.)



Sales Analysis Report
Analysis Products, Inc.

Page 7 of 22

Many of these conclusions are based on prior experience[1] which has shown our techniques to be valid. Other data can be collected by future changes to your accounting and billing packages which will allow us to perform even better analyses and lead to better problem discovery and correction.

The results of the sales analysis suggest that certains organizational changes can improve the overall efficiency of the operation. The March figures, in particular, bear this out. You will note below a suggested change that we feel will correct the problems noted in the analysis above.

Figure 5 -- Proposed Reorganization

The process of switching to this new organization will not be an easy one. However, the reports seem to suggest many reasons why it should not be postponed. In particular, the separation of Manufacturing from Development should have significant impact.

Also, we feel strongly that merging East and West Coast Development will help. As we have suggested in past reports, there has always been considerable replication of effort due to this geographic separation. You will recall the events leading up to the initial contract with our firm.

[1] See the 1970 report titled "Organizational Changes and Sales Margin" and other documents referenced in that document. Futher reports are available if you need them.

Figure FRA-1 -- Document containing a flush left graphics frame

If a frame contains the selection, its border is displayed.

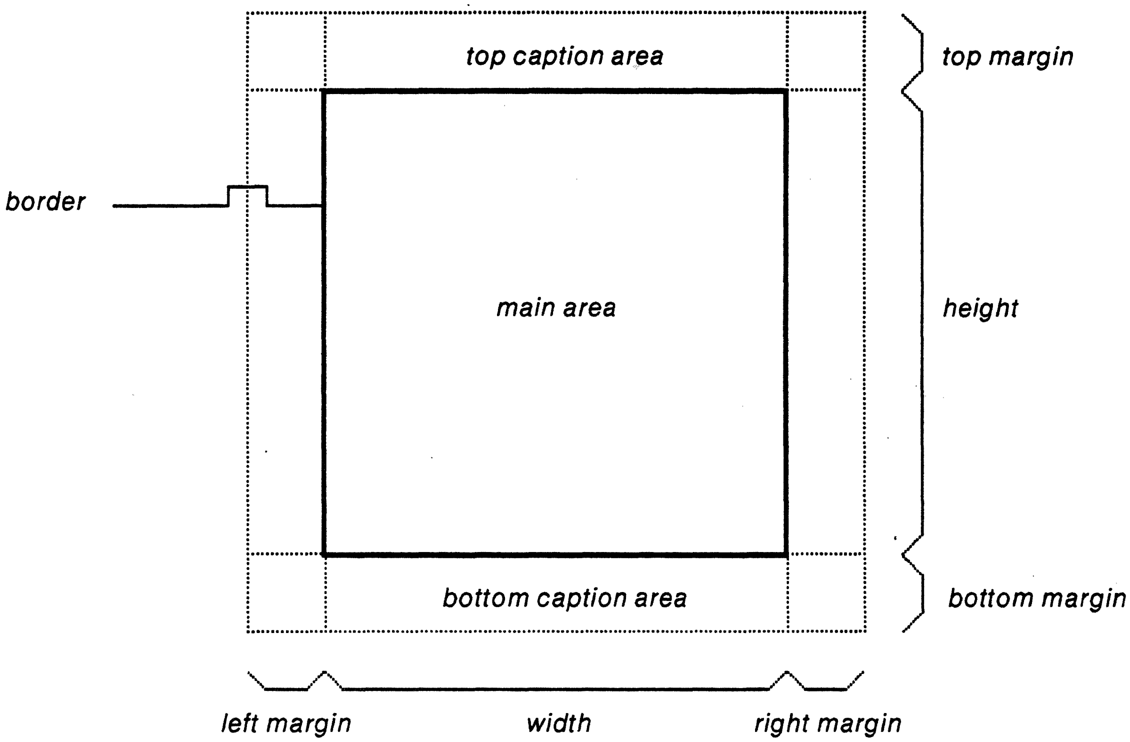
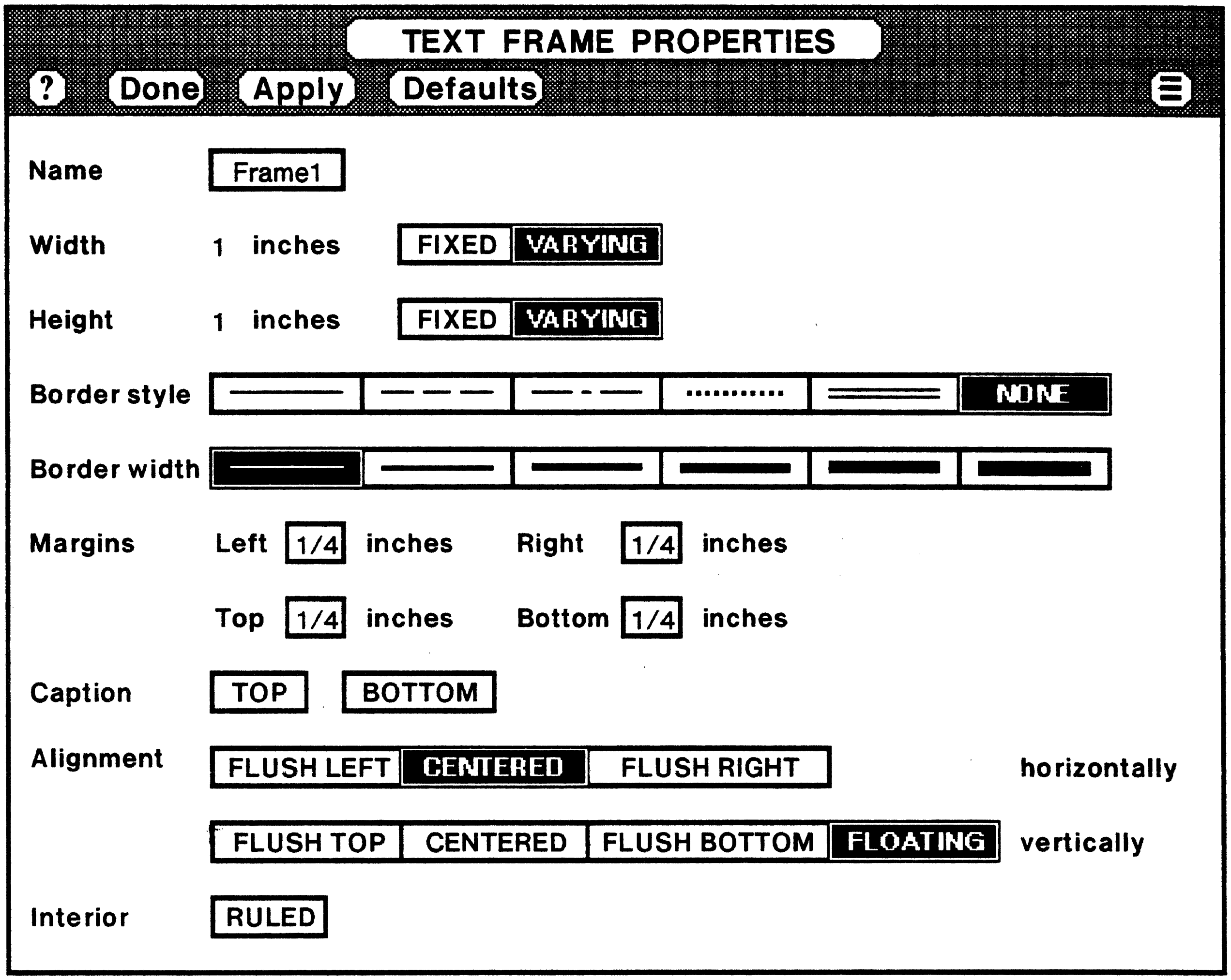


Figure FRA-2 -- The structure of a frame

The captions are optional. The margins may be zero.



TEXT FRAME PROPERTIES

Name: Frame1

Width 1 inches FIXED [✓ VARYING]

Height 1 inches FIXED [✓ VARYING]

Border style ——— | — — — | — - — | .... | ═══ | [✓ NONE]

Border width [✓ ━] | — | ━ | ━ | ━ | ━

Margins Left 1/4 inches Right 1/4 inches

Top 1/4 inches Bottom 1/4 inches

Caption TOP BOTTOM

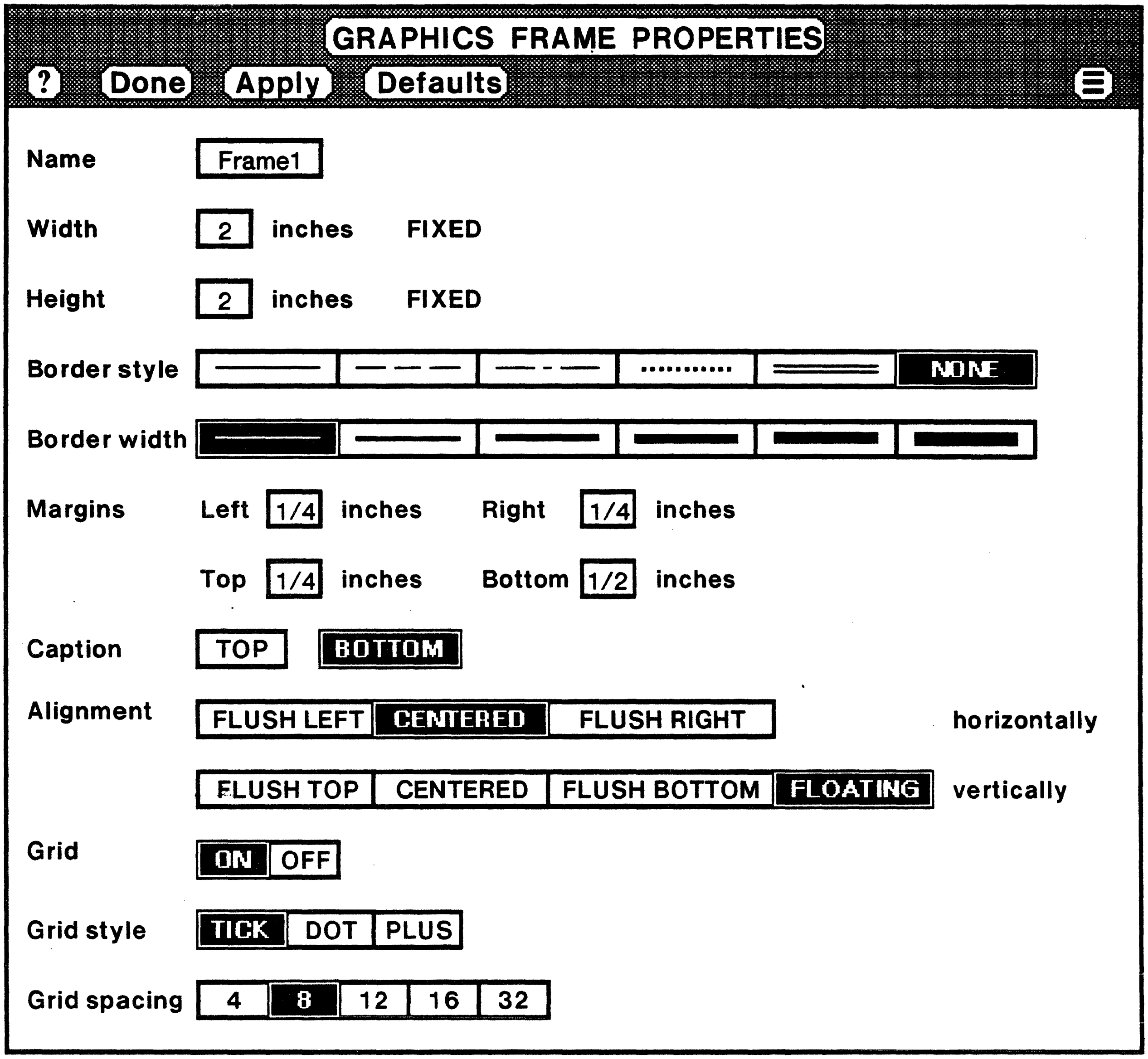
Alignment FLUSH LEFT [✓ CENTERED] FLUSH RIGHT horizontally

FLUSH TOP CENTERED FLUSH BOTTOM [✓ FLOATING] vertically

Interior RULED

Figure FRA-3 -- Text Frame property sheet

The settings shown here are the defaults.
The Width and Height parameters are modifiable only if they are FIXED.
The Border appearance choices are the same as for graphic rectangles.



Name Frame1

Width 2 inches FIXED

Height 2 inches FIXED

Border style ——— | - - - - | — - — | .... | ═══ | [✓ NONE]

Border width [✓ ━] | ━ | ━ | ━ | ━ | ━

Margins Left 1/4 inches Right 1/4 inches

Top 1/4 inches Bottom 1/2 inches

Caption TOP [✓ BOTTOM]

Alignment FLUSH LEFT [✓ CENTERED] FLUSH RIGHT horizontally

FLUSH TOP CENTERED FLUSH BOTTOM [✓ FLOATING] vertically

Grid [✓ ON] OFF

Grid style [✓ TICK] DOT PLUS

Grid spacing 4 [✓ 8] 12 16 32

Figure FRA-4 -- Graphics Frame property sheet

The settings shown here are the defaults.
The Width and Height parameters are modifiable only if they are FIXED.
The Border appearance choices are the same as for graphic rectangles.

12. TABLES

Star's table facility lets users enter tabular statistical and/or textual data easily. It automates much of the conventional formatting, freeing users from the tedium of tab setting. Simple tables can be constructed by users with little specialized training. [The Tables facility is not included in the first release of Star-1.]

USER OVERVIEW

A *table* is one of the special objects provided in Star to facilitate document creation. It enables users to create tables of any desired size, to fill them in, and to change them, all without using tabs. The user enters data in individual *fields* in a table. Within a field, ordinary text editing operations apply. The NEXT key is used to move between entries. As a user enters or modifies the contents of an entry, Star automatically adjusts the sizes of columns and rows to keep them aligned. A user may "fix" a column at its current width, providing automatic carriage return for textual information. Lines are automatically drawn between the rows and columns of a table; a user may alter individual lines to achieve a highly tailored appearance.

Every table has *property sheets*, which provide a complete description of the table. For the majority of users, most of these parameters may be treated as informative only -- a user need deal with only a few basic properties. Star provides ways to do the common types of table manipulation directly, without using the property sheet. A user may also have a number of customized, "predefined" tables with appropriate property settings, which he includes in his documents via abbreviation expansion or by copying from some other source such as a graphics "transfer sheet."

Through table properties a user can vary the behavior and appearance of a table in several ways. He can:

choose to fill in the table by column or by row,

control the alignment of the contents of rows and columns,

make rows and columns fixed in size, or allowed to grow automatically,

change the spacing between rows and between columns,

cause lines to be drawn between rows and between columns,

create headings to be centered over each column,

specify constraints on column values, and

automatically compute column values.

DETAILED SPECIFICATION

OBJECTS

Table

A *table* is a type of frame consisting of a rectangular array of fields with ruling lines between rows and columns and around the table as a whole (see Figures TAB-1, TAB-9 and TAB-11). A table as a whole has the same capabilities and characteristics as other frames (see *Frames*). It may be placed in text (including text in a text frame), whereupon it becomes "anchored" to a text character. Or it may be placed in a graphics frame along with any number of graphic objects such as lines, shaded rectangles, or other frames. The latter enables users to employ Star's graphics facilities to obtain sophisticated table appearances. Tables are opaque objects, like other frames, and obscure anything "underneath" them. Editing of table entries does not change the table's "level" in a graphics frame; only manipulations of the table as a whole do so (e.g. Move; see *Graphics*).

Unless frozen by the user, table rows and columns automatically grow or shrink to accommodate the data entered. Row/column alignment and spacing are automatically preserved. The table as a whole grows or shrinks to accommodate its rows and columns, if possible (see "Change Contents of Table Entry" below). The smallest possible table is a 1x1 single-entry table.

During pagination, Star attempts to keep an entire table (in fact, any frame) on one page. If the table is longer than one page, Star puts the entire table on a single "long" page. When this page is printed on paper, Star breaks the table between rows if possible, optionally reproducing the column headings at the top of each new page. The generated pages have the correct headings, footings and page numbers. Thus if a three-page table occurs on page 15, on the display the pages will seem to skip from 15 to 18, while on printed paper the table will appear on pages 15, 16 and 17. Tables that are not an integral number of pages long are rounded up to a page boundary during printing.

A table has a border just like any other frame (see *Frames*).

Table Entry

A *table entry* is a field which is part of a table. Each table entry is associated with a specific row and column in the table. A table entry is similar to an ordinary field as described in *Field Definition*. They have the same legal contents: one or more paragraphs; anything permitted in a document is permitted inside a table entry. They can have the same constraints on their values; for example, numeric values only. Thus table entries can contain text characters, formatting characters, fields, frames, footnotes and equations. Footnotes are placed at the bottom of the page containing the reference, as always. Page Format Characters are ignored.

Table Row

A table *row* is a constrained horizontal arrangement of table entries. Row alignment and spacing are automatically maintained. Rows can be selected, providing a convenient handle for performing common table manipulations. Rows have properties, which can be changed individually.

Table Column

A table *column* is a constrained vertical arrangement of table entries. Column alignment and spacing are automatically maintained. Columns can be selected, providing a convenient handle for performing common table manipulations. Columns have properties, which can be individually changed. Columns provide the means of assigning field properties to table entries; all entries in a column (except its heading) have exactly the same field properties.

Tables can be given additional structure by dividing columns into sub-columns and sub-rows. Only columns can be divided, not rows.

A column may have zero or more sub-columns. Each sub-column adds a new column header (see Figure TAB-11); this permits columns to be grouped together into logical units. Divided columns can themselves be divided. Dividing can go on indefinitely; there is no predefined limit.

A divided column may have SINGLE or REPEATING sub-rows. In the SINGLE case, the column has exactly one entry per containing row. This is the normal situation. (A "containing row" is defined to be either an outer-level row in the table as a whole or, if the column is part of a subdivided column, the containing row is an outer-level row of the subdivided column itself, treated as a table.) In the REPEATING case, the column may have several entries per containing row (see Figure TAB-11). Each entry in a REPEATING column is like a list of sub-rows. Such structure is typical of record files; it allows rows to mirror record structures accurately (see *Records Processing*).

Column Headings

A table can have a "header row" composed of the labeling for the columns (see the "State" row in Figure TAB-1). The presence or absence of column headings is controlled through the Table property sheet, as are their properties. The header row cannot be selected as a unit. If the document containing the table is "locked" (see *Field Definition*), the column headings cannot be edited; otherwise the text of the headings may be edited in the normal manner. The NEXT and SKIP keys operate differently on the header row than other rows, as described later.

Table Ruling Line

A table *ruling line* is a graphic line which can be used to change the appearance of a table. It's appearance options are similar to those of graphic line transfer symbols (see *Graphics*), including being invisible. All the lines between rows, between columns, and/or around the table may be made visible or invisible in a single step, or their appearance may be altered individually. (Note: invisible ruling lines are still present in a table, just as invisible rectangle borders are still present in a rectangle.) All visible ruling lines are shown on both the display and hardcopy. All invisible ruling lines are displayed visibly (dotted) when Show BOUNDARIES is turned on in the Window option sheet. An invisible ruling line is also displayed dotted if the cursor is near it while a mouse button is down -- "near" meaning within the selective range of graphic lines.

Table Property Sheets

A number of property sheets are used in conjunction with tables.

Table Property Sheet (Figure TAB-2)

The Table property sheet controls a few overall characteristics of a table: the number of rows and columns, whether the number can increase during typing, and the fill-in order. It has a Display parameter which provides access to other table properties.

> **Number of rows** (text) - This controls the number of rows in a table. It may be set to any positive integer, subject to the restrictions described below. It indicates the number of rows at the outermost level of the table; sub-rows that appear in REPEATING columns are not reflected in this parameter. If a user increases or decreases the Number of Rows, rows are added to or removed from the bottom of the table. Properties of added rows are taken from the previous last row in the table. If its Height is VARYING, new rows are allocated 1/16 inch heights (plus 1/16 inch top and bottom margins). A user may also change the number of rows by adding, moving, copying or deleting rows directly. Such direct changes update the Number of Rows parameter if the affected rows are at the outermost level of the table.
>
> The number of rows may be increased arbitrarily. If there is not enough room for all of the rows in the table frame, information is clipped (retained but not displayed), and a warning message is issued. The number of rows may not be decreased in a way that would discard non-empty rows. If the user attempts to make such a change, the system displays an error message and resets the parameter to the value it had previously.
>
> **Number of columns** (text) - This controls the number of columns in a table. It operates similarly to the Number of Rows parameter. It indicates the number of columns at the outermost level of the table; sub-columns in DIVIDED columns are not reflected in this parameter. (They are reflected in each DIVIDED column's Number of Subcolumns parameter.)
>
> **Number:** FIXED/VARYING (choice) - This controls whether an additional row or column is added when a user invokes NEXT with the caret in the last entry in the table -- in the lower right corner. It is modifiable for the Number of Rows parameter if the fill-in order is by row, otherwise for the Number of Columns parameter. If FIXED is chosen, no additional entries are added by NEXT; Star simply proceeds to the next

field following the table in the fill-in sequence. If VARYING is chosen and the fill-in order is by row (column), a new row (column) is added at the bottom (right) of the table, and the caret is placed in its first entry. Note: This parameter does not affect the Move/Copy/Delete Row/Column operations; it affects only the NEXT key.

Fill in by: ROW/COLUMN (choice) - This enables a user to specify the fill-in order. The fill-in order determines the action of the NEXT and SKIP keys in stepping through a table, as described in a subsequent paragraph.

Table Frame Property Sheet (Figure TAB-3; see also Figure TAB-9)

The Table Frame property sheet controls the size, alignment and appearance of the rectangular area for a table. Except for the omission of a RULED option, table frames have exactly the same properties as text frames (see *Frames*). Only one property needs to be mentioned here:

Name (text) - All tables are named. The name cannot be empty. The default table name is "Table#", where # is a positive integer representing the table's chronological order in the document. Users may change the name, but table names must adhere to the same restrictions as field names (see *Field Definition*). In particular, they must be unique in the document.

As with other frames, if a table is embedded in a graphics frame, the table's Alignment property disappears.

Row Property Sheet (Figure TAB-4)

The Row property sheet controls the formatting of the row contents. Normally all rows have the same properties.

Contents: FLUSH TOP/CENTERED/FLUSH BOTTOM (choice) - This controls the alignment of the contents of the row.

Height: size (text) - This displays the current height of the row, expressed in the current units. A user may change it only if the parameter below is set to FIXED.

Height: FIXED/VARYING (choice) - This controls whether the row's height automatically changes during typing. If FIXED is chosen, it does not change. If VARYING is chosen, the row automatically assumes the minimum height necessary to hold its contents.

Margins: Top and Bottom (text) - These two parameters control the top and bottom "margins" in the row. Every table entry is like a miniature page in that it has edges (the ruling lines) and top, bottom, left and right margins inset from those edges. (The left and right margins are supplied by columns.) These margins act exactly like page margins.

Column Property Sheet (Figures TAB-6 and TAB-7)

The Column property sheet controls the formatting of the column contents. In addition it has field properties which define the legal column contents.

Name (text) - All columns in a table are named. The name cannot be empty. The default name is "Column# ", where # is a table- or column-relative positive integer. Users may change the name. Each column's name is appended to the name of the table to form a full "inherited" name: "<table name>.<column name>", for example "Table1.Column1". Full column names must adhere to the same restrictions as field names (see *Field Definition*). In particular, they must be unique in the document. Sub-columns in a divided column inherit the name of the containing column, just as columns inherit the table name: "<table name>.<column name>.<sub-column name>". This continues through all the levels of nesting. For example, the column in Figure TAB-11 whose heading is labeled "Year" might be named "Table1.Column2.Column3.Column2". Note: The text in column headings, if any, is completely independent from the column names. Editing the headings does not affect the names, and vice versa.

Structure: DIVIDED, **Number of Subcolumns, and** SINGLE/REPEATING (state, text and choice) - These parameters control whether the column is divided into sub-columns and sub-rows.

If DIVIDED is on, the column may contain sub-columns and sub-rows. In that case most of the other column properties disappear (see Figure TAB-7); the user sets those individually for each sub-column. DIVIDED may be turned off only if there is no more than one non-empty sub-column in the column; otherwise information would be lost. The column adopts the properties and contents of the non-empty sub-column, if any; otherwise, the properties of the leftmost sub-column are assumed by the column.

The Number of Subcolumns text parameter appears only if DIVIDED is on. It serves exactly the same function for the divided column that the Number of Columns parameter does for the table as a whole.

SINGLE/REPEATING appears only if DIVIDED is on. If REPEATING is on, each entry in the column may contain sub-rows. SINGLE may be turned on only if there is no more than one non-empty sub-row in each entry in the column. In such a case, each entry in the now-SINGLE row adopts the contents of the non-empty sub-row, if any.

Contents: FLUSH LEFT/CENTERED/FLUSH RIGHT/DECIMAL ALIGNED (choice) - This controls the alignment of the contents of the column. Alignment is handled as described for tabs in *Formatting and Layout*, except that the column width may expand to accommodate long entries. For decimal-aligned columns, the entry is:

- aligned on the first decimal point (comma for European installations) following or preceding a numeric character. If there is no decimal point, it is implied after the last numeric character. Spaces are ignored (in the U.S.) in determining the alignment point.

- centered if the entry contains no numeric characters.

Setting this parameter changes the alignment of all paragraphs in the column. A user may alter the alignment of individual paragraphs by selecting them and modifying their properties in the normal way. However any subsequent change to this parameter will override all such individual changes.

Width: size (text) - This displays the current width of the column, expressed in the current units. A user may change it only if the parameter below is set to FIXED.

Width: FIXED/VARYING (choice) - This controls whether the column's width automatically changes during typing. If FIXED is chosen, it does not change. (This enables columns containing paragraphs of text to force lines to break.) If VARYING is chosen, the column automatically assumes the minimum width necessary to hold its contents.

Margins: Left and Right (text) - These two parameters control the left and right "margins" in the column. Every table entry is like a miniature page in that it has edges (the ruling lines) and top, bottom, left and right margins inset from those edges. (The top and bottom margins are supplied by rows.) These margins act exactly like page margins.

The other column properties are the same as in the Field property sheet (see *Field Definition*). As with fields, columns have a Display option called FIELD SUMMARY which enables the Field Summary sheet to be displayed (see *Field Definition*).

Column Headings Property Sheet (Figure TAB-5)

The Column Headings property sheet controls the formatting of the headings at the top of the columns. In addition to the normal row properties, this "header row" may be visible or invisible, and it may be replicated at the top of every printed page.

Visibility: SHOW (state) - This controls the visibility of column headings. If it is on, the header row is displayed at the top of the table. The contents of the header row, if any, are preserved when SHOW is turned off and restored when SHOW is turned on.

Visibility: REPEAT EVERY PAGE (state) - This appears only if SHOW is on. If it is on, the header row is printed at the top of every page in a multi-page table. If it is off, the header row is printed only at the beginning of the table.

The other properties of column headings are the same as for other rows, with the exception that the alignment has both a horizontal and a vertical component.

Ruling Line Property Sheet (Figure TAB-8)

The Ruling Line property sheet controls the appearance of a ruling line. The Style and Width options are the same as for graphic line transfer symbols (see *Graphics*), but there are no Line End or Constraint options.

Table Entry Property Sheet

An individual table entry has no properties. Its size and formatting are controlled by the row and column to which it belongs. Its field properties are set on a column-wide basis. An individual table entry itself cannot be selected, but its contents can be selected and edited. A table entry can contain one or more paragraphs. These paragraphs may have individual character, paragraph and tab properties as described in *Formatting and Layout*. In particular note that paragraph indentation is added to the left and right margins supplied by the column.

ACTIONS

Create Table

A user can create a table by inserting a table frame into a document using the KEYBOARD key. Two distinct options appear in the Special keyboard set, one to insert a table filled in by row, and the other to insert a table filled in by column. After insertion the entire table is selected, making it convenient to invoke Show Properties so that the number of rows and columns can be changed. The default property settings of these tables are as follows.

Table as a whole (see Figures TAB-2 and TAB-3):

default name
2 rows by 2 columns, plus a header row for column headings
number of rows (if fill-in-by-row) or columns (if fill-in-by-column) is varying
fill-in order according to the option chosen from the Special keyboard set
table 2" by 2" in size, plus 1/4" left, right and top margins and 1/2" bottom margin
a bottom caption area
varying width and height
centered horizontally, floating vertically in containing text
border invisible

Rows (see Figure TAB-4):

flush top contents
1/16" minimum height, varying
1/16" top and bottom margins

Columns (see Figure TAB-6):

default name
no description
no sub-columns or sub-rows
decimal aligned contents
1/16" minimum width, varying
1/16" left and right margins
contents of ANY type
not required to be filled in
no format, range or length restrictions
never skipped
no fill-in rule

Column headings (see Figure TAB-5):

displayed
repeated at the top of every page
contents centered horizontally and vertically
1/8" minimum height, varying
1/8" top and bottom margins (columns provide the left and right margins)

Ruling lines (see Figure TAB-8):

solid style
unit width

These settings are suited for statistical information. After changing the properties, the inserted table will still be selected. If the user invokes NEXT, the caret is placed in the first table entry, in the upper left corner (the first column heading if column headings are displayed). Or the user may place the caret in any other entry directly with the mouse. All entries are initially empty.

Tables may also be created by copying them from other sources, e.g. transfer sheets. Or they may be predefined with different properties and/or contents and inserted into a document via abbreviation expansion.

Tables, like other frames, may be inserted either into text (including text in text frames) or into graphics frames.

Create Row

There are several ways to create a new row in a table:

(1) A user may increase the Number of Rows property.

(2) He may move or copy a row from another table, or copy a row in the same table (described in "Move Row" and "Copy Row" below).

(3) He may place the caret in the last table entry -- in the lower right corner -- and invoke NEXT. A blank top-level row is created, and the caret is placed in its first entry. The properties of the new row are taken from the top-level row above. This works only if (a) the table is being filled in by row, (b) the Number of Rows is VARYING, (c) a top-level row is to be created, and (d) the new row is to go at the bottom of the table.

(4) But the simplest way is just to select a row and begin typing. A new row is appended below the selected row, and the typed text is inserted in the first entry of the new row. This is true whether the selected row is at the top level of the table or is a sub-row of another row. If it is a top-level row, the Number of Rows parameter (which must be VARYING) is increased by one. The properties of the new row are taken from the row above. Note that a row cannot be inserted at the top of a table or set of sub-rows in this way; the row must be inserted elsewhere and moved to the top.

If there is insufficient room for the new row, information is clipped (retained but not displayed), and a warning message is issued.

Create Column

As with rows, there are several ways to create a new column in a table:

(1) A user may increase the Number of Columns property. In the case of subcolumns, he may increase the appropriate column's Number of Subcolumns property.

(2) He may move or copy a column from another table, or copy a column in the same table (described in "Move Column" and "Copy Column" below).

(3) He may place the caret in the last table entry -- in the lower right corner -- and invoke NEXT. A blank top-level column is created, and the caret is placed in its first entry. The properties of the new column are taken from the top-level column to its left. This works only if (a) the table is being filled in by column, (b) the Number of Columns is VARYING, (c) a top-level column is to be created, and (d) the new column is to go at the right end of the table.

(4) But the simplest way is just to select a column and begin typing. A new column is appended to the right of the selected column, and the typed text is inserted in the first entry of the new column (in the column heading if column headings are being displayed). This is true whether the selected column is at the top level of the table or is a sub-column of another column. If it is a top-level column, the Number of Columns parameter (which must be VARYING) is increased by one; otherwise the Number of Subcolumns parameter is increased for the containing column. The properties of the new column are taken from the column to its left. Note that a

column cannot be inserted at the left end of a table or set of sub-columns in this way; the column must be inserted elsewhere and moved to the left.

If there is insufficient room for the new column, information is clipped (retained but not displayed), and a warning message is issued.

Select Table (see Figure TAB-10)

A user can select a table like any other frame, by clicking SELECT near one of its border lines, as described in *Frames.* Alternatively a user can select a table by clicking SELECT over any selected row or column which is at the outermost level of the table (i.e. is not a sub-row or sub-column of some other row or column). When a table is selected, its entire rectangular area is inverted. When a table is selected by itself or as part of a graphics selection, the graphics "soft keys" are displayed (see *Graphics*).

Select Row (see Figure TAB-10)

A user can select a row by first selecting anything in the row and then invoking the Select Row command in the document auxiliary menu. The smallest row containing the selection/caret is selected. If the selected row is a sub-row of a larger containing row, a click of the SELECT button anywhere in the selected row will expand the selection to the containing row. If SELECT is clicked again, the selection expands to the next larger containing row, if any. And so on. if there is no containing row, clicking SELECT in a selected row selects the entire table. Another click at this point will wrap back around to a character in a table entry. Thus successive clicks of SELECT in a selected row expands the selection up through the row hierarchy exactly as it does in the character-word-sentence-paragraph hierarchy.

Multiple rows can be selected. A user selects a row, then places the cursor inside the last row to be selected and clicks the ADJUST button. All the rows between the selected row and the cursor location become selected. The selection may be extended in either direction. However it may not be extended across table boundaries or out of one set of sub-rows and into another.

Select Column (see Figure TAB-10)

A column is selected in an analogous fashion using the Select Column command in the document auxiliary menu. Mulitiple clicking advances through the column hierarchy just as with rows. Multiple columns can be selected with the ADJUST button.

Select Ruling Line (see Figure TAB-10)

A user can select a ruling line like any other graphic line, by clicking SELECT anywhere along its trajectory. He can select multiple ruling lines by selecting one and extending to the last one, as with multiple row or column selections.

Select Contents of Table Entry

A user can select the contents of a table entry just as he does the contents of an ordinary field (see *Field Fill-In*). In most cases, it is exactly like selecting text in a document. In the case of a computed entry, the contents are selectable but not editable. A table entry itself (the field) cannot be selected, just its contents.

NEXT and SKIP in a fill-in-by-row table

The NEXT and SKIP keys provide alternate ways to select the contents of table entries without using the mouse. When the selection or caret is in a table, the NEXT and SKIP keys step through the table entries selecting the contents of non-empty entries and placing a caret in empty entries. NEXT in a fill-in-by-row table proceeds through the rows in left-to-right top-to-bottom order. If a column has sub-rows, NEXT activates all the sub-row entries in left-to-right top-to-bottom order before proceeding to the following entry. If the caret is in the last table entry, in the lower right corner, NEXT *adds a new row* and places the caret in its first entry (see "Create Row" below).

If an entry is computed (i.e. its column has a fill-in rule), NEXT evaluates the fill-in rule causing a value to appear in the entry, then automatically proceeds to the following entry. This entry may also be computed, causing several entries in succession to be evaluated, before the selection finally stops on a non-computed entry.

If the current selection is a row, NEXT activates the first entry in the following row. If there is no following row at the selected row's level, NEXT activates the next entry in the containing row. If there are no more entries in the containing row, NEXT proceeds as if the containing row itself were selected. If there is neither a following row nor a containing row, then the last row in the table must be selected. In this case, NEXT simply proceeds to the next field in the document's fill-in order. (A detailed example of the operation of NEXT and SKIP is given in Figure TAB-12.)

If the current selection is a column, NEXT activates the entry after the top entry in the column.

If the current selection is a table, NEXT activates the first entry in the table, in the upper left corner (the first column heading if column headings are displayed).

The operation of the SKIP key is similar to a Select Row command followed by a NEXT. SKIP evaluates all computed table entries skipped over. SKIP when a table is selected evaluates all the computed entries in the table and then proceeds to the next field in the document's fill-in order. Note that SKIP never adds a new row or column.

NEXT and SKIP in a fill-in-by-column table

In a fill-in-by-column table, NEXT and SKIP proceed in a corresponding way down each column and sub-column. Note that if a column has a fill-in rule, NEXT will evaluate *every* entry in it and then will proceed to the top of the following column.

NEXT and SKIP in column headings

Column headings, if present, are treated differently by NEXT and SKIP. NEXT activates a column heading *only if the caret is already in one of them or if the entire table is selected.* NEXT and SKIP never go from an ordinary table entry into a heading, nor from any other field in a document into a heading. Furthermore, column headings (the "header row") are always filled in by row, regardless of whether the table itself is filled in by row or by column, and thus column headings are usually defined at the start of the table fill-in process. Within the header row, NEXT and SKIP operate as in other rows. For example, if the caret is in the Name entry in the table in Figure TAB-11, NEXT activates headings in the following order: Children, Child Name, Degrees, Degree, Age, Schools, School Name, Year, King, etc. SKIP skips out of the header row to the King entry.

Change Contents of Table Entry

Once a user has created a table, he may place the insertion caret in any entry and type into it. Normal text editing operations apply. The user need not worry about row or column size; the sizes are automatically adjusted as necessary. After filling in an entry, a user may push the NEXT key to activate the next entry in the row or column. Thus a table can be filled in without touching the mouse.

For varying-width columns, automatic carriage returns are not supplied. Instead, each character that the user types is placed to the right of the preceding character until the contents becomes as wide as the widest entry in the column. After that, each character causes the entire column to become wider. This may affect the positions of other columns in the table; the table grows as necessary. The user can type New Line characters to manually control the width of a column. He may also type New Line characters to create multi-line entries. If this makes an entry taller than any other entry in its row, the entire row becomes taller. This may affect the positions of other rows in the table; again the table grows as necessary. The user may freeze row heights and column widths at desired values as described below under "Fix/Vary Row Height" and "Fix/Vary Column Width."

If a column's width cannot be increased (it is FIXED), the system attempts to break the line and make the row taller. If a row's height cannot be increased (it is FIXED), the system attempts to make the column wider. If neither can be done, the system issues a warning message; further typing causes information to be clipped (retained but not displayed) until the user manually makes the entry larger.

Changes to an entry may be rejected if they violate any of the entry's Range, Format or Length restrictions. Table entry validation is performed exactly like other field validation (see *Field Fill-In*).

Copy Row

A user can copy one or more rows by selecting them and pushing the COPY key. The cursor changes to a right-pointing double arrow to indicate that a copy is in progress (see Figure HOV-4). The user then points between two rows (or above the first or below the last) and clicks SELECT. The new rows are positioned where he pointed. If the destination is ambiguous, for example between two sub-rows in different containing rows, appropriate feedback is provided to indicate and control the meaning. The Number of Rows parameter is increased if the copied rows were placed at the top level. If there is insufficient room for the new rows (see "Create Row" above), the command is aborted with an error message.

Rows can be copied to other tables. The destination must have exactly the same column structure or the operation will be rejected. The column constraints on the entries must also be compatible. The width, margins and alignment of the entries in the copied rows will change to match those of the destination table. Their field properties will also change if necessary. If a column is computed, copied entries in that column are left unchanged, but the document is marked invalid until the Update Fields command is executed (see *Field Fill-In*).

Copy Column

A user can copy one or more columns by selecting them and pushing the COPY key. The cursor changes to an upward-pointing double arrow to indicate that a copy is in progress (see Figure HOV-4). The user then points between two columns (or to the left of the first or right of the last) and clicks SELECT. The new columns are positioned where he pointed. If the destination is ambiguous, for example between two sub-columns in different containing columns, appropriate feedback is provided to indicate and control the meaning. The Number of Columns parameter is increased if the copied columns were placed at the top level; otherwise the Number of Subcolumns parameter is increased for the containing column. If there is insufficient room for the new columns (see "Create Column" above), the command is aborted with an error message.

Columns can be copied to other tables. The destination must have exactly the same row structure or the operation will be rejected. The height, margins and alignment of the entries in the copied columns will change to match those of the destination table.

Copying columns between tables is a little trickier than rows because columns are named. Any column name that is not unique in its new location (i.e. whose full inherited name is not unique in the document) is reset to its default name, and a warning message is issued. Any fill-in rule in a copied column which referenced columns in the old table will continue to access them if the old table is still in the same document, which may not be the desired result. For this reason a warning message is also issued whenever a computed column is copied to another table. The user may have to manually correct these names and fill-in rules.

Move Row or Column

A user can move one or more rows or columns by selecting them and pushing the MOVE key. This works exactly like the Copy Row and Copy Column commands except that the rows or columns are deleted from their original location. Rows can be moved between tables, which will change the Number of Rows parameters of both the source and destination tables if the moved rows are at the top level. Moving columns between tables changes the Number of

Columns/Subcolumns parameters of both the source and destination tables.

Delete Row or Column

A user can delete one or more rows or columns by selecting them and pushing the DELETE key. The Number of Rows (Columns) parameter is reduced if the rows (columns) were at the top level. Deleting subcolumns reduces the appropriate Number of Subcolumns parameter. If the last remaining row, sub-row, column or sub-column is deleted from a table, it is replaced by an equivalent empty structure. For example, if all the rows in a table are deleted, a single empty row would remain. If then all the columns are deleted, a 1x1 single-entry table would remain (plus possibly a heading entry). This is the minimum possible table.

Move/Copy/Delete Table

A user can move, copy or delete an entire table by selecting it and pushing the MOVE, COPY or DELETE key. These work for tables exactly as for other frames (see *Frames*).

Move/Copy/Delete Ruling Line

Ruling lines cannot be moved, copied or deleted.

Show/Change Table Properties

A user can set the properties of a table by selecting it and pressing the PROPERTIES key. This displays the Table property sheet. The Table Frame and Column Headings property sheets are displayed by linking to them through the Display parameter.

Show/Change Row Properties

A user can set the properties of a row by selecting it and pressing the PROPERTIES key. This displays the Row property sheet. If more than one row is selected, the properties of the top row are displayed.

Show/Change Column Properties

A user can set the properties of a column by selecting it and pressing the PROPERTIES key. This displays the Column property sheet. If more than one column is selected, the properties of the leftmost column are displayed. In addition to selecting and changing individual columns, column properties may be set through the Column Summary property sheet. This property sheet is displayed by linking to it through the Display parameter in the Table or Column property sheet.

Show/Change Column Headings Properties

A user can alter the properties of the column headings by linking to the Column Headings property sheet via the Display parameter in the Table property sheet.

Show/Change Ruling Line Properties

A user can alter the appearance of a ruling line in a table by selecting it and pressing the PROPERTIES key. He can then set it to any of a variety of styles and widths including invisible and double-line. He makes a line invisible by setting its Style to NONE. Invisible ruling lines appear dotted when Show BOUNDARIES is turned on in the Window option sheet (see Figure TAB-9).

Fix/Vary Table Size

A user can "fix" the size of a table by setting its Width and/or Height property to FIXED. The table will then remain at its current size, neither growing nor shrinking as its rows and columns change. The rows and columns will have to apportion themselves according to the allocated space. This may cause some table edits (e.g. Create Row/Column) to clip information because there is not enough room to display it. When a table has a fixed size, the user may return it to a varying size by setting its Width and/or Height property to VARYING.

Fix/Vary Row Height

A user can "fix" the height of a row by setting its Height property to FIXED. The row will then remain at its current height, neither growing nor shrinking as its entries are edited. When a row has a fixed height, the user may return it to a varying height by setting its Height property to VARYING.

If both row height and column width are FIXED, then a table entry cannot grow. Characters that would require it to grow are accepted (with an audible tone and a warning message on each successive character) and retained but are not displayed until the user makes the entry larger or relaxes the size constraints.

Fix/Vary Column Width

A user can "fix" the width of a column by setting its Width property to FIXED. The column will then remain at its current width, neither growing nor shrinking as its entries are edited. When a column has a fixed width, the user may return it to a varying width by setting its Width property to VARYING.

Change Table Size

A user can change the size of a table by selecting it and altering its Width or Height text property, if its width or height is FIXED. He may also invoke the Stretch command (the STRETCH soft key) or the Magnify command (the MAGNIFY soft key) to manually change the size. The Stretch and Magnify commands work as in graphics. They set the table's width and/or height to FIXED, if they were not already so, depending on what was changed.

Stretching and magnifying alter the size of the area inside the table's border; they do not change its margins. This may cause the table's rows and columns to reformat themselves, if their widths and/or heights are VARYING. A table may become too small to hold all of its contents. In that case, information is clipped (retained but not displayed), and a warning message is issued.

If a user wishes to change a table's size while leaving its width and/or height VARYING, he may edit its rows or columns or change its margins.

Change Row Height

If the height of a row is FIXED, a user can change its height by altering its Height text property. If the height of a row is VARYING, a user can change its height by altering its contents, for example by inserting or deleting carriage returns.

Change Column Width

If the width of a column is FIXED, a user can change its width by altering its Width text property. If the width of a column is VARYING, a user can change its width by altering its contents, for example by inserting or deleting carriage returns.

As an optimization, the widths of all columns in a table can be made equal by selecting anything in the table and invoking the Set Column Widths Equal command in the document auxiliary menu. This command FIXES all column widths and sets them all to the current width of the table divided by the number of columns. The effect can be understood by imagining that ruling lines are visible between every column and sub-column in the table; the distance between all adjacent ruling lines becomes the same. (A user can subsequently alter individual column widths in the normal way, including changing them back to VARYING.)

Create Sub-rows

A use can partition a DIVIDED column into sub-rows by selecting it and turning on its REPEATING Subrows property. This causes each entry in the column to contain one sub-row. These sub-rows initially have default properties, which the user may change. The user can add additional sub-rows by the Create Row operation. Sub-rows behave in most respects like other rows.

Only columns whose DIVIDED property is on can have sub-rows. Note that this means that the header of any column which can contain sub-rows will be divided (see Figure TAB-11).

Create Sub-columns

A use can partition a column into sub-columns by selecting it and turning on its DIVIDED Structure property. This causes the column to contain one sub-column and its Number of Subcolumns property to be set to one. Alternatively he can select the column and invoke the Divide Column command in the document auxiliary menu. This causes the column to be split into two sub-columns, its DIVIDED property to be turned on, and its Number of Subcolumns property to be set to two. These sub-columns initially have default properties and an empty header, which the user may change. The user can add additional sub-columns with the Create Column operation. Sub-columns behave in most respects like other columns.

Move/Copy/Delete Sub-row or Sub-column

Sub-rows and sub-columns are selected, moved, copied and deleted like other rows and columns.

Copy Properties of Table, Row, Column, Column Headings, Ruling Line

A user can copy properties between tables, rows, columns, column headings, or ruling lines. He selects a table or part of a table, presses the SAME key, and designates with the SELECT button a source from which to copy the properties. The corresponding structure in the source is highlighted while the mouse button is held down. For example, if a row or rows are selected when Copy Properties is invoked, a row in the source is highlighted. As the user moves the cursor around in the source. different rows will highlight, always the smallest row containing the cursor. When the user releases the mouse button, the command is carried out.

The effect is as tollows if the source and destination are:

Tables - copies all property settings from the source's Table and Table Frame property sheets except its name and its number of rows and columns.

Rows - copies all property settings from the source's Row property sheet.

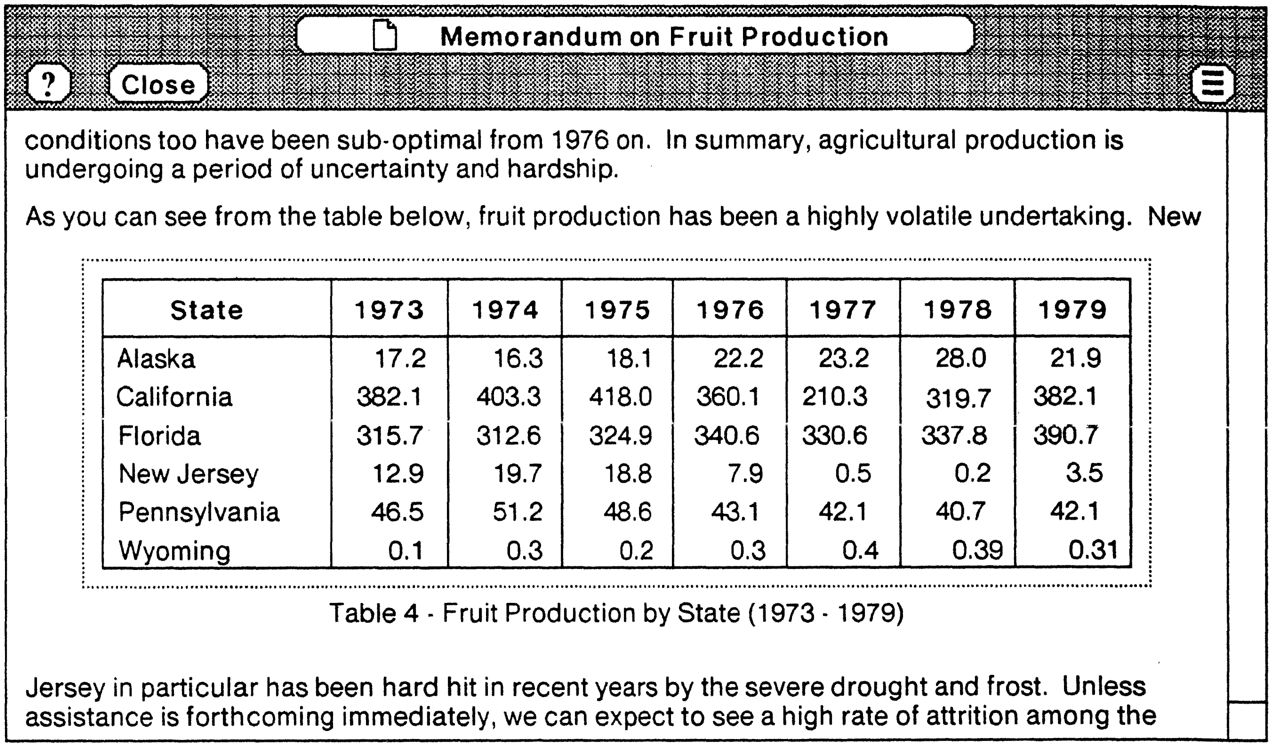
Columns - copies all property settings from the source's Column property sheet.

Column Headings - copies all property settings from the source's Column Headings property sheet.

Ruling Lines - copies all property settings from the source's Line property sheet. Properties can be copied between ruling lines in the same table or different tables or between ruling lines and graphic line transfer symbols.

auxiliary commands used for tables

...
Divide Column
Select Column
Select Row
Set Column Widths Equal
...



conditions too have been sub-optimal from 1976 on. In summary, agricultural production is undergoing a period of uncertainty and hardship.

As you can see from the table below, fruit production has been a highly volatile undertaking. New

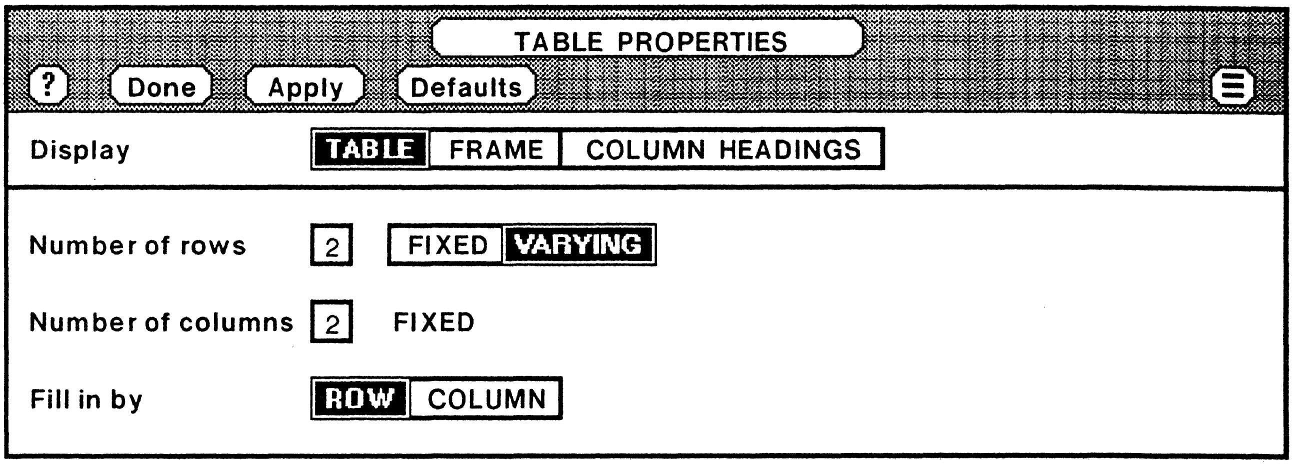
State	1973	1974	1975	1976	1977	1978	1979
Alaska	17.2	16.3	18.1	22.2	23.2	28.0	21.9
California	382.1	403.3	418.0	360.1	210.3	319.7	382.1
Florida	315.7	312.6	324.9	340.6	330.6	337.8	390.7
New Jersey	12.9	19.7	18.8	7.9	0.5	0.2	3.5
Pennsylvania	46.5	51.2	48.6	43.1	42.1	40.7	42.1
Wyoming	0.1	0.3	0.2	0.3	0.4	0.39	0.31

Table 4 - Fruit Production by State (1973 - 1979)

Jersey in particular has been hard hit in recent years by the severe drought and frost. Unless assistance is forthcoming immediately, we can expect to see a high rate of attrition among the

Figure TAB-1 -- Sample table in a document window

The caption is part of the table frame.
The frame border is shown here (dotted) for clarity; normally it would be invisible.



Display [✓ TABLE] FRAME COLUMN HEADINGS

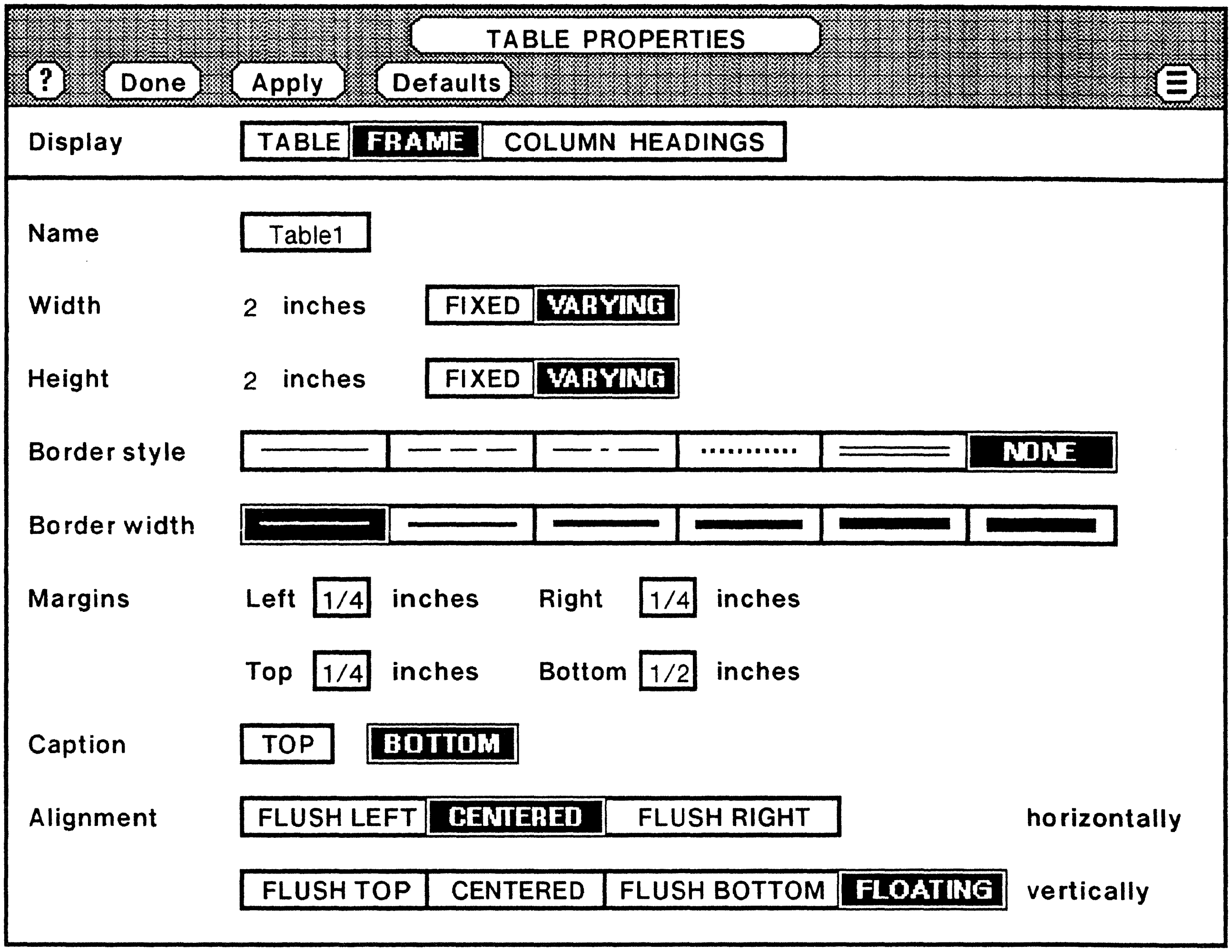
Number of rows 2 FIXED [✓ VARYING]

Number of columns 2 FIXED

Fill in by [✓ ROW] COLUMN

Figure TAB-2 -- Table property sheet

The settings shown here are the defaults for a fill-in-by-row table. Note that the number of columns is always FIXED with respect to the action of the NEXT key. However if fill in by COLUMN is specified, the FIXED/VARYING choice is placed next to the number of columns, and the number of rows is always FIXED.



Display TABLE [✓ FRAME] COLUMN HEADINGS

Name Table1

Width 2 inches FIXED [✓ VARYING]

Height 2 inches FIXED [✓ VARYING]

Border style ——— | — — — | — - — | .... | ═══ | [✓ NONE]

Border width [✓ ━━━] | ─── | ━━━ | ━━━ | ━━━ | ━━━

Margins Left 1/4 inches Right 1/4 inches

Top 1/4 inches Bottom 1/2 inches

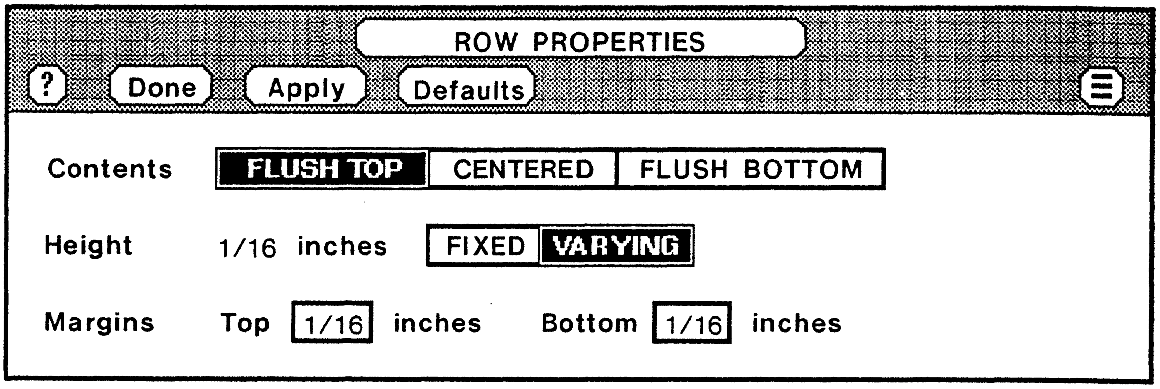
Caption TOP [✓ BOTTOM]

Alignment FLUSH LEFT [✓ CENTERED] FLUSH RIGHT horizontally

FLUSH TOP CENTERED FLUSH BOTTOM [✓ FLOATING] vertically

Figure TAB-3 -- Table Frame property sheet

Except for the absence of a RULED option, these are exactly the same properties as for text frames.
The settings shown here are the defaults.
The Width and Height numbers are modifiable only if FIXED, and they do not include the margins.
If a table is embedded in a graphics frame, the Alignment property disappears.



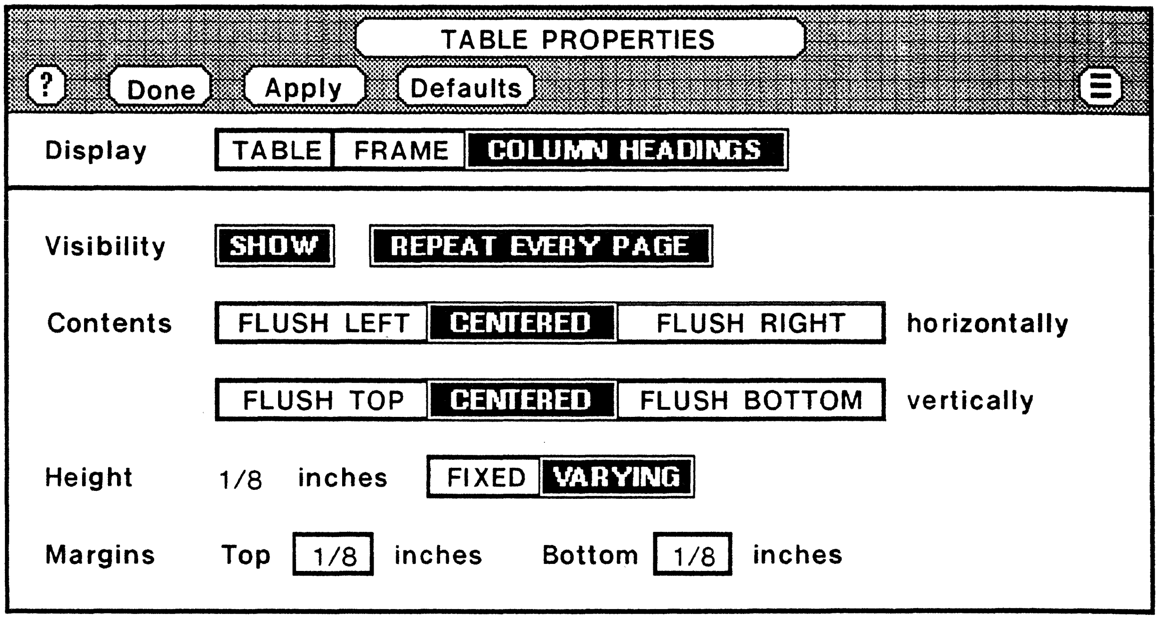
Contents [✓ FLUSH TOP] CENTERED FLUSH BOTTOM

Height 1/16 inches FIXED [✓ VARYING]

Margins Top 1/16 inches Bottom 1/16 inches

Figure TAB-4 -- Property sheet for a row

The settings shown here are the defaults.
The Height text parameter is information-only unless the height is FIXED.



Display TABLE FRAME [✓ COLUMN HEADINGS]

Visibility [✓ SHOW] [✓ REPEAT EVERY PAGE]

Contents FLUSH LEFT [✓ CENTERED] FLUSH RIGHT horizontally

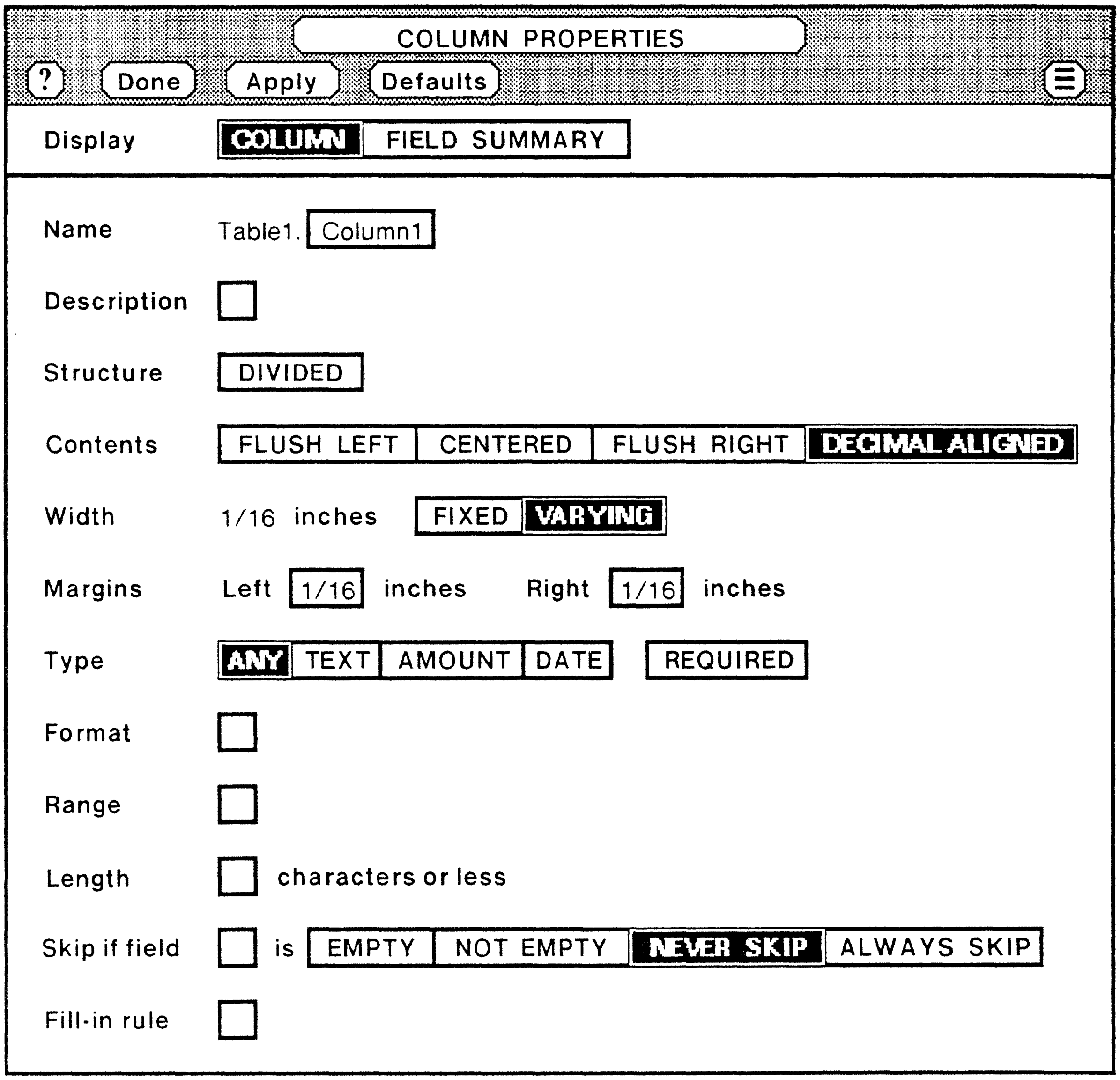
FLUSH TOP [✓ CENTERED] FLUSH BOTTOM vertically

Height 1/8 inches FIXED [✓ VARYING]

Margins Top 1/8 inches Bottom 1/8 inches

Figure TAB-5 -- Property sheet for the column headings

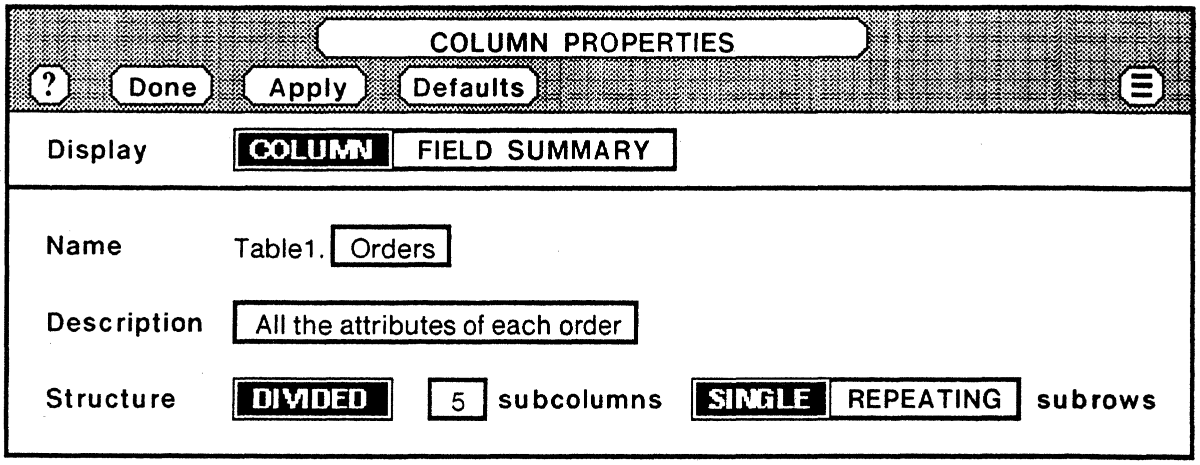
The settings shown here are the defaults.
The Height text parameter is information-only unless the height is FIXED.



Display	[✓ COLUMN] FIELD SUMMARY
Name	Table1. Column1
Description	□
Structure	DIVIDED
Contents	FLUSH LEFT CENTERED FLUSH RIGHT [✓ DECIMAL ALIGNED]
Width	1/16 inches FIXED [✓ VARYING]
Margins	Left 1/16 inches Right 1/16 inches
Type	[✓ ANY] TEXT AMOUNT DATE REQUIRED
Format	□
Range	□
Length	□ characters or less
Skip if field	□ is EMPTY NOT EMPTY [✓ NEVER SKIP] ALWAYS SKIP
Fill-in rule	□

Figure TAB-6 -- Property sheet for an undivided column

The settings shown here are the defaults.
The Width text parameter is information-only unless the width is FIXED.
The properties from Type down are exactly the same as for fields.



Display [✓ COLUMN] FIELD SUMMARY

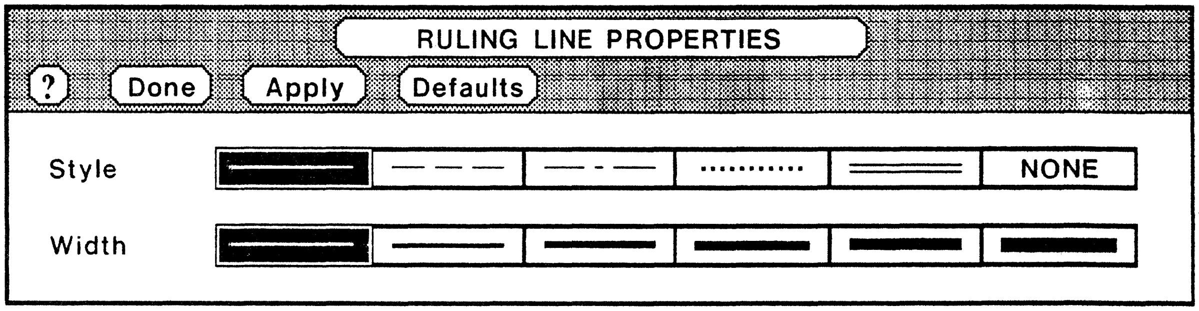
Name Table1. Orders

Description All the attributes of each order

Structure [✓ DIVIDED] 5 subcolumns [✓ SINGLE] REPEATING subrows

Figure TAB-7 -- Property sheet for a divided column

The Subcolumns and Subrows parameters appear only if the column is DIVIDED.
The other column properties (shown in Figure TAB-5) appear only if DIVIDED is off.

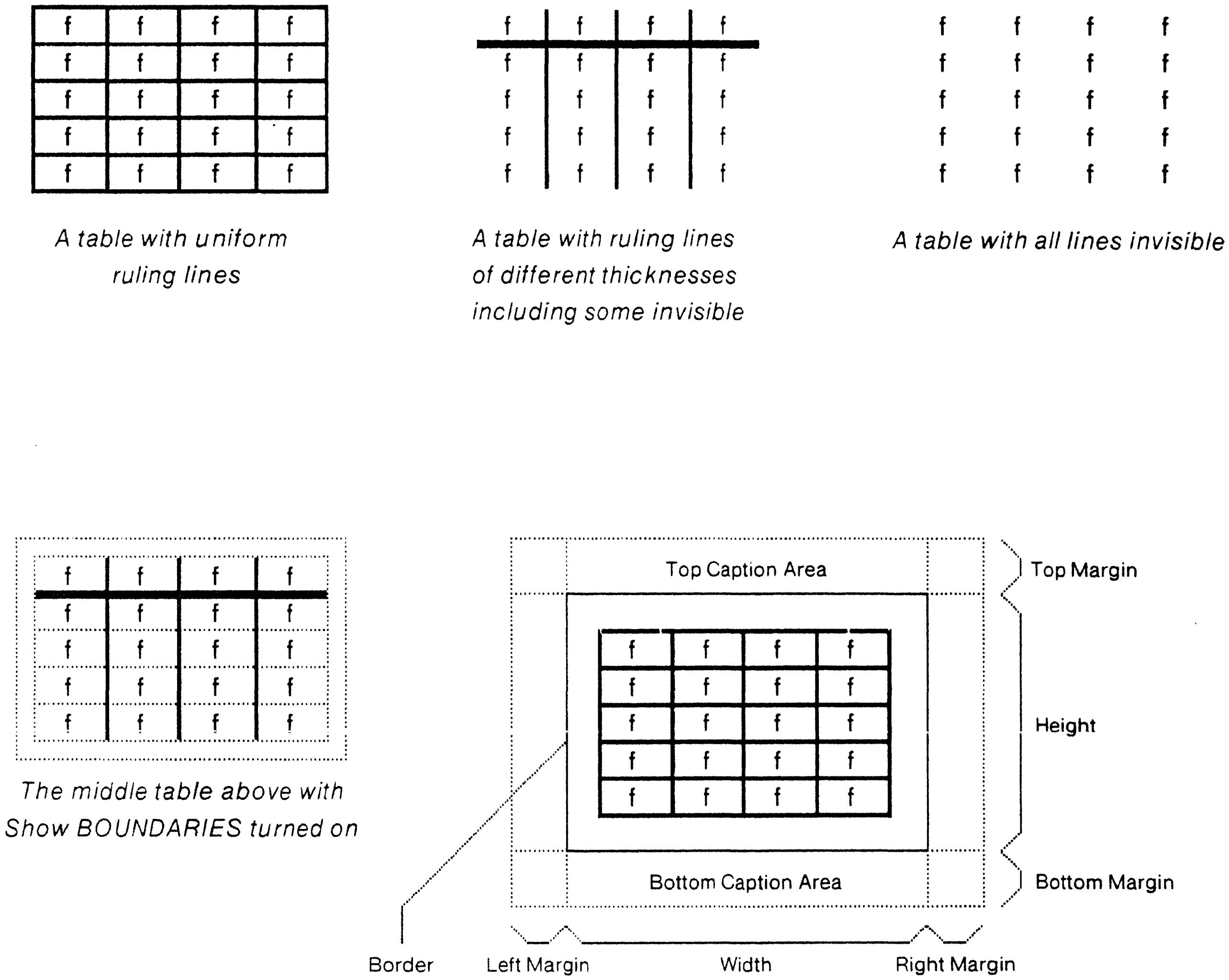


Style [✓ ———] — — — | — - — | .... | ═══ | NONE

Width [✓ ———] | ——— | ━━━ | ━━━ | ▬▬▬ | ▬▬▬

Figure TAB-8 -- Property sheet for a table ruling line

The settings shown here are the defaults.
The Style and Width options are the same as for general graphic lines.



A table with uniform ruling lines

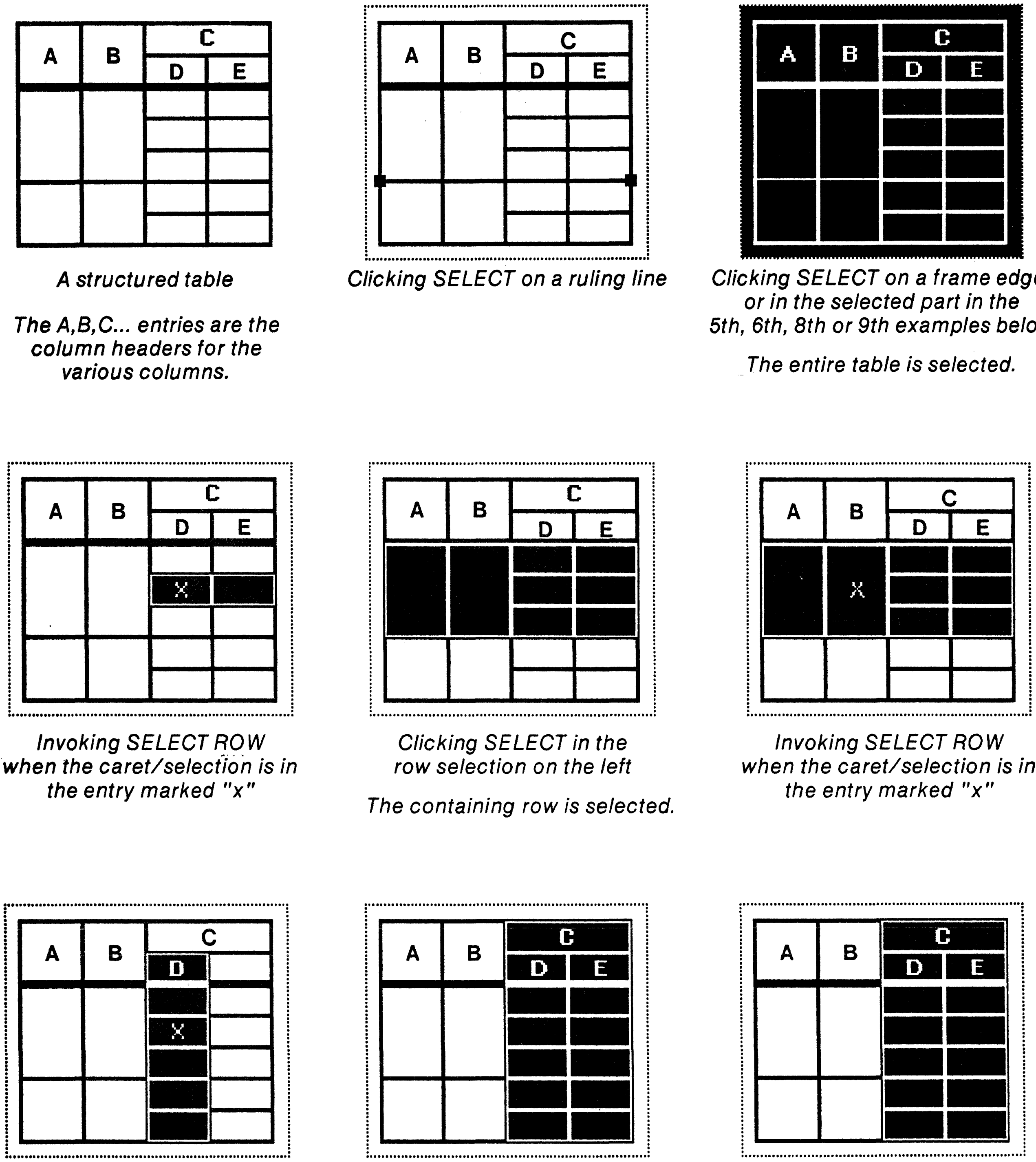
A table with ruling lines of different thicknesses including some invisible

A table with all lines invisible

The middle table above with Show BOUNDARIES turned on

The main areas of a table frame

Figure TAB-9 -- Examples of tables



A structured table

The A,B,C... entries are the column headers for the various columns.

Clicking SELECT on a ruling line

Clicking SELECT on a frame edge, or in the selected part in the 5th, 6th, 8th or 9th examples below

The entire table is selected.

Invoking SELECT ROW when the caret/selection is in the entry marked "x"

Clicking SELECT in the row selection on the left

The containing row is selected.

Invoking SELECT ROW when the caret/selection is in the entry marked "x"

Invoking SELECT COLUMN when the caret/selection is in the entry marked "x"

Clicking SELECT in the column selection on the left

The containing column is selected.

Invoking SELECT COLUMN when the caret/selection is in the "C" entry

Figure TAB-10 -- How the parts of a table are selected

Name	Children				
	Child Name	Degrees		Schools	
		Degree	Age	School Name	Year
King	Elmer	BA	21	Lawrence	1970
				Chatsworth	1972
		MA	24	Redondo	1975
	Louise	BS	22	Hillcrest	1973
				Blythe	1979
Logan					

Figure TAB-11 -- An example of a highly structured table

Using the table in Figure TAB-11, the following is a detailed example of the meaning of the NEXT and SKIP keys when the caret or selection is in various entries. Both a fill-in-by-column and a fill-in-by-row table are represented. (Note: the fill-in-by-column entries assume that either the header row is not

	Fill in by row		Fill in by column	
Caret is in	NEXT goes to	SKIP goes to	NEXT goes to	SKIP goes to
King	Elmer	Logan	Logan	Elmer
Logan	blank (Child Name)	*	Elmer	Elmer
Elmer	BA	Louise	Louise	BA
Louise	BS	Logan	blank (Child Name)	BA
blank (Child Name)	blank (Degree)	*	BA	BA
BA	21	MA	MA	21
MA	24	Lawrence	BS	21
BS	22	Hillcrest	blank (Degree)	21
blank (Degree)	blank (Age)	*	21	21
21	MA	MA	24	Lawrence
24	Lawrence	Lawrence	22	Lawrence
22	Hillcrest	Hillcrest	blank (Age)	Lawrence
blank (Age)	blank (Sch. Name)	*	Lawrence	Lawrence
Lawrence	1970	Chatsworth	Chatsworth	1970
Chatsworth	1972	Redondo	Redondo	1970
Redondo	1975	Louise	Hillcrest	1970
Hillcrest	1973	Blythe	Blythe	1970
Blythe	1979	Logan	blank (Sch. Name)	1970
blank (School Name)	blank (Year)	*	1970	1970
1970	Chatsworth	Chatsworth	1972	*
1972	Redondo	Redondo	1975	*
1975	Louise	Louise	1973	*
1973	Blythe	Blythe	1979	*
1979	Logan	Logan	blank (Year)	*
blank (Year)	**	*	***	*

* SKIP goes to the next field in the document's fill-in order.

** A new Logan-level row is added if the table is defined to have a varying number of rows. Otherwise NEXT goes to the next field in the document's fill-in order.

*** A new Children-level column is added if the table is defined to have a varying number of columns. Otherwise NEXT goes to the next field in the document's fill-in order.

Figure TAB-12 -- An example of the use of NEXT and SKIP

(PAGES 203 THROUGH 204 INTENTIONALLY BLANK)

(PAGES 203 THROUGH 204 INTENTIONALLY BLANK)

17. EQUATIONS

The equations facility enables users to type mathematical equations in a correctly type-set way. The user needs to know nothing about mathematics or the rules for typesetting equations. Filling in an equation is very much like filling in a form.

USER OVERVIEW

Equations are one of the special objects provided in Star to facilitate document composition in technical environments. Star provides no facilities for *interpreting* the equations, but only typesets them correctly.

The user enters an *equation* with the KEYBOARD key in the same way that a frame is specified. The user types ordinary text in an equation. When a special construct is desired, such as a summation sign together with its upper and lower terms, the user simply keys the appropriate KEYBOARD key combination, and moves between the constituents of the summation structure with the NEXT key. At all times, the equation is correctly type-set, relieving the user from making typographic and positioning decisions.

DETAILED SPECIFICATION

OBJECTS

Equation

An *equation* is a *special object* within a text scroll. The interpretations of selection, Move, Copy, Delete, and Next all have special meaning within equations. An equation also has special semantics relative to its surrounding text that cause it to assume the surrounding column margins, and to be rendered in the same place within the text scroll. An equation is always rendered as a *separate line* (or set of lines). Later versions of Star will allow an equation to have a width less than the full column width and thus equations will optionally occur *in-line*.

Equations can occur anywhere that text can. An equation is indivisible, even if it contains several lines. If the renderer attempts to break an equation across a page boundary, the equation is moved to the next page, leaving white space on the previous page. In this case, no text flows upward around the equation as it would for an anchored frame. Sample equations are shown in Figure EQU-1.

An equation has a rectangular boundary, whose width is always equal to the surrounding column margins, and whose height is determined by the enclosed material. User-settable margins are available to provide a buffer of blank space around the equation. Setting the top and bottom margins causes the overall height of the equation to increase. Setting the left and right margins keeps the equations characters (including the equation number, if any) from approaching the column/page margins.

An equation may consist of multiple lines. All lines after the first are aligned horizontally, depending on the location of "equality" characters in the first line. Equality characters consist of the set

$\{=, \simeq, \approx, \doteq, \neq, \leftrightarrows, \rightleftarrows, \rightarrow, \leadsto, \Rightarrow, \propto, <, \leq, \ll, >, \geq, \gg, \subset, \subseteq, \supset, \supseteq, \not\supset, \in, \notin\}$.

If a secondary line contains an equality character, this character is aligned horizontally with the first equality character occurring in the first line. If there is no equality character in the first line, all secondary lines are separately centered horizontally. If there is an equality character in the first line but none in the given secondary line, the first character in the secondary line is aligned horizontally with the first symbol following the equality character in the first line.

A multiple line equation is centered as an entire group of lines between the margins. Equation numbers, if any, do not participate in the centering.

All equations contain ordinary text and *equation structures*.

Equation Structures and Arguments

An equation structure is a stylized piece of an equation for which Star has special formatting knowledge. Equation structures are made up of predetermined symbols (such as the summation sign), together with *argument fields* which are filled in by the user at equation input time. Star equation structures and their rules for selection include:

Summation - The large Greek sigma, together with upper and lower centered arguments in a smaller typeface.

Product - The large Greek pi, together with upper and lower centered arguments in a smaller typeface.

Union - The large union sign, together with upper and lower centered arguments in a smaller typeface.

Intersection - The large intersection sign, together with upper and lower centered arguments in a smaller typeface.

Limit - The word "lim" together with lower centered argument in a smaller typeface.

Integral - The large integral sign, together with upper right corner, and lower right corner arguments in a smaller typeface.

Line Integral - The large line integral sign, together with upper right corner, and lower right corner arguments in a smaller typeface.

Vertical Bar - A vertical bar of fixed height, together with upper right corner, and lower right corner arguments in a smaller typeface.

Fraction - A horizontal line of variable width, together with mutually centered numerator and denominator arguments in the normal typeface.

Diagonal Fraction - A diagonal line of fixed length, together with upper left and lower right arguments in the normal typeface.

Scripts - Lower right corner and upper right corner arguments adjusted to the height of the preceding character, set in a smaller typeface.

Four Scripts - Upper left corner, lower left corner, upper right corner, and lower right corner arguments set in a smaller typeface, together with a center argument placed on the normal baseline.

Parenthesis - A pair of parentheses assuming one of a set of discrete sizes, equal to or greater than the height of the enclosed argument, and centered vertically on the surrounding line.

Bracket - A pair of brackets assuming one of a set of discrete sizes, equal to or greater than the height of the enclosed argument, and centered vertically on the surrounding line.

Brace - A pair of braces assuming one of a set of discrete sizes, equal to or greater than the height of the enclosed argument, and centered vertically on the surrounding line.

Left-Brace - A left brace assuming one of a set of discrete sizes, equal to or greater than the height of the following argument, and centered vertically on the surrounding line.

Square Root - A root symbol assuming one of a set of discrete sizes, equal to or greater than the height of the enclosed argument, together with a horizontal line.

Nth Root - A root symbol assuming one of a set of discrete sizes, equal to or greater than the height of the enclosed argument, together with a horizontal line and an argument representing the power of the root.

Over Bar - An overbar of variable length, equal to the width of the single enclosed argument.

Block - A user-settable rectangular array of arguments. A matrix would be constructed by embedding a block inside brackets. The block is the only equation structure that possesses its own property sheet. See Block Property Sheet below, and Figure EQU-4.

An argument within a structure may consist of multiple lines. The argument is formatted exactly as if it were a full multiple line equation at the top level, and the argument as a whole is subject to the formatting constraints applied by its enclosing structure.

Font Level

Every equation may contain up to three font levels: regular, small, and smallest. Characters typed initially into an equation at the top level are given the regular font level. Characters in structures (including the sum, product, etc. signs) where a reduced font is called for are demoted: regular characters become small, and small characters become smallest. Smallest characters usually only occur in the argument of a structure that is itself an argument of a structure. Further embedding of structures does not further reduce the font level. The only other way characters below the regular font level appear is if the user changes the overall font level of the equation via the Equation Property Sheet (see below) in order to make the equation fit into a smaller space.

Upon first entering an equation, the regular font is identified from the default equation properties, which are built into the implementation. If more than one choice for the equation's regular font exist, then the user may change the regular font at any time via the Equation Property sheet. The small font level is 2/3 the size of the regular and the smallest font level is 1/2 the size of the regular. If a character in the smallest font is not available, it is promoted to the small level. If a character in the small font is not available, it is promoted to the regular level.

Equation Property Sheet

The equation property sheet (see Figures EQU-2 and EQU-3) contains parameters relating to equations, and is displayed when the user selects anywhere within the equation, and presses PROPERTIES. The property sheet contains a display parameter that allows the user to switch between FONT IDENTIFICATION and SPACING. The font identification parameters are: identification of the regular font family; and font level. Both parameters affect the current selection within the equation. The identification of the regular font family is an information-only parameter if there is only one available font family.

The spacing parameters are margins (top, bottom, left, and right). Changing the spacing parameters affects what is selected within the equation. If the whole equation is selected, the spacing affects the equation boundary. If a structure or an argument is selected, the spacing affects just that structure or argument. If a character is selected, the spacing of that individual character is affected. For an extended selection, only the first character or structure in the selection is affected, and a warning message is posted in the message area: "Changing the spacing of an extended equation selection affects only the first character or structure".

Block Property Sheet

The block property sheet (see Figure EQU-4) contains parameters relating to the block structure, and is displayed when the user selects a block and presses PROPERTIES. A block can be selected by clicking upward in a block element until the whole structure is selected, or by extending an arbitrary selection in one block element into any other block element. In addition to the block properties, the user may also set font identification and spacing properties by invoking the appropriate display parameters in the property sheet. Spacing parameters for blocks affect the border between the exterior of the block elements and surrounding symbols. Spacing between the rows and columns is controlled in the block property sheet proper as explained in the next paragraph.

The block property sheet contains parameters for the number of rows and number of columns as well as the spacing between rows and the spacing between columns. When the user invokes DONE, the number of rows and the number of columns are checked for various constraints. The number of rows or the number of columns may not be reduced in such a way that data is lost, i.e. the discarded rows or columns must be empty. If the number of rows or columns is ridiculously large, Star aborts the command, displays an error message, and resets the values to their old settings. If the inter-column spacing plus the width of the existing entries would force the equation to be wider than its containing margins, then Star aborts the command, displays an error message, and resets the values to their old settings.

ACTIONS

Insert Equation

The user may insert an equation into a document using the KEYBOARD key, as described in *Keyboard Sets*.

The user may also insert an equation by moving or copying an existing one from elsewhere, or by expanding an abbreviation containing an equation.

Insert Equation Structure

While typing at any time within an equation, the user may press KEYBOARD, as described in *Keyboard Sets*. The default keyboard is set to "Equation Structures". While holding down the KEYBOARD key, the user may enter the desired structure by striking the desired corresponding letter. Alternatively, while holding the KEYBOARD key down, the user may open the keyboard window by pressing the SHOW KEYBOARD key. All of the available equation structures are shown on their corresponding keyboard keys, as depicted in Figure EQU-5. Upon entry of a structure, predetermined symbols (such as the summation sign) are displayed, and the caret is automatically positioned within the first argument.

The user advances through the arguments of an empty structure by pressing NEXT. Pressing NEXT in the last argument of a new structure causes the caret to return to the baseline in the equation following the structure.

Scripts - Within an equation, scripts are entered either with the SUPERSCRIPT or the SUBSCRIPT key. Both keys actually enter the full script structure including both arguments, but the SUPERSCRIPT key is an optimization that is equivalent to SUBSCRIPT plus NEXT.

Four Scripts - The tensor structure must be entered via the appropriate KEYBOARD combination, and cannot be entered via the SUPERSCRIPT or SUBSCRIPT keys. When a tensor is entered, the cursor is positioned in the central argument field. NEXT then advances consecutively to the lower pre-script, the upper pre-script, the lower post-script, and the upper postscript.

Blocks - Block structures are initially created as two rows by two columns. If the user wishes a different size block structure, he changes the number of rows or columns via the Block Property Sheet (see Figure EQU-4). The NEXT key is used to advance through a block, column by column within a row, row by row, using NEXT.

Create Multiple Line Equation

A multiple line equation is created by typing the carriage return (New Line or New Para) at the top level of the equation. The equation height expands and the cursor is positioned aligned with the first equality character in the top line, if any. Otherwise the cursor is centered with respect to the top line. If the carriage return is inserted in the middle of an existing line, the line is split. Further secondary lines in a multiple line equation may be created with the return key in the same manner.

Create Multiple Line Argument

A multiple line argument is created by typing the carriage return (New Line or New Para) within the argument. The argument height expands and the cursor is positioned aligned with the first equality character in the top line, if any. Otherwise the cursor is centered with respect to the top line. If the carriage return is inserted in the middle of an existing line, the line is split. Further secondary lines in a multiple line argument may be created with the return key in the same manner.

If the argument is centered (such as a limit to a summation), then the multiple line argument is centered as a whole, while observing the alignment of equality characters. If the argument is left flush (such as the argument to a left brace) then the multiple line argument is left flush as a whole, while observing the alignment of equality characters.

Select Equation Character, Position Caret

Any character of an equation can be selected in the same way that an ordinary text character is. Like ordinary text, an insertion caret appears to the right of the highlighted selection. Most equation structures have components that are not characters (such as the summation sign), and thus are not selectable in isolation. When the user attempts to select such a symbol, the entire structure is selected (see Select Equation Structure, below).

The insertion caret can be placed in front of the first character in an equation or in an embedded argument (during selection, or during Move and Copy) by positioning the cursor within the left half of that character and pressing SELECT.

The insertion caret can be placed in an empty argument field (during selection, or during Move and Copy) by positioning it in a small live selection region at the appropriate location.

Select Argument

An argument is selected by double clicking an enclosed character, or by multiple clicking up through an enclosed structure. An argument cannot be selected (for purposes of examining its property sheet for instance) by extending the selection from the first character in the argument to the last. Such an extended selection would only select the individual characters of the argument.

When an argument is selected, an insertion caret appears at the right end of the selection. Typing with an argument selected causes new characters to be added to the argument.

Select Equation Structure

An equation structure is selected by either multiple clicking up through an enclosed argument, or by single clicking anywhere within the rectangular extent of the structure not occupied by a selectable character.

When a structure is selected, an insertion caret appears to the right of the structure, positioned on the baseline exterior to the structure. Typing with a structure selected causes new characters to be added to the right of the structure.

Select Equation

An equation is selected by multiple clicking to the top level of the equation.

When an equation is selected, an insertion caret appears at the right end of the selection. Typing with an equation selected causes new characters to be added to the equation at the top level.

Extend Equation Selection

A selection in an equation may be extended in the same way selections in ordinary text are extended. However, because of the embedded nature of equation structures, a selection is not allowed to extend from midway inside an embedded argument to somewhere outside. When the user extends the selection across an argument boundary, the selection expands to include the entire embedded structure. (Thus, it is impossible to make an extended selection of just the upper and lower limit arguments of a summation). The root of an extended selection is remembered as long as the selection is in force; thus, if the selection is shrunk, the correct set of arguments is highlighted, as if the selection had been strictly grown from the root to that point.

Next

Pressing the NEXT key inside of an equation, in general, advances the selection through the equation in a depth first way. The progression of selections through a complex equation is illustrated in Figure EQU-6. The result of pressing the NEXT key always selects an argument or a structure. If a selected argument field is empty, a lone caret is positioned correctly.

If the entire equation is selected (see last illustration in Figure EQU-6), pressing NEXT causes the caret to be placed in the surrounding text scroll immediately after the equation.

Skip

Pressing the SKIP key inside an equation causes the caret to be advanced in the same manner as the NEXT key except that *the caret stops only in empty argument fields.*

Tab (Enter Equation Number)

Pressing the TAB key inside an equation at the top level, or at the top level within any line of a multi-line equation, is interpreted as a right flushed tab (to the margin less the right border spacing). This allows the user to enter an equation number with the correct formatting. Pressing the TAB key within an equation under any other circumstances causes no action.

Change Font Level

The Equation Property Sheet possesses the overall font level property. Setting the font level causes the nested font sublevels to be incremented appropriately. The top level of the equation is set to what the user specifies, and embedded levels are incrementally defined from that font level in the usual way.

Change Alphabet

The user may enter a Greek symbol or a mathematical symbol by using the standard virtual keyboard mechanism described in *Keyboard Sets.*

Enter Accent

The user may enter any of the standard accents provided in Star by switching to the European Letters and Accents virtual keyboard, keying the desired accent or accents, and then keying the symbol to be modified. The accents act as dead keys, just as they do in regular document text.

Move

A selection in an equation can be moved to any other location within the same or a different equation. The moved segment of the equation assumes the appropriate font level of its destination. Since there is no meaning associated with moving an argument, an argument selection is automatically converted to an extended selection of all the argument's contents before the move actually takes place.

Text may not be moved between an equation and ordinary text outside an equation. Later versions of Star may relax this restriction.

Copy

The Copy command within equations is analogous to Move.

Delete

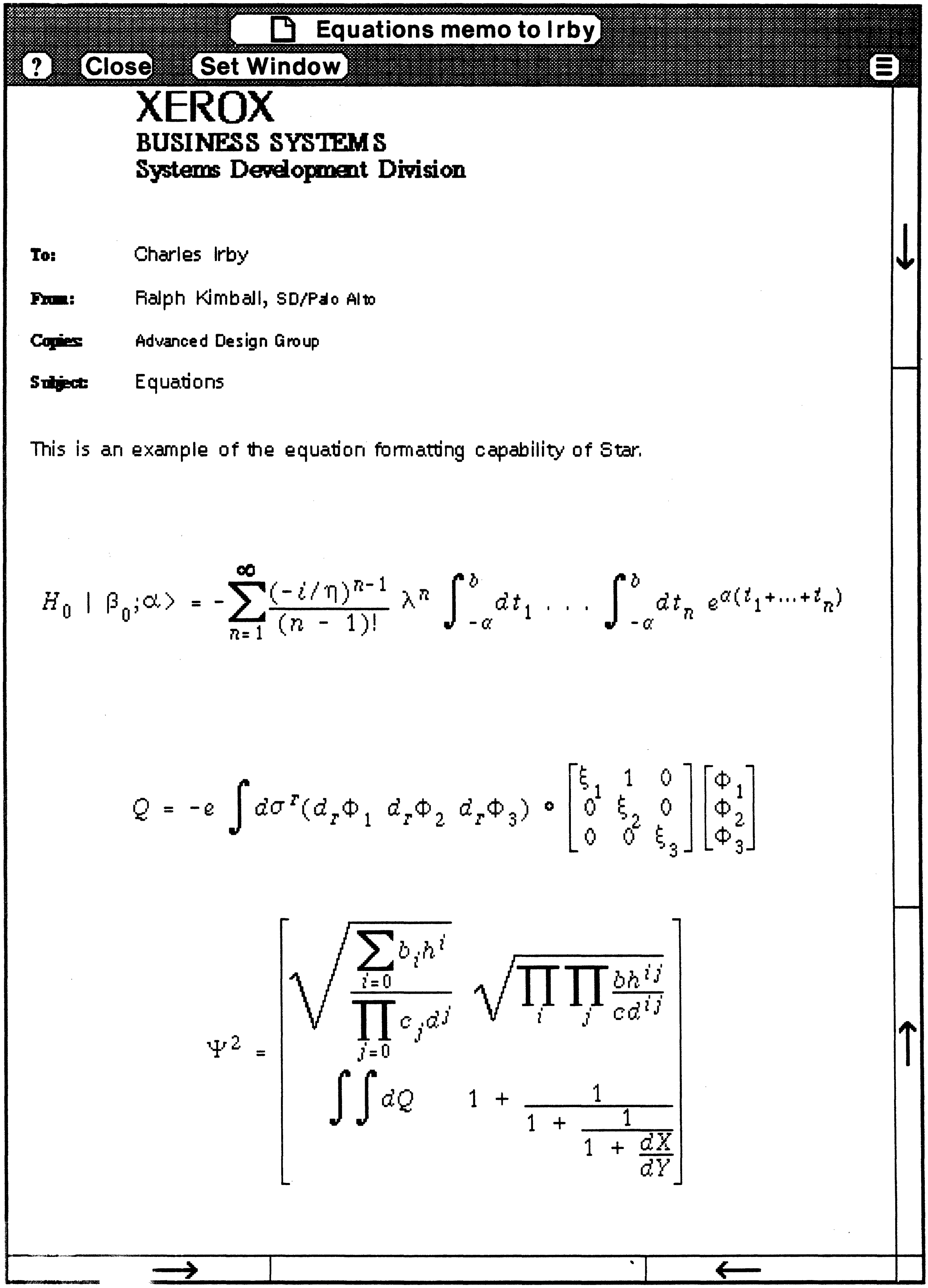
A selection within an equation may be deleted. Since there is no meaning associated with deleting an argument, an argument selection is automatically converted to an extended selection of all the argument's contents before the deletion actually takes place.

Copy Properties

The Copy Properties command can only be applied between like objects within equations, that is, between equations themselves, between structures, between arguments, or between equation characters.

Search, Substitute

In this version of Star, SEARCH and SUBSTITUTE do not apply to the contents of equations.



XEROX
BUSINESS SYSTEMS
Systems Development Division

To: Charles Irby

From: Ralph Kimball, SD/Palo Alto

Copies: Advanced Design Group

Subject: Equations

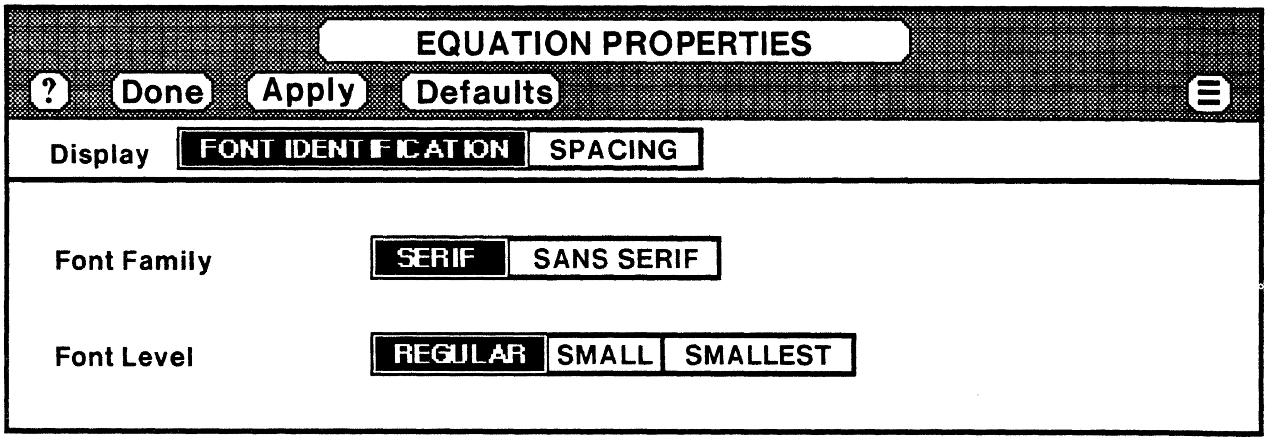
This is an example of the equation formatting capability of Star.

$$H_0 \mid \beta_0;\alpha\rangle = -\sum_{n=1}^{\infty}\frac{(-i/\eta)^{n-1}}{(n-1)!}\,\lambda^n \int_{-a}^{b} dt_1 \;.\;.\;.\; \int_{-a}^{b} dt_n \; e^{a(t_1+\ldots+t_n)}$$

$$Q = -e\int d\sigma^r(d_r\Phi_1 \;\; d_r\Phi_2 \;\; d_r\Phi_3) \circ \begin{bmatrix} \xi_1 & 1 & 0 \\ 0 & \xi_2 & 0 \\ 0 & 0 & \xi_3 \end{bmatrix} \begin{bmatrix} \Phi_1 \\ \Phi_2 \\ \Phi_3 \end{bmatrix}$$

$$\Psi^2 = \begin{bmatrix} \sqrt{\dfrac{\sum\limits_{i=0} b_i h^i}{\prod\limits_{j=0} c_j d^j}} \;\; \sqrt{\prod\limits_i \prod\limits_j \frac{bh^{ij}}{cd^{ij}}} \\ \displaystyle\int\!\!\int dQ \qquad 1 + \cfrac{1}{1 + \cfrac{1}{1 + \cfrac{dX}{dY}}} \end{bmatrix}$$

Figure EQU-1 -- A window showing a document with equations

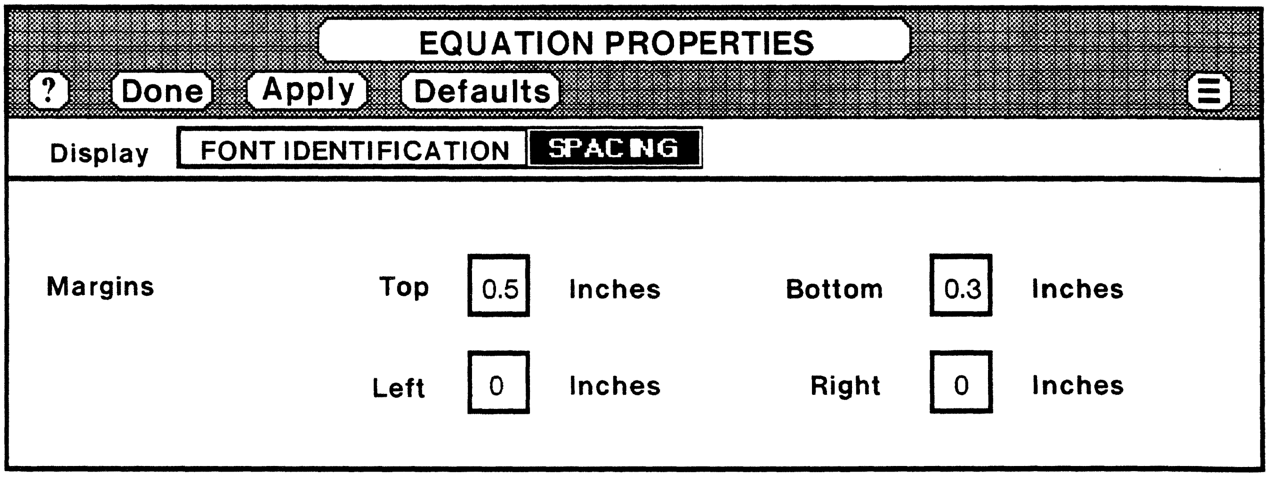


Display [✓ FONT IDENTIFICATION] SPACING

Font Family [✓ SERIF] SANS SERIF

Font Level [✓ REGULAR] SMALL SMALLEST

Figure EQU-2 -- The Equation Property Sheet
Showing Equation Font Identification

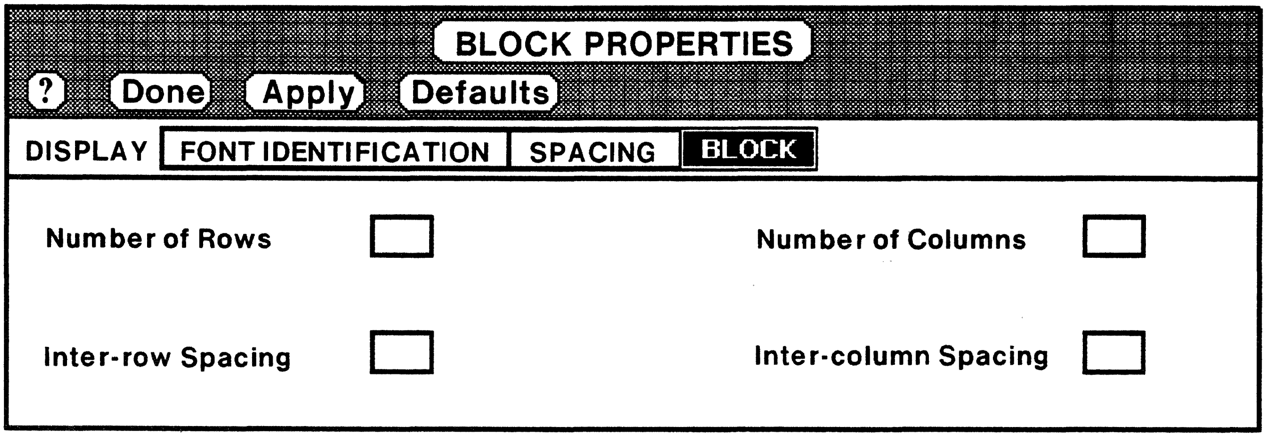


Display FONT IDENTIFICATION [✓ SPACING]

Margins Top 0.5 Inches Bottom 0.3 Inches

Left 0 Inches Right 0 Inches

Figure EQU-3 -- The Equation Property Sheet
Showing Spacing



BLOCK PROPERTIES

? Done Apply Defaults

DISPLAY | FONT IDENTIFICATION | SPACING | BLOCK

Number of Rows □ Number of Columns □

Inter-row Spacing □ Inter-column Spacing □

Figure EQU-4 -- The Block Property Sheet

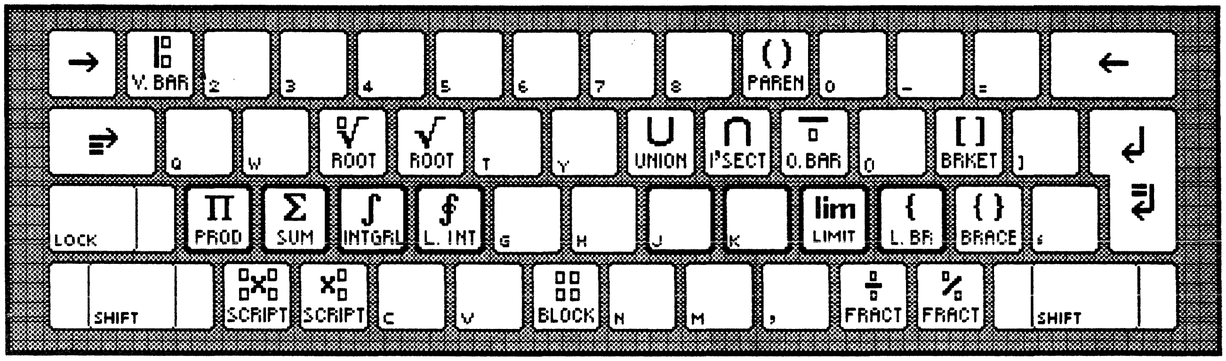
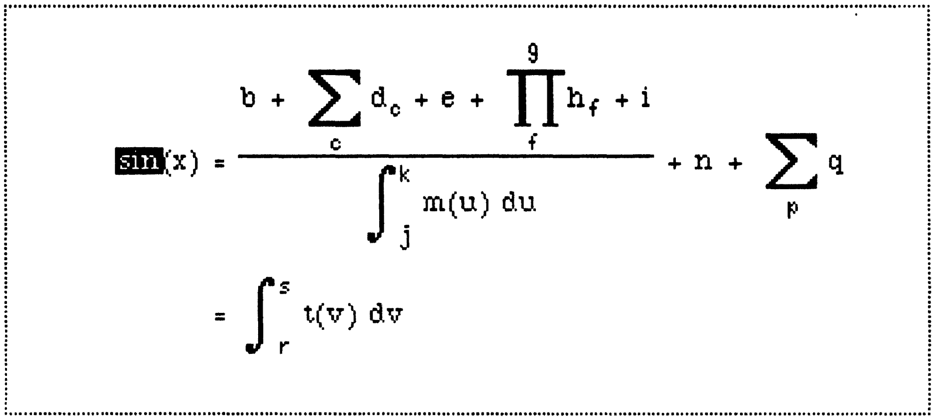


Figure EQU-5 -- The Equation Structures Virtual Keyboard

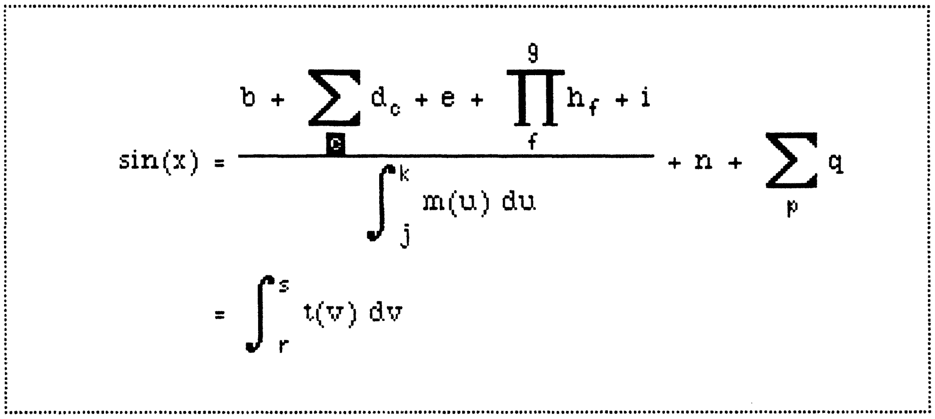
Original Selection:



$$\sin(x) = \frac{b + \sum_{c} d_c + e + \prod_{f}^{g} h_f + i}{\int_{j}^{k} m(u)\,du} + n + \sum_{p} q$$

$$= \int_{r}^{s} t(v)\,dv$$

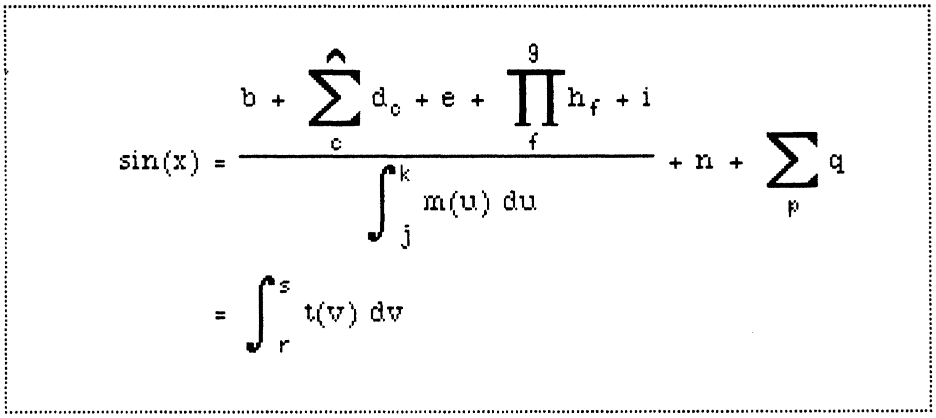
NEXT:



$$\sin(x) = \frac{b + \sum_{c} d_c + e + \prod_{f}^{g} h_f + i}{\int_{j}^{k} m(u)\,du} + n + \sum_{p} q$$

$$= \int_{r}^{s} t(v)\,dv$$

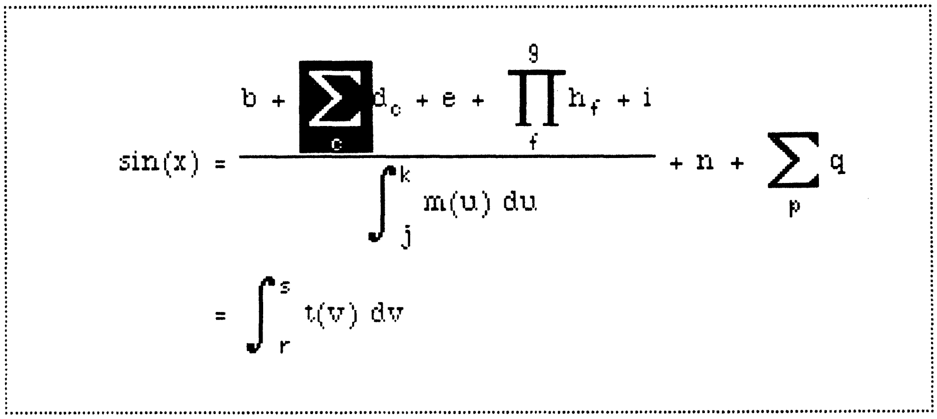
NEXT:



$$\sin(x) = \frac{b + \overset{\wedge}{\sum_{c}} d_c + e + \prod_{f}^{g} h_f + i}{\int_{j}^{k} m(u)\,du} + n + \sum_{p} q$$

$$= \int_{r}^{s} t(v)\,dv$$

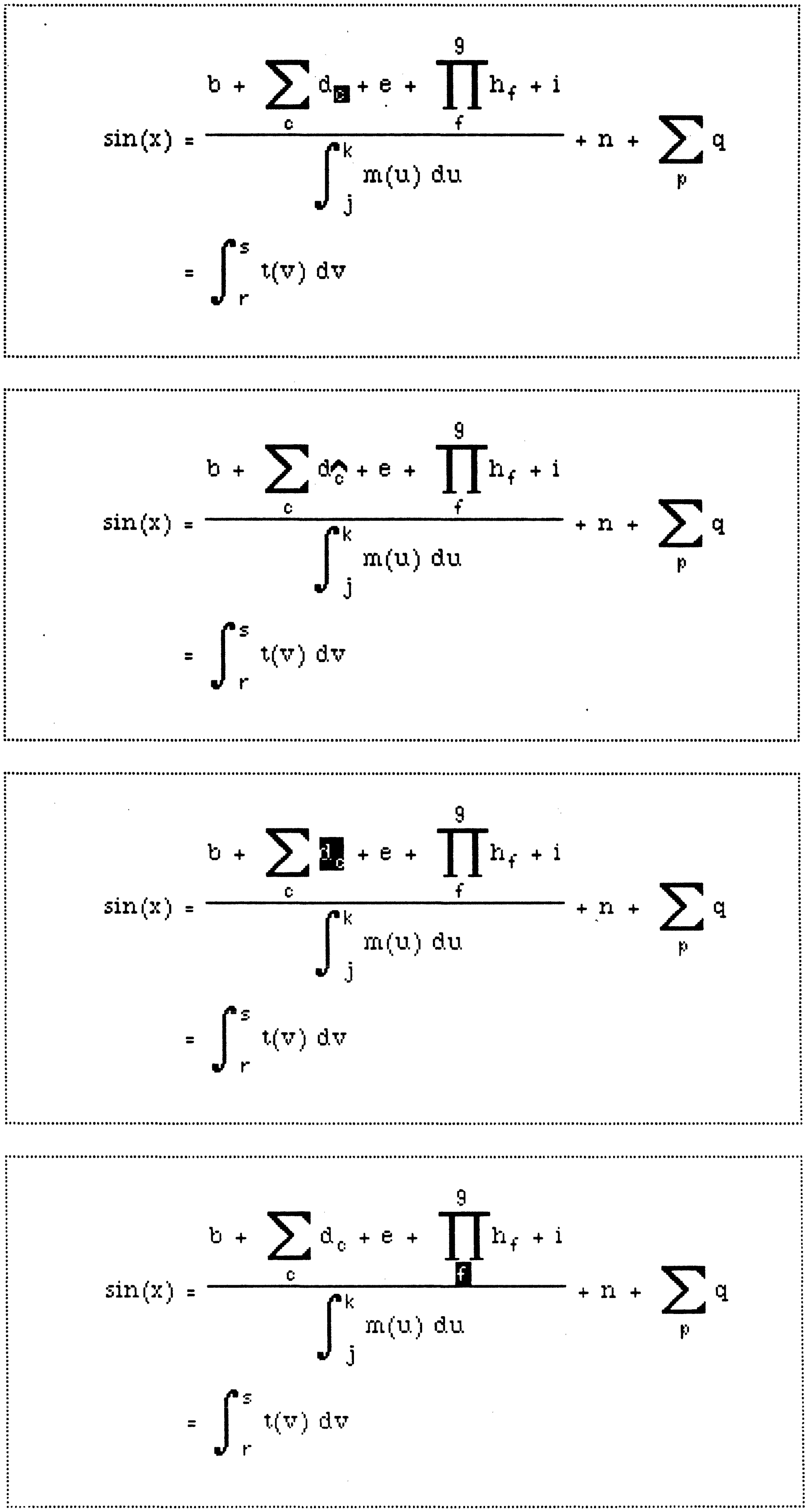
NEXT:



$$\sin(x) = \frac{b + \sum_{c} d_c + e + \prod_{f}^{g} h_f + i}{\int_{j}^{k} m(u)\,du} + n + \sum_{p} q$$

$$= \int_{r}^{s} t(v)\,dv$$

Figure EQU-6 -- The Behavior Of NEXT.



NEXT:

$$\sin(x) = \frac{b + \sum_{c} d_{\boxed{c}} + e + \prod_{f}^{g} h_f + i}{\int_j^k m(u)\,du} + n + \sum_p q$$

$$= \int_r^s t(v)\,dv$$

NEXT:

$$\sin(x) = \frac{b + \sum_{c} d_{\hat{c}} + e + \prod_{f}^{g} h_f + i}{\int_j^k m(u)\,du} + n + \sum_p q$$

$$= \int_r^s t(v)\,dv$$

NEXT:

$$\sin(x) = \frac{b + \sum_{c} \boxed{d_c} + e + \prod_{f}^{g} h_f + i}{\int_j^k m(u)\,du} + n + \sum_p q$$

$$= \int_r^s t(v)\,dv$$

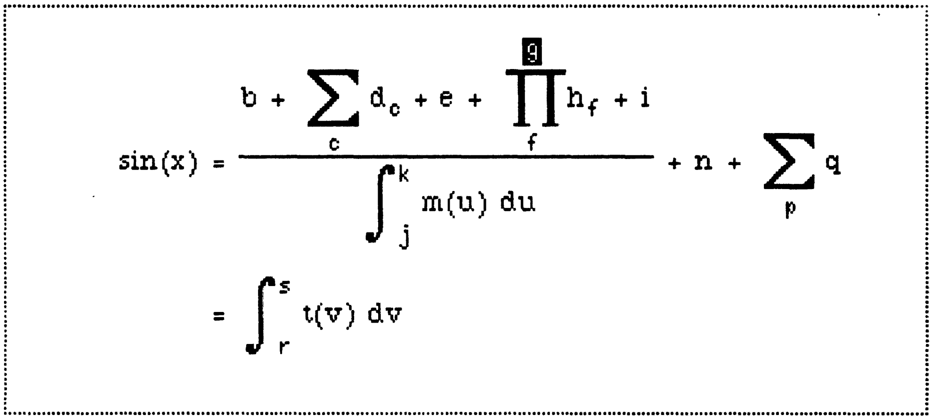
NEXT:

$$\sin(x) = \frac{b + \sum_{c} d_c + e + \prod_{\boxed{f}}^{g} h_f + i}{\int_j^k m(u)\,du} + n + \sum_p q$$

$$= \int_r^s t(v)\,dv$$

Figure EQU-6 (continued) -- The Behavior of NEXT

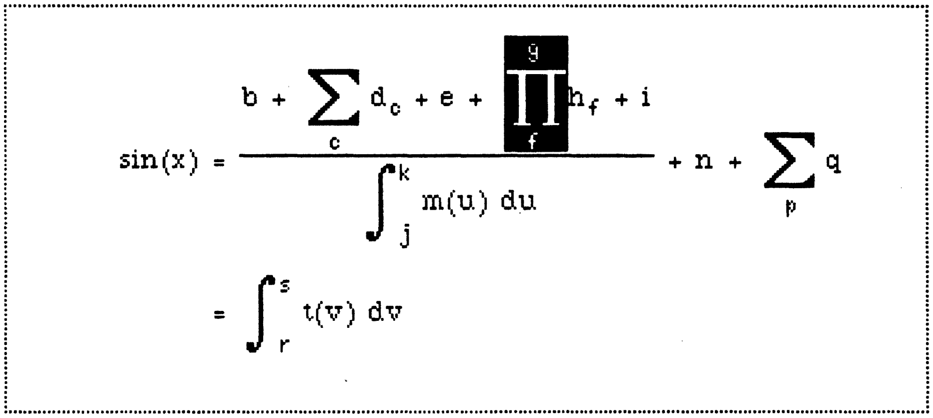
NEXT:



$$\sin(x) = \frac{b + \sum_{c} d_c + e + \prod_{f}^{\boxed{g}} h_f + i}{\int_{j}^{k} m(u)\,du} + n + \sum_{p} q$$

$$= \int_{r}^{s} t(v)\,dv$$

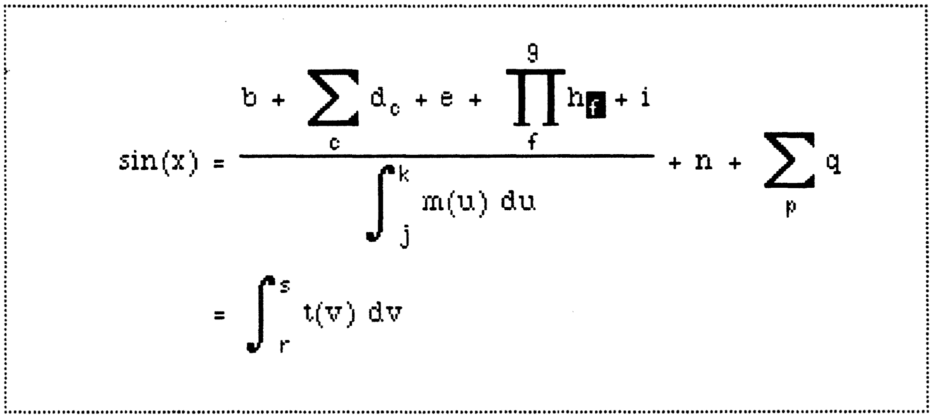
NEXT:



$$\sin(x) = \frac{b + \sum_{c} d_c + e + \boxed{\prod_{f}^{g}} h_f + i}{\int_{j}^{k} m(u)\,du} + n + \sum_{p} q$$

$$= \int_{r}^{s} t(v)\,dv$$

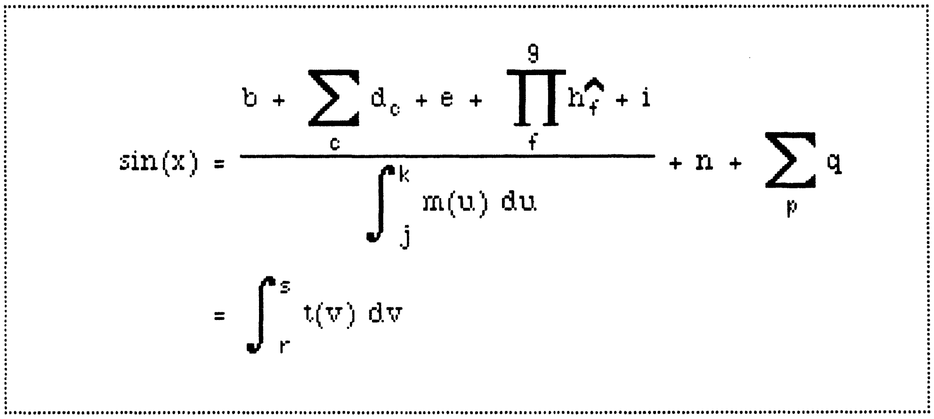
NEXT:



$$\sin(x) = \frac{b + \sum_{c} d_c + e + \prod_{f}^{g} h_{\boxed{f}} + i}{\int_{j}^{k} m(u)\,du} + n + \sum_{p} q$$

$$= \int_{r}^{s} t(v)\,dv$$

NEXT:

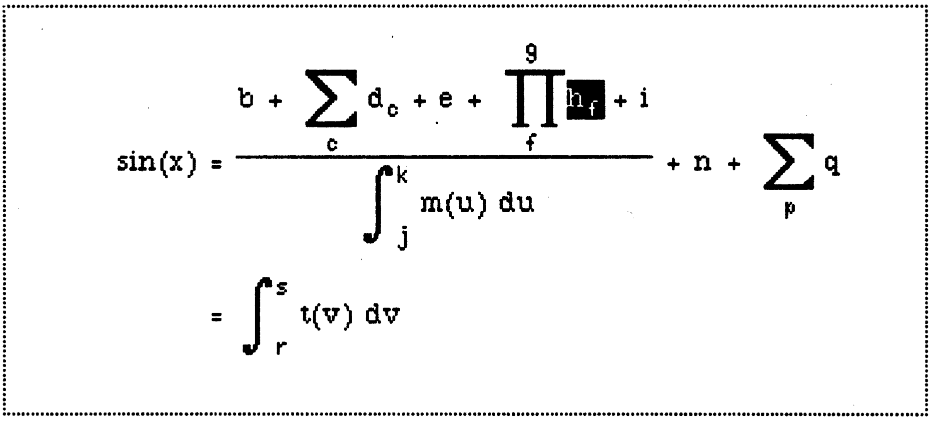


$$\sin(x) = \frac{b + \sum_{c} d_c + e + \prod_{f}^{g} h_{f}^{\wedge} + i}{\int_{j}^{k} m(u)\,du} + n + \sum_{p} q$$

$$= \int_{r}^{s} t(v)\,dv$$

Figure EQU-6 (continued) -- The Behavior of NEXT.

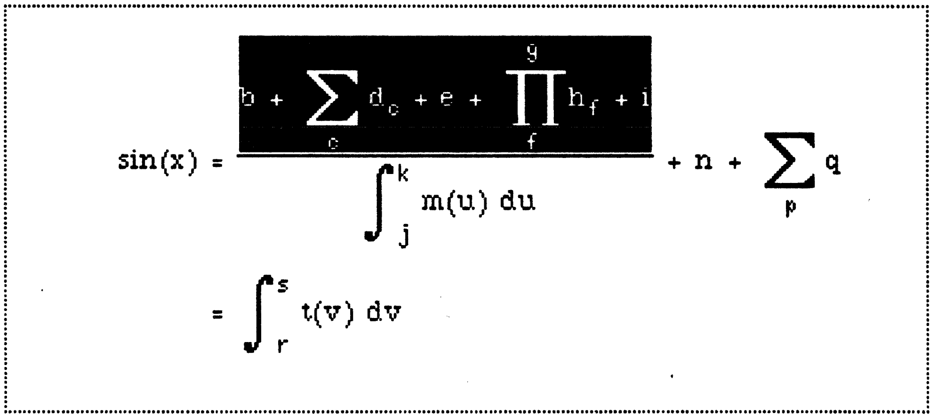
NEXT:



$$\sin(x) = \frac{b + \sum_{c} d_c + e + \prod_{f}^{g} h_f + i}{\int_{j}^{k} m(u)\, du} + n + \sum_{p} q$$

$$= \int_{r}^{s} t(v)\, dv$$

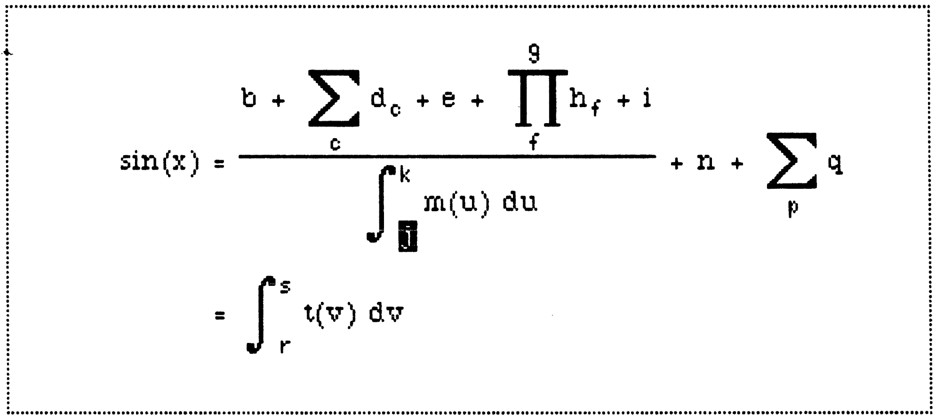
NEXT:



$$\sin(x) = \frac{b + \sum_{c} d_c + e + \prod_{f}^{g} h_f + i}{\int_{j}^{k} m(u)\, du} + n + \sum_{p} q$$

$$= \int_{r}^{s} t(v)\, dv$$

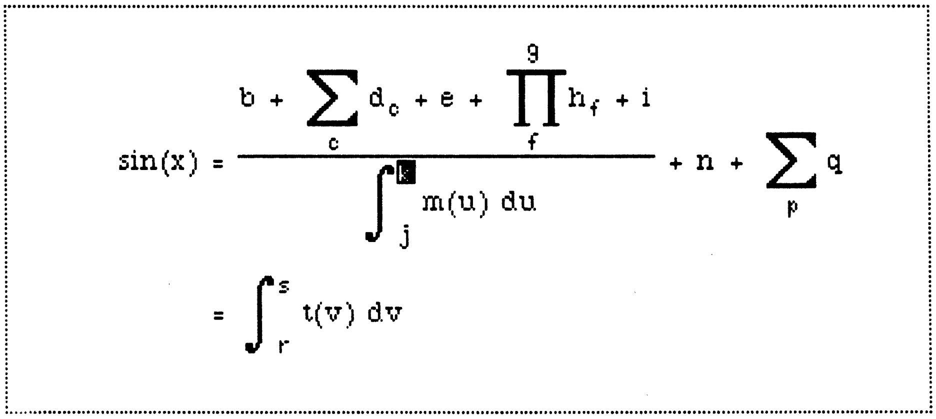
NEXT:



$$\sin(x) = \frac{b + \sum_{c} d_c + e + \prod_{f}^{g} h_f + i}{\int_{j}^{k} m(u)\, du} + n + \sum_{p} q$$

$$= \int_{r}^{s} t(v)\, dv$$

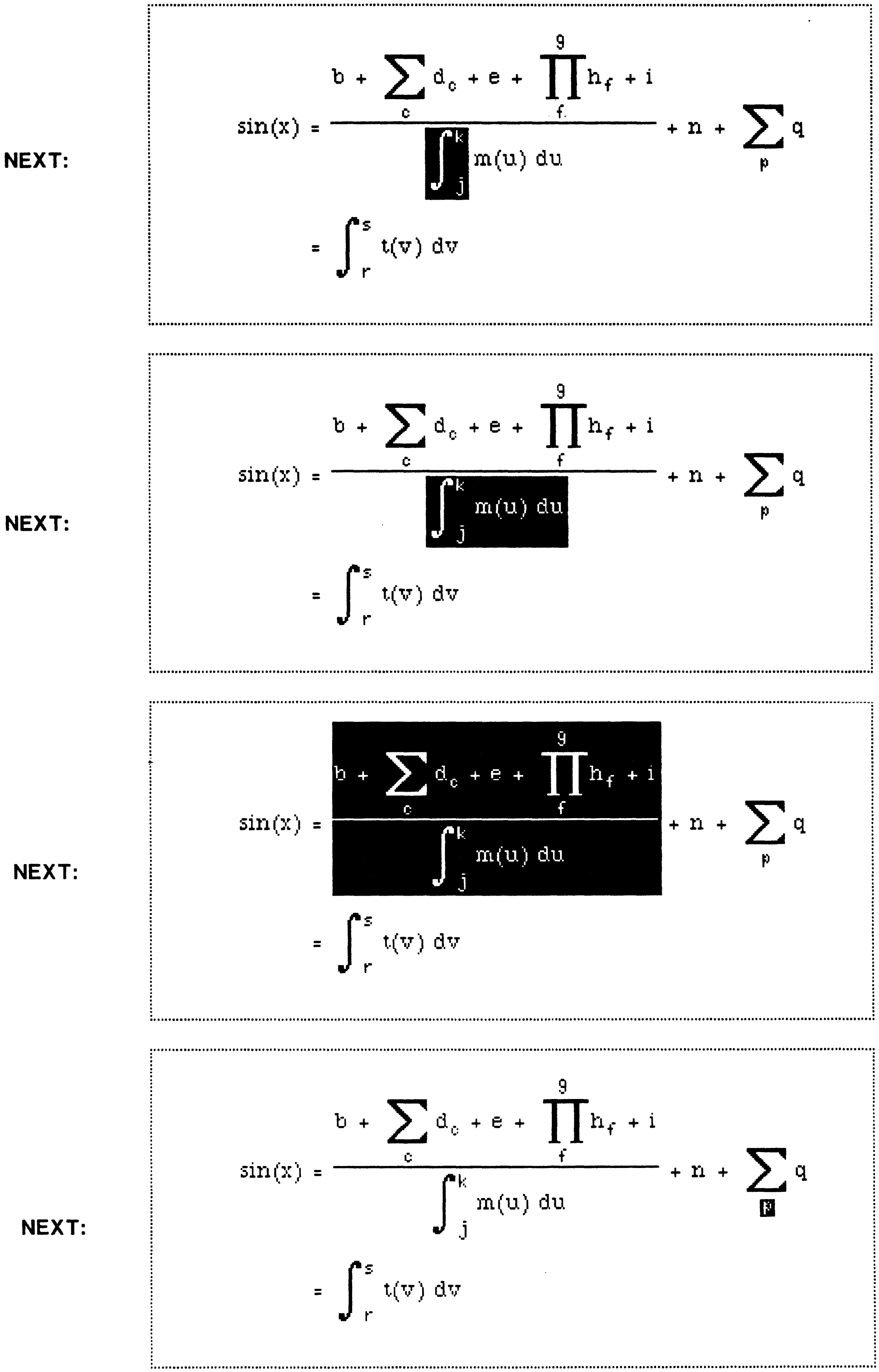
NEXT:



$$\sin(x) = \frac{b + \sum_{c} d_c + e + \prod_{f}^{g} h_f + i}{\int_{j}^{k} m(u)\, du} + n + \sum_{p} q$$

$$= \int_{r}^{s} t(v)\, dv$$

Figure EQU-6 (continued) -- The Behavior of NEXT.



NEXT:

$$\sin(x) = \frac{b + \sum_{c} d_c + e + \prod_{f}^{g} h_f + i}{\int_{j}^{k} m(u)\ du} + n + \sum_{p} q$$
$$= \int_{r}^{s} t(v)\ dv$$

NEXT:

$$\sin(x) = \frac{b + \sum_{c} d_c + e + \prod_{f}^{g} h_f + i}{\int_{j}^{k} m(u)\ du} + n + \sum_{p} q$$
$$= \int_{r}^{s} t(v)\ dv$$

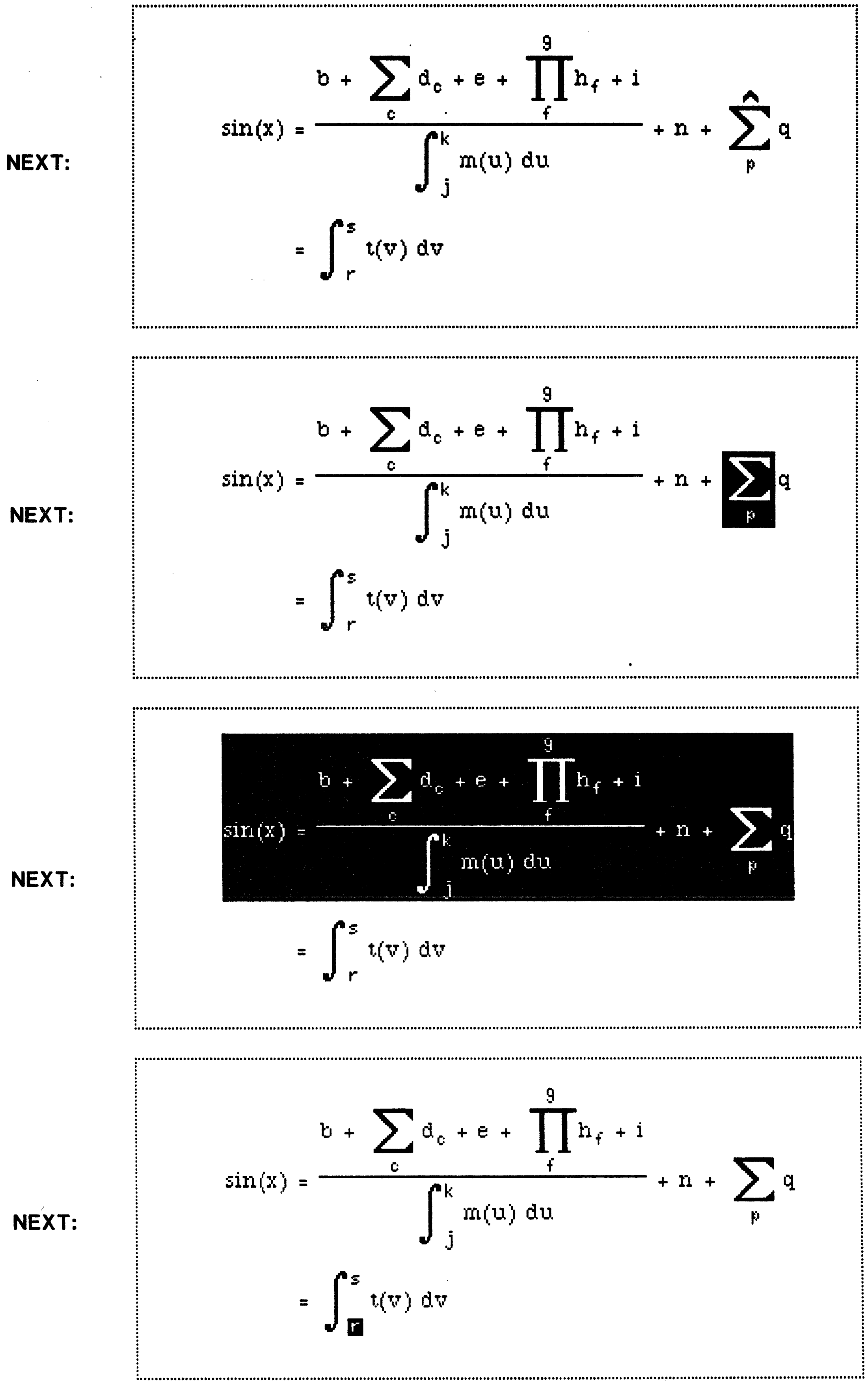
NEXT:

$$\sin(x) = \frac{b + \sum_{c} d_c + e + \prod_{f}^{g} h_f + i}{\int_{j}^{k} m(u)\ du} + n + \sum_{p} q$$
$$= \int_{r}^{s} t(v)\ dv$$

NEXT:

$$\sin(x) = \frac{b + \sum_{c} d_c + e + \prod_{f}^{g} h_f + i}{\int_{j}^{k} m(u)\ du} + n + \sum_{p} q$$
$$= \int_{r}^{s} t(v)\ dv$$

Figure EQU-6 (continued) -- The Behavior of NEXT.



NEXT:

$$\sin(x) = \frac{b + \sum_{c} d_c + e + \prod_{f}^{g} h_f + i}{\int_{j}^{k} m(u)\, du} + n + \hat{\sum_{p}} q$$
$$= \int_{r}^{s} t(v)\, dv$$

NEXT:

$$\sin(x) = \frac{b + \sum_{c} d_c + e + \prod_{f}^{g} h_f + i}{\int_{j}^{k} m(u)\, du} + n + \boxed{\sum_{p}} q$$
$$= \int_{r}^{s} t(v)\, dv$$

NEXT:

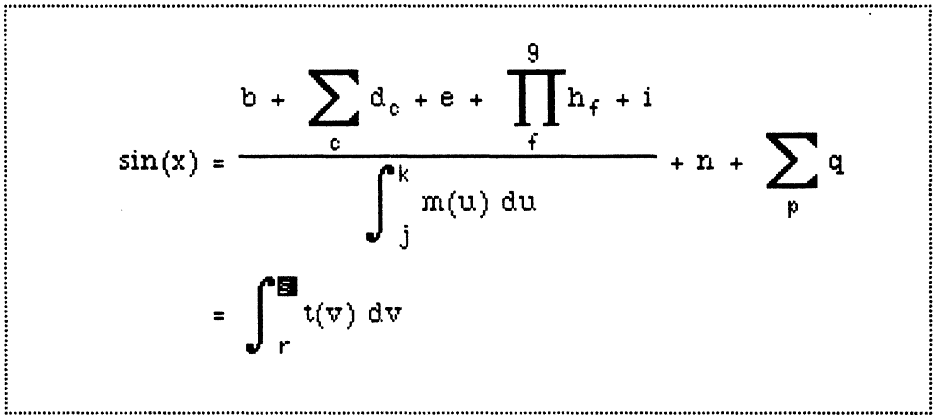
$$\boxed{\sin(x) = \frac{b + \sum_{c} d_c + e + \prod_{f}^{g} h_f + i}{\int_{j}^{k} m(u)\, du} + n + \sum_{p} q}$$
$$= \int_{r}^{s} t(v)\, dv$$

NEXT:

$$\sin(x) = \frac{b + \sum_{c} d_c + e + \prod_{f}^{g} h_f + i}{\int_{j}^{k} m(u)\, du} + n + \sum_{p} q$$
$$= \int_{\boxed{r}}^{s} t(v)\, dv$$

Figure EQU-6 (continued) -- The Behavior of NEXT.

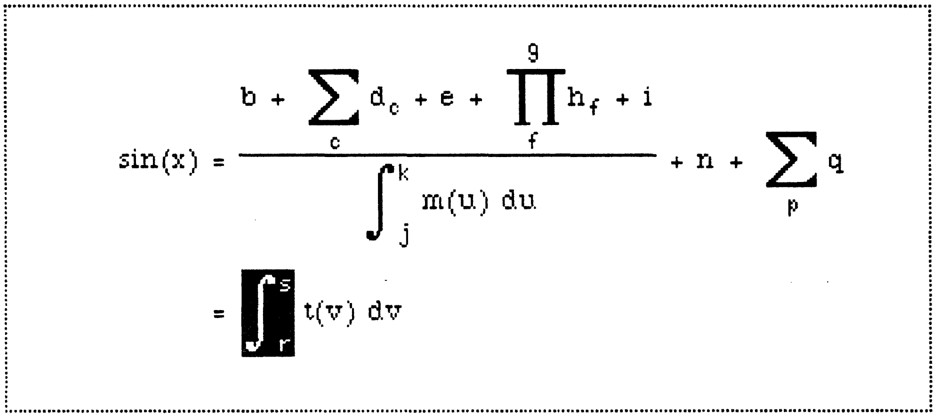
NEXT:



$$\sin(x) = \frac{b + \sum_{c} d_c + e + \prod_{f}^{g} h_f + i}{\int_{j}^{k} m(u)\, du} + n + \sum_{p} q$$

$$= \int_{r}^{s} t(v)\, dv$$

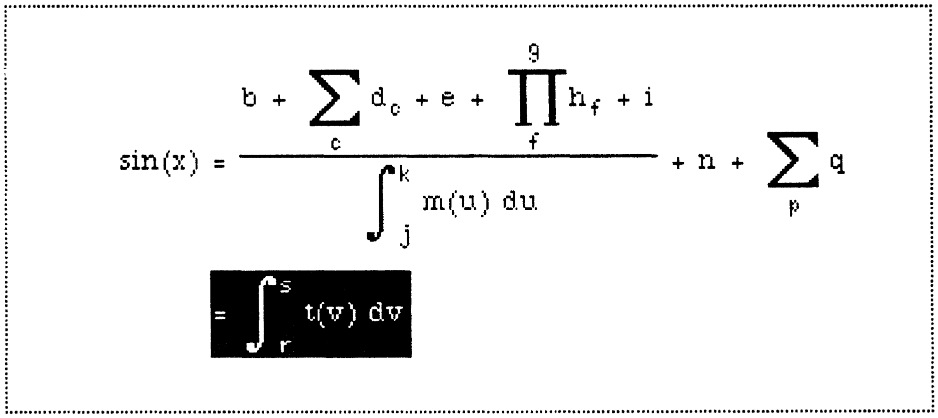
NEXT:



$$\sin(x) = \frac{b + \sum_{c} d_c + e + \prod_{f}^{g} h_f + i}{\int_{j}^{k} m(u)\, du} + n + \sum_{p} q$$

$$= \int_{r}^{s} t(v)\, dv$$

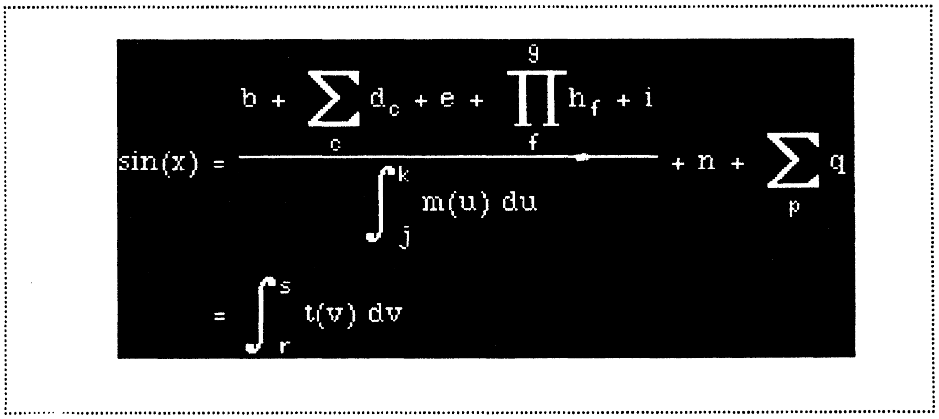
NEXT:



$$\sin(x) = \frac{b + \sum_{c} d_c + e + \prod_{f}^{g} h_f + i}{\int_{j}^{k} m(u)\, du} + n + \sum_{p} q$$

$$= \int_{r}^{s} t(v)\, dv$$

NEXT:



$$\sin(x) = \frac{b + \sum_{c} d_c + e + \prod_{f}^{g} h_f + i}{\int_{j}^{k} m(u)\, du} + n + \sum_{p} q$$

$$= \int_{r}^{s} t(v)\, dv$$

Figure EQU-6 (continued) -- The Behavior of NEXT.

(THIS PAGE INTENTIONALLY BLANK)

19. DOCUMENT FILING

Document filing features provide simple, but flexible, mechanisms to store, retrieve, and organize documents. The structures and actions defined in this section are used throughout Star to operate on collections of documents.

Via the Ethernet, file storage can be distributed and easily shared. One user can access documents stored on file servers by other users from any workstation connected to the Ethernet including Star and the 860 IPS.

USER OVERVIEW

Documents are stored in *containers* that are analogous to those found in an office environment, namely, *file drawers* and *folders*. Documents, record files and folders are *file objects* that can be manipulated using document filing features. These containers and file objects are organized hierarchically, in that a file drawer may contain documents, record files, and/or folders, and folders may contain documents, record files, and/or other folders.

A container such as a folder can be *opened* so that its contents can be accessed via a *File Window,* which is similar to a Document Window. When access is no longer required, the container can be closed, at which point its representation on the display is reduced to iconic form.

There are no special "store" and "retrieve" commands. The user is never required to type a document's name to retrieve it. The basic mechanisms for manipulating documents and other file objects are the MOVE, COPY, and DELETE commands. These actions all operate on the current *selection*, which can be a single file object or a sequence of file objects. A FIND command is also provided for more automated retrieval. Using FIND, the user is able to isolate one or more file objects which satisfy certain logical criteria.

The document filing system also provides facilities for maintaining the attributes of containers and file objects, such as the order in which the contents of a folder or a file drawer are sorted, the creation date of a document, etc. These attributes reside on the *property sheet* associated with each object.

When contained in a file drawer, file objects can be shared among many users. The System Administrator can assign varying levels of *access rights* to a file drawer relative to specific users. Thus, the security associated with a file object is determined by a combination of user identification and that user's access privileges.

Reference icons were discussed in the *Desktop* section of this document. There we stated that some icons can have both a "real" and a "reference" form. The user facilities for creating and manipulating the reference form of data icons are described in this section. Reference data icons refer to real data icons in a file drawer. They permit filing containers to be *cross indexed* by creating multiple references to the contents of the containers. They can also be used to keep references to large documents, folders and record files (such as dictionaries) without incurring the storage overhead required by the actual objects.

DETAILED SPECIFICATION

OBJECTS

Container Objects

File drawers and folders are the objects used for storing file objects. The relationship among containers is basically hierarchical. File drawers may contain folders, documents and record files. Folders may contain documents, record files and other folders. Documents and record files are at the bottom of the hierarchy and may not contain other file objects (exception: see *Records Processing*).

File Drawers - File drawers are represented on the Desktop by file drawer icons. All file drawers are *owned* and can be *shared*. The owner or System Administrator are responsible for initially assigning the *access rights* of other users to the drawer. Access Rights are set at the file server control terminal.

Unlike documents and folders, a file drawer should be thought of as a reference to a collection of objects rather than as a concrete object itself. The actual storage for file drawers is provided by a file server. Access to file drawers is provided through the centralized *Directory* (see *Directories*).

Folders - Folders may reside in a file drawer, in another folder, in the Directory (in the case of the Abbreviations folder and the dictionary folders), or on the Desktop (as well as inside function icons such as Out Baskets and Printers). The ability to place folders within other folders provides as many levels of hierarchy as the user requires, without introducing additional objects.

File Objects

File objects (those which can be filed, i.e. contained in file containers) are Star documents, folders, record files, reference icons for any of these types, non-Star documents and non-Star record files. Note that folders are the only objects which are both container objects and file objects.

Any of these container and file objects may exist on the Desktop, independent of any container. Just as a document may reside on the Desktop in iconic or window form, so can a folder or file drawer.

File Window

A File Window displays the contents of file drawers and folders. The same window may also be used to display the contents of file objects inside the window. When the contents of a document are visible, the File Window becomes a Document Window (see *Documents*). Likewise when the contents of a record file are displayed, it becomes a Records Processing Window (see *Records Processing*). The only visible changes in the window are the menu and the title, and perhaps the window width.

The title of the File Window indicates the name and type of object whose contents are being displayed. For example, if a file drawer icon is opened on the Desktop, the resulting file window title includes a small (13x13 pixel) file drawer icon and the name of the file drawer. In this case, the main display area of the window shows the folders and documents contained in the file drawer. If one of the folders is subsequently opened, a small folder icon and the folder name are appended to the title of the window, and the window shows the contents of the folder (see Figures FIL-1 and FIL-2).

The contents of file drawers and folders shown in the window are displayed in the form of Object Descriptors. Each descriptor is composed of a small (16x16 pixel) file object icon, followed by fields that describe the object: name, size, last change date and create date. The size is not shown unless the SHOW SIZE parameter is on in the Set Window property sheet (its default value is off). As for other Set Window parameters, the current state is maintained until changed by the user. The size units depend upon the type of object described:

Documents: number of pages at last pagination;
Folders: number of level 1 items contained;
Record files: number of records.

The File Window contains scroll and thumb areas, as described in *Desktop*. The window menu always contains the commands ? and Close; the auxiliary menu always contains the Make Document and Set Window commands. A Close All command appears in the menu whenever a file object is open inside the window (e.g., an open document within an open folder). Two other commands, Show Next and Show Previous appear in the window menu when a file object is opened from within a file window. When invoked, these commands cause automatic closing of the current file object and the selecting and opening of the immediately following or preceding file object. In addition, if a file drawer or any of its folder contents are opened, a Redisplay command is shown in the file window. This is for the purpose of refreshing the window contents that have changed in a shared environment (see *Sharing*). Other commands become visible in the menu as appropriate according to the type of object being displayed in the window.

Document Property Sheet

The properties or attributes associated with a document that have to do with filing exist on the Document Property Sheet (see Figure FIL-3).

Name - The name of the document is a text parameter. It is also part of the corresponding document descriptor, shown in the File Window or the window title. The name of a document may be modified only when it is visible on the property sheet, however.

A document name is a string of up to 100 characters (actually 100 bytes). Any character in the OIS Character Set may be used, including spaces. Practical use of these characters is limited only by which characters have representations in the workstation's system font (which could be quite large for some installations). The OIS Character Set is described in detail in *System Overview.* Document names need not be unique within the file system.

Miscellaneous Properties - Several other properties are displayed. These only provide information and cannot be directly modified by the user. They include:

> **Created On** - document creation date and time. This parameter is set by the system when the document is created via a Copy command.
> **Created By** - user who created the document. This parameter is set by the system when the document is created via a Copy command.
> **Last Change** - date and time of last modification (any change to contents or properties).
> **Changed By** - name of user who last modified the document. (The system automatically sets this to the user's name when a user changes the contents or properties of the document.)
> **Size as of Last Paginate** - shows the number of pages in the document.

Folder Property Sheet

Folder properties that pertain to filing exist on the Folder Property Sheet (see Figure FIL-4).

Name - The name of the folder is a text parameter. Its contents are restricted in the same ways as the name of a document.

Uniquely Named Contents - This is a state parameter indicating whether the objects placed in this folder must be uniquely named. If it is on, no file object whose name matches an object already existing in the top level contents of this folder can be moved or copied into the folder. Turning on this state parameter is allowed only if the folder is empty or the current contents happen to be all uniquely named.

Sorted By - The potential sort order of the documents and/or folders contained in the folder is defined by a combination of text and choice parameters. The first choice parameter indicates whether the folder is to be SORTED or UNSORTED. If the SORTED choice is made, other text and choice parameters appear which define the sort order. The text parameter may contain the names of fields (properties) on the property sheets for documents and folders: Name, Created On, Created By, Last Change, Changed By. The contents of these fields are used in sorting the object descriptors in the File Window.

Beside the name of each field is a choice parameter indicating whether the sort on the field is to be in ascending or descending order. The choices are labelled "A-Z" and "Z-A". When the sorting of a container occurs, it is done in the order that fields are listed. For example, the sort order specified in Figure FIL-4 would list the contents alphabetically by name. Those with the same name would be further sorted by Last Change date, the most recently changed first.

Miscellaneous Properties - Other information is displayed, which may not be directly changed by the user. The items shown are the following:

Created On - folder creation date and time.
Created By - user who created the folder.
Last Change - date and time of last modification. This applies only to the first level of contents. It does not reflect, for example, changes within contained documents, or changes to the contents of contained folders.
Changed By - name of user who last modified the folder.
Number Of Contained Items - the number of items contained at the top level in this folder.

File Drawer Property Sheet

The File Drawer Property Sheet contains information-only parameters. It appears when Show Properties is invoked on a file drawer icon on the Desktop (see Figure FIL-5). The parameters are set by a System Administrator at the file server.

Name - The name of the file drawer is a text string assigned at the time it is registered in the file server. Within a file server, filedrawer names must be unique.

File Server - This is the name of the file server in which the file drawer has been registered.

Space Allocated - This indicates the upper bound on storage for this file drawer.

Owner - The name of the user with owner status.

Sorted By - The sort order of the contents is based on named fields, and is similar to the sort order for folder contents described above.

Uniquely Named Contents - This is the same as for folders.

Miscellaneous Properties - Other miscellaneous information is displayed. The items shown are the following:

Created On - file drawer registration date and time.
Created By - user who registered the file drawer (a System Administrator).
Last Change - date and time of last modification, with same restrictions as for a folder.

Changed By - name of user who last modified the file drawer.

Number Of Contained Items - the number of items directly contained in this file drawer.

Access Rights - Shown in Figure FIL-5 is an informational text version of the Access Rights choice parameter with the choices being None, Read, and Full (see Security). Access rights are specified for each user in the column to the right (including the owner) and for "Everybody Else."

Users - This is a list of names of users (see *Directories).* It is used to designate which users are to have specific types of access to the file drawer contents.

Security

Security of filed information in Star is maintained primarily by access rights to a file drawer relative to individual users. Such access rights can be assigned only to file drawers. The assigned access rights apply to all of their contents. The security discussion below is specific to file drawers. File objects on individual Desktops are private to those Desktops.

The levels of access rights for a file drawer are increasingly inclusive, i.e., a higher level of access provides the same capabilities as a lower level of access rights, plus some additional capabilities. The specific levels of access rights are defined below:

None Access - denies all access to the file drawer's contents and its property sheet.

Read Access - allows a user to view the file drawer contents and their properties. The file drawer contents may be copied if the user also has Full Access to the destination container. (A user always has Full access to his Desktop.)

Full Access - allows a user to move, delete, or change the properties and contents of the file drawer.

Each file drawer has an *owner*. The owner is assigned when the file drawer is created at the file server and can only be changed by the owner or a System Administrator.

Access rights to a file drawer can be assigned to the 1) owner of the object, 2) an access list of users and 3) "everybody else". This is done at the file server.

Reference Data Icon

A *reference data icon* is an icon that "refers to" a separate data icon in a file drawer. It can refer only to a real data icon, never to another reference icon. Inside the reference icon is a pointer to the referenced object; this pointer has no manifestation to the user. Part of the pointer is the file drawer in which the referent resides. The referent must remain in that file drawer (although not necessarily in the same place) in order for the reference icon to be valid.

A reference data icon should be considered only as a "likely way" to access an object in a file drawer. If the referent is available, then the user can display it via a reference icon. If it is no longer available, then the display operation fails and the user has no further Star-supported recourse. A referenced object is unavailable if it is not in the same file drawer that it was in when the reference was created, or if the user does not have at least READ access to the file drawer. For example, if the object is now on a Desktop or in some other file drawer, reference icons will not be able to find it. However, if an object that has been moved out of a file drawer is replaced subsequently in that same file drawer (possibly in a different location), the reference icons will then be able to find it. (If the object is not in *exactly* the same location in the file drawer, Star will automatically initiate a search for it in the drawer. The user is warned that such a search is taking place and is given the option of stopping it with the STOP key. If he does stop it, the referent remains unavailable.) Star generates an error message if an attempt is made to *open* or *print* a reference to an unavailable object. All other actions on reference data icons do not depend on the availability of referents.

As icons, reference data icons have all the status of real data icons. They can be opened, closed, filed, mailed, printed, moved to Desktops, etc. They have the same property sheets as their referents; in Star-1 these properties cannot be changed through the reference; i.e., all properties appear as information-only. One new property is added to reference icons: CONTAINED IN. This shows the File Server/File Drawer/Folder hierarchy in which the real data icon is expected to be found. This tells the user where to get the real icon should he wish to do so.

ACTIONS

Open

Open allows the user to view the contents of file drawers, folders, record files, documents and so forth. It is invoked as described in *System Overview*. When a folder or file drawer icon is opened, a File Window is created showing the file objects contained in it.

Additional file objects can be opened in a File Window. For example, a folder icon on the Desktop can be opened, showing its contents, followed by opening a folder in the list of object descriptors in the window, followed by opening another folder or document, etc. The icon and name of each open object is appended to the title of the window. The contents of the opened object can be edited if the user has Full access rights to the file drawer containing it; otherwise it is read-only.

The only exception to the normal opening of file objects is in the case of a file drawer that is being simultaneously accessed by more than one user. When a user has opened a document or record file in a file drawer, no other user can open that particular document or record file. (See *Sharing*.)

For the initial product release: Opening a document or record file contained in a remote container (i.e. in a file drawer or in a folder in a file drawer) moves it from that container to the local workstation. The user who opens it has the illusion that it has always remained in that container. But another user who opens the same container will not see the "opened" object. When closed, the object is automatically returned to the remote container. If the container is deleted while the "contained" object is open (i.e. removed) or if the file server is unavailable, closing the "contained" object places it on the Desktop. For a user with Read access, a copy is made; if read-only windows are un-available, the user will be warned that it's only a copy.

After opening the File Window, the user may scroll the object descriptors in the window as described in *Desktop*.

Close

Close is invoked from the window menu as described in *System Overview*. If a single file object is open, this command causes the File Window to be replaced on the Desktop by its icon. If more than one file object is open within the same file window, however, only the last opened object is closed. For example, if a document were opened inside a file drawer, invoking Close will return the document to icon size and redisplay the file drawer contents.

Close All

Close All is invoked from the window menu as described in *System Overview*. This command causes the File Window to be replaced on the Desktop by an icon corresponding to the file object first opened. For example, if a file drawer is open as well as one of its contained folders, invoking Close All causes the window to be replaced on the Desktop by the file drawer icon.

Show Next and Show Previous

These commands are invoked from the window menu as described in *System Overview*. Show Next causes the currently opened object to be closed and the next object in the list to be opened. Show Previous causes the same procedure but opens the preceding object. When the beginning or end of the list of objects has been reached, "There are no more items to be opened" appears in the message window.

Redisplay

The Redisplay command is used to refresh the displayed contents of sharable containers (file drawers and their contents, shared in-baskets, reference folders, etc.). Once invoked, it causes a new set of information to be retrieved and painted on the display as if the file object had just been opened. If possible, the scroll position is maintained with the same object at the top of the window. If any of the contained objects were selected, Redisplay deselects them.

Select

A file drawer, folder, record file, or document icon may be selected on the Desktop or within a File Window. (Note that only icons in the main display area of a File Window may be selected, not those in the window title.) When the SELECT button is pressed, the object under the cursor (if any) is highlighted, and, when the button is released, the object is selected. If the cursor is over an icon in a File Window, the entire object descriptor is inverted.

Within a File Window, the user cannot select or edit the name of an object (either in the window title or in object descriptors in the main display area). The name and other attributes shown in the object descriptors must be changed via the property sheet for that object.

Adjust Selection - Once a selection has been made within a File Window, it can be adjusted to include multiple icons using the ADJUST button. While the button is held down, the selection is extended to the icon nearest the mouse cursor.

Because of sharing, the current display of a file drawer's contents may be out of date. In that case, only those shown in the (possibly out-of-date) display can be selected and operated on. For example, making an extended selection of everything in a container window and invoking DELETE will delete only those shown; objects added by another user since the last Redisplay will not be affected. That is, the ADJUST button does not provide an implicit inclusion of icons not currently shown in the file window.

Deselect - The current selection is deselected whenever any of the following conditions occur:

the selection is deleted,
a new selection is made,
the object containing the selection is closed (the closed object becomes selected),
the currently selected object is opened (the first thing in the window is selected),
the selected object is moved or copied into a closed container,
the file window is refreshed (Redisplay).

Move and Copy

Move and copy are the major actions that manipulate file objects. Both functions operate on the currently selected object(s). They are invoked as described in *System Overview*.

In general, a file object may be moved or copied to a specific location in a list of objects, or it may be placed within a closed object. For example, a document selected on the Desktop can be moved to another place on the Desktop or into a folder icon. If the selection is not placed in a closed object, it remains selected.

The ability to move or copy documents and folders both into and out of file drawers and their contained folders is subject to security constraints. Whenever the access rights of a user with respect to file drawer contents would be violated by a move or copy action, an error message is displayed.

The following rules and conditions are enforced regarding the moving and copying of file objects.

Source for Move and Copy - The sources of objects that are to be moved or copied are the following:

> **Desktop Icons** - Any file object in iconic form on the Desktop (a folder, record file or document) can be selected as the operand for move or copy.
>
> **Icons Within a File Window** - Any file object in a File Window can be selected as the operand for move or copy (if the user has appropriate access rights). The contents of the System Divider in the directory may only be copied.

Destination for Move and Copy - The possible destinations of file objects that are to be moved or copied are the following:

> **Desktop Icons** - A file drawer or folder in iconic form can be designated as the destination for all file objects including other folders. Document icons can never be the destination of other file objects. Record files can be the destination of documents or folders (see *Records Processing* for details). The placement of objects in closed icons is determined by the Sort Order on the object's property sheet if the object is currently in sorted order. If the container is not in sorted order, the new file objects are placed at the front (top) of the container's contents.
>
> The Desktop itself can be designated as the destination for a selected object by pointing at an empty spot on the Desktop. The selection is placed on the Desktop in the available icon position(s) nearest the mouse cursor.
>
> Multiple selections can be moved or copied to the Desktop; but, if there is not enough room to display all the icons, or enough storage space for the data, none are moved and an error message is given. If a folder or file drawer is the destination, the multiple objects may "scatter" to match the container's current sort specifications. If they do, they become de-selected; multiple selections inside a file window must be contiguous.
>
> If the folder or file drawer into which the document or folder is to be moved or copied has been designated to have uniquely named contents, the name of the moved or copied object must be unique relative to the top level contents. If such is not the case, the error message is displayed and the move or copy action is aborted.
>
> Documents, record files, and folders can be moved to other Desktop icons, such as the Out Basket and Printer. The effects of these operations are described in the relevant sections.

File Windows - Selected file objects can be placed between icons in a File Window (subject to the container's sort order constraints) or within folder icons in a File Window (unless access constraints preclude such an action). File drawers can never be moved or copied into a File Window.

As in moving or copying to an icon, if the uniqueness of contents property is on, no duplicate names are allowed in moving or copying to a File Window.

Cursor Appearance - Once a selection is made, the MOVE or COPY key is pushed, and a mouse button is depressed, the cursor takes on the shape of the selected icon. (When more than one object is selected, the mouse cursor looks like a folder.)

If the cursor is over a Desktop icon that can legally contain the selection, the icon flashes. It continues to flash until the mouse cursor is moved off the icon or the mouse button is released. If the mouse button is released, the selection is placed within the object represented by the icon.

If the mouse cursor is inside a File Window, a horizontal, right-pointing arrow appears in addition to the cursor. The arrow is constrained to move in increments such that it always points at legal destinations. It points either between object descriptors or directly into a folder icon. If the arrow points into a folder icon, the icon flashes until either the mouse button is released or the cursor is moved to a different location. If the mouse button is released while the icon is flashing, the selection is deposited within the icon and the selection is deselected. If it is released while the arrow is between icons, the display is updated to show the selected object(s) in their new positions, and the object(s) remain selected.

In the case of icons being moved or copied into sorted containers, the final destination of the selected object(s) is subject to sort field criteria. In this case, the horizontal pointing arrow is still used, but the moved or copied object(s) will very likely not be deposited in the location between icons pointed to by the user.

If the cursor enters a document window or any other window which cannot contain the selection, the cursor becomes a question mark. If the mouse button is released while in this state, the move or copy command is aborted with an error message.

After the mouse button is released, the cursor returns to its normal shape.

Delete

The Delete function is used to eliminate file objects. It is invoked by making a selection, either on the Desktop or within the File Window, and pushing the DELETE key.

The effect of deleting a file drawer icon is slightly different than deleting file objects. File drawer icons should be thought of as references to a collection of objects rather than concrete objects. Thus, deleting a file drawer icon from the Desktop only eliminates the reference rather than the data contained in the drawer; the file drawer icon disappears, but its contained objects remain in the system. The contents of the file drawer become available again if the user obtains another copy (reference) of the file drawer from the Directory. Actual deletion of a file drawer and its contents is done by a System Administrator at the file server (see *Directories*).

The ability to delete file objects from a file drawer depends on the access rights of each user. Only those users with Full access rights may do so. Any user with insufficient privileges who tries to delete an object from a file drawer, will receive an error message. The delete action is ignored in this case.

The UNDO command can be used to retrieve the last deleted object if no intervening commands have been used.

Make Document

The Make Document command in the File Window auxiliary menu is used to create a new document containing a textual representation of the contents of the filing window. The created document is a snapshot only; it is not updated automatically if the object changes. When the command is invoked, a document is created and a small document icon is attached to the cursor as if the document had been moved or copied. The user places the document at the desired destination by pointing with the cursor, and pressing either mouse button.

Sharing

Sharing of file objects is restricted to those file objects contained in a file drawer. Because of the potential distribution of users accessing the contents of file drawers, feedback of one user's actions cannot be immediately reflected to other users. Invoking Redisplay will refresh a file window with the current contents. For example, a document may have already been deleted from a file drawer but is still visible in iconic form on someone else's display. Any attempt to move, copy, delete, or show properties for such an object will result in an error message.

With sharing, users at different workstations can simultaneously open the same file drawer or folder (if contained in a file drawer) but not the same document or record file. When a user has opened a document or record file in a file drawer, no other user can open that particular document or record file.

Find

The Find command allows the user to locate documents and/or folders based on fields associated with those objects. It is similar to the search facility described in *Text Editing*. It is invoked by opening either a file drawer or folder, making a selection, and pushing the FIND key. This causes a Find Option Sheet to appear (see FIL-6).

The Find Option Sheet displays the following parameters:

Search For - This is a choice parameter indicating whether the search is for a document, record file, folder, any of these three, or a document containing fields.

With (field) / Matching - This is a table of text parameters that defines the conditions that must be met to satisfy the search. It is similar to the specification of filters associated with record files (see *Records Processing*). At the top of the table are listed those fields that can be used in the search.

> When the search is for a DOCUMENT WITH FIELDS, the columns are initially blank; the user enters the field names and values to search for (see FIL-7). Fill-in is by column; when the user invokes NEXT in the last (second) field in a column, a new column is created and the caret placed in it.
>
> Otherwise the column headers contain the following fields: Name, Created On, Created By, Last Change and Last Changed By. The user types in attributes to search for. Fill-in is by row.

In either case, the legal search patterns are as defined in *Field Definition*.

Range - This is a choice parameter that indicates over what file objects the search is to be done. The choices are Entire Contents, Rest of Contents, and Current Selection.

Once the Start menu command on the Find Option Sheet has been executed, searching begins. It continues down through the hierarchy of file objects until a match is found. (The search direction is first through the list of objects at the top level and then down through the container levels as necessary. For the initial product release, the search is only through one level) When an object is located which matches the search criteria, the window changes as necessary to show the object in iconic form in its normal location in the filing hierarchy. This may involve opening and/or closing other file objects and scrolling such that the object is visible. The icon is selected in the window. The user may then manipulate the File Window contents in the normal fashion.

When searching for a DOCUMENT WITH FIELDS, each document that is encountered is examined to see if it contains the fields named in the search criteria. If so, then the contents of those fields are matched against the corresponding field patterns. If all patterns match, then the document icon is displayed in the window, as above.

After a match is found, the search for others matching the same criteria can be continued by pressing the FIND key again as described in *Text Editing*. A Search may be stopped at any time by depressing the STOP key.

Show Properties

The user can display the properties associated with a selected file object by pushing the PROPERTIES key. This causes a property sheet to be displayed, according to the kind of object selected. If more than one object is selected and Show Properties is invoked, the following error message is issued: "Multiple objects are selected -- show properties is illegal." If the user does not have Full access to the object selected (or is not its owner), and he attempts to change a parameter on the property sheet, an error message is displayed.

Copy Properties

The SAME key has no effect on File Drawer, Folder or Document icons.

Create

New file objects are created by copying an existing object. To create a new object, the user selects the icon corresponding to the desired object and copies it to the Desktop or to any other legal destination. (The rules associated with the copy function apply as described above.)

Documents, record files, and folders may be copied to a location and used immediately. These types of objects have exactly the same contents and properties as the source from which they were copied (except Created On, Created By, Last Change, and Changed By).

In the case of file drawers, however, an existing file drawer can only be copied from the Directory (see *Directories*). This establishes a reference to the contents of the file drawer. New (i.e., not previously defined) file drawers are created via a registration process at the file server.

Sorting

A sort order can be specified for the contents of file drawers and folders. The sorting of these objects is done using the same fields defined in Searching above. The characters in these fields are sorted based on the standard collating sequence (see *Keyboard Sets*). Fields that are empty or nonexistent are treated as a character before the first character of the collating sequence.

Create Reference Data Icon

A reference data icon can be created by selecting any real file object in a file drawer and invoking the Make Reference command in the file drawer/folder auxiliary menu. Make Reference is similar to the Make Document command. When invoked, a reference data icon is created and attached to the cursor as if it had been moved or copied. It is placed at a destination by pressing either mouse button. It remains selected for convenience in a further Move or Copy.

A reference data icon can also be created by copying an existing one. The new one will point to the same data icon as the old one. The reference cannot be changed.

Open, Close, Move, Copy, Delete, File, Mail, Print, etc. Reference Data Icon

All of the actions applicable to real data icons apply to reference data icons. They affect the reference icons themselves and not their referents. The only actions that affect referents are:

Open - When a reference data icon is opened, *the contents of the referent are displayed in the window.* The window appears as a normal window for that type of object, except that the window title contains the reference icon and name. This is exactly equivalent to opening a data icon inside a file drawer. A user may open a reference data icon if he has at least READ access *to the drawer containing the referent.* If he has Full access to it, he may change its contents. When the user closes the window, it returns to a reference icon. (Note that reference data icons do not give users any extra sharing capability; although several users may have reference icons to a document or record file, it is still true that only one may have it open at a time.)

Print - When a reference data icon is printed, *the contents of the referent are printed.* A user must have at least READ access *to the referent* in order to print a reference data icon. If he has only READ access to it, all print-time actions which would alter its contents (such as REPAGINATE) are ignored. (Aside: A folder can be modified *simultaneously* by two or more people. The implementation will have to worry about printing such a folder, just as it has to worry about copying it.)

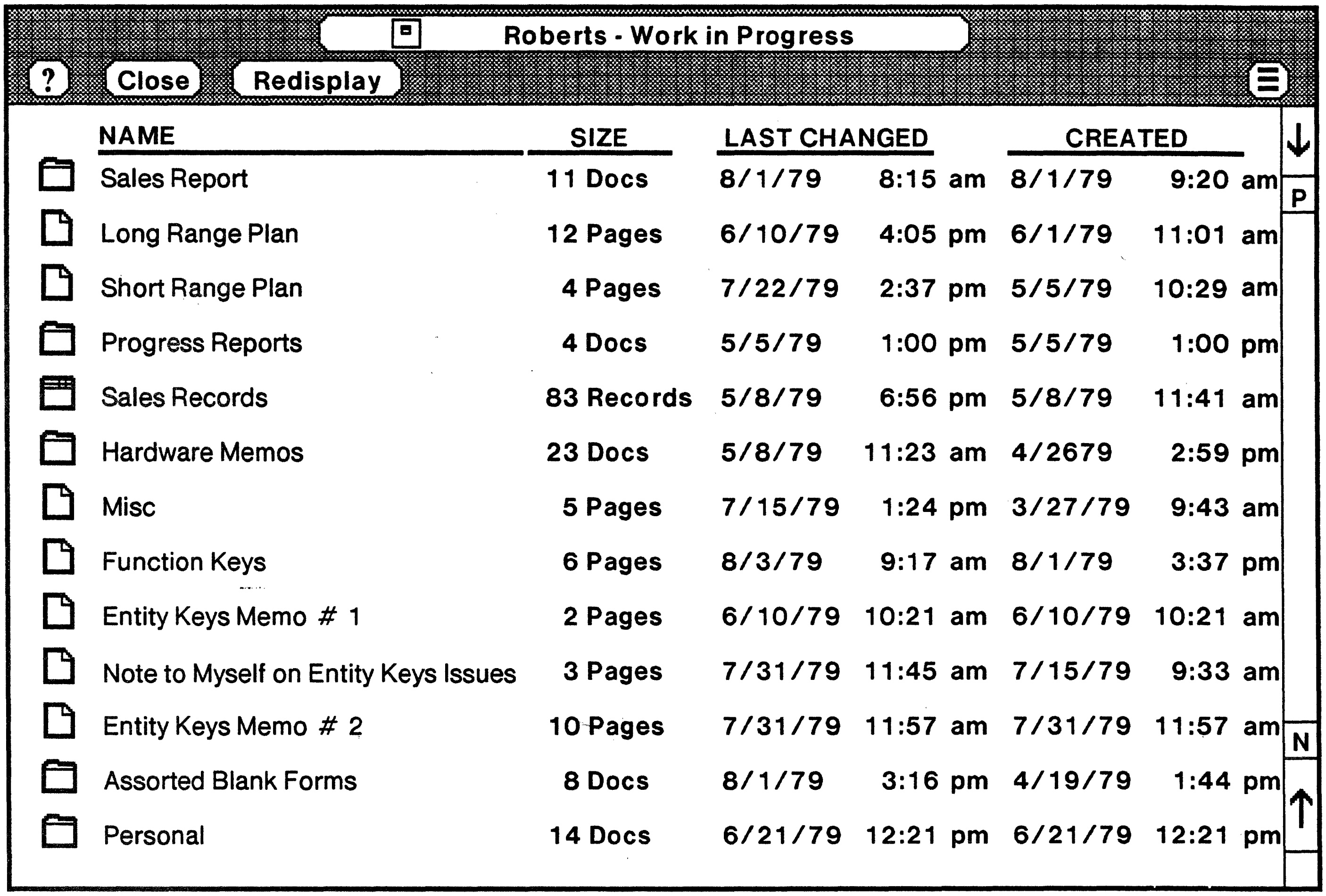
There is no way to move, copy or delete a real document, folder or record file with a reference icon. The user must go to the real version in the file drawer if he wants to do that.

Show Properties of Reference Data Icon

All of the properties of real data icons are the same for reference data icons. In Star-1, for the reference icon, they are information-only and cannot be changed.

Create Cross Index

Multiple citations to documents, folders and/or record files can be constructed by creating reference icons referring to them and placing the references in a folder or file drawer. Such a set of references is called a "cross index." There is no limit (except for physical resources) to the number of cross indices that can be created, nor to the number of times that a given document, folder or record file can be referenced.

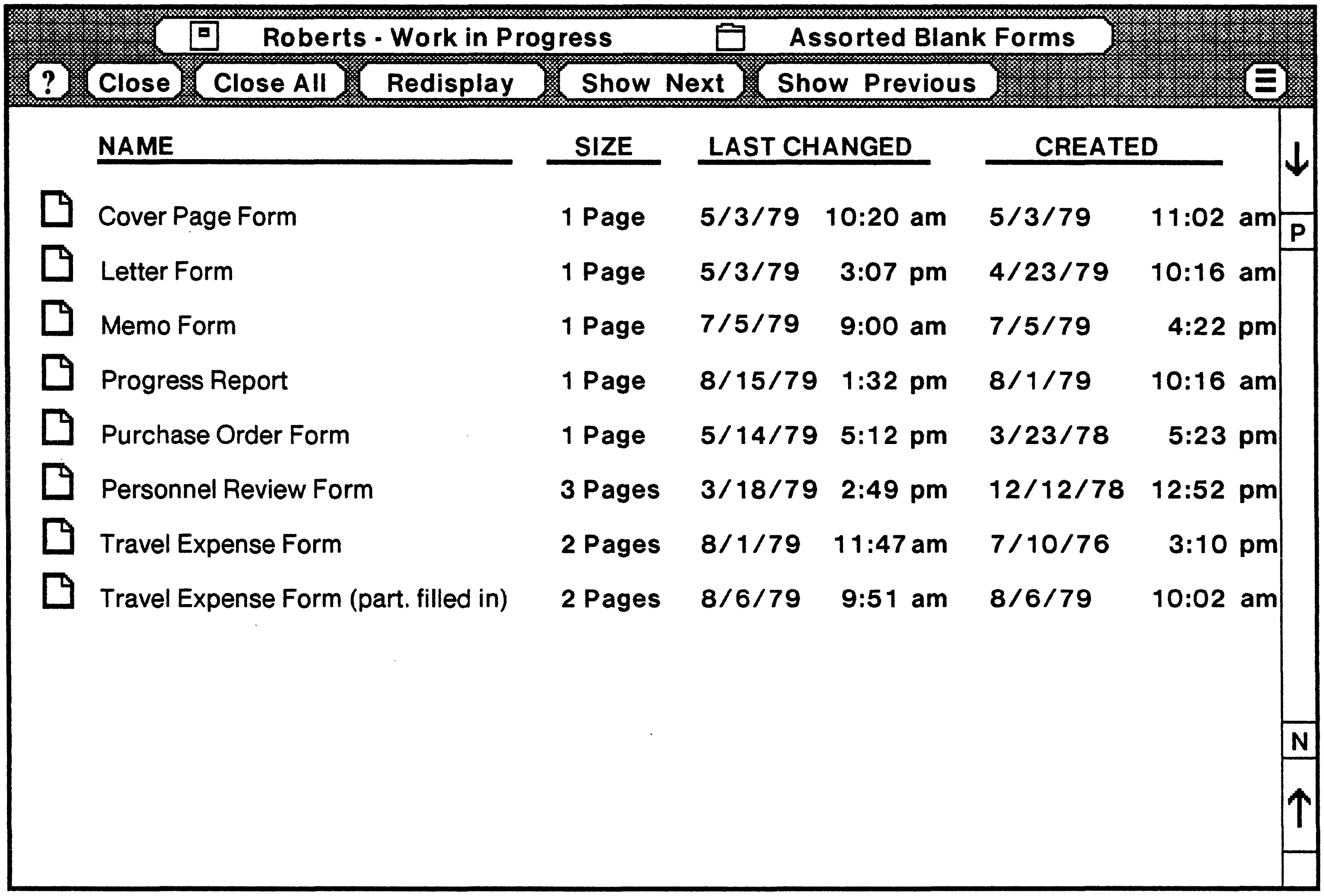


NAME	SIZE	LAST CHANGED		CREATED	
Sales Report	11 Docs	8/1/79	8:15 am	8/1/79	9:20 am
Long Range Plan	12 Pages	6/10/79	4:05 pm	6/1/79	11:01 am
Short Range Plan	4 Pages	7/22/79	2:37 pm	5/5/79	10:29 am
Progress Reports	4 Docs	5/5/79	1:00 pm	5/5/79	1:00 pm
Sales Records	83 Records	5/8/79	6:56 pm	5/8/79	11:41 am
Hardware Memos	23 Docs	5/8/79	11:23 am	4/2679	2:59 pm
Misc	5 Pages	7/15/79	1:24 pm	3/27/79	9:43 am
Function Keys	6 Pages	8/3/79	9:17 am	8/1/79	3:37 pm
Entity Keys Memo # 1	2 Pages	6/10/79	10:21 am	6/10/79	10:21 am
Note to Myself on Entity Keys Issues	3 Pages	7/31/79	11:45 am	7/15/79	9:33 am
Entity Keys Memo # 2	10 Pages	7/31/79	11:57 am	7/31/79	11:57 am
Assorted Blank Forms	8 Docs	8/1/79	3:16 pm	4/19/79	1:44 pm
Personal	14 Docs	6/21/79	12:21 pm	6/21/79	12:21 pm

Figure FIL-1 - Open File Drawer

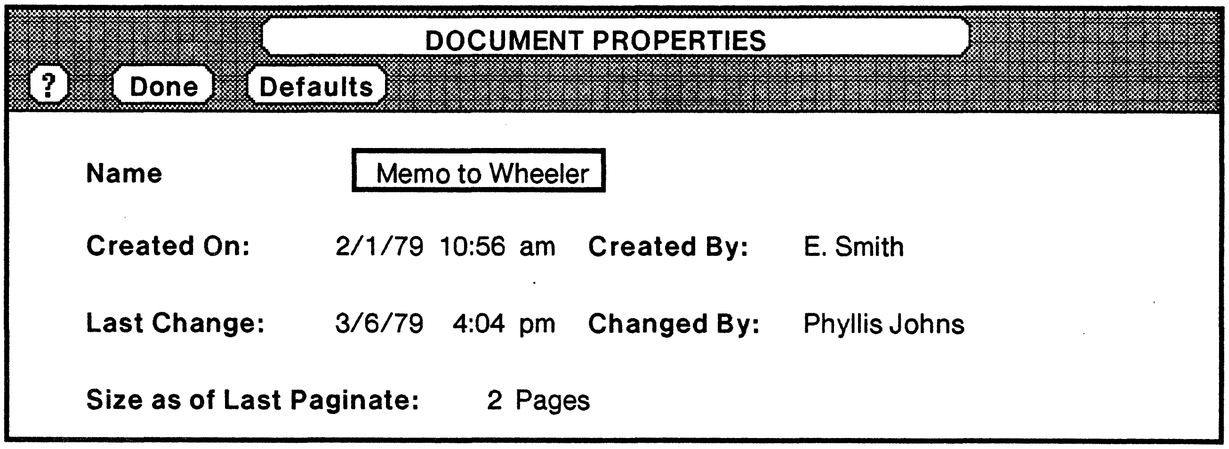
auxiliary Folder and File Drawer menu:

Make Document
Set Window



NAME	SIZE	LAST CHANGED		CREATED	
Cover Page Form	1 Page	5/3/79	10:20 am	5/3/79	11:02 am
Letter Form	1 Page	5/3/79	3:07 pm	4/23/79	10:16 am
Memo Form	1 Page	7/5/79	9:00 am	7/5/79	4:22 pm
Progress Report	1 Page	8/15/79	1:32 pm	8/1/79	10:16 am
Purchase Order Form	1 Page	5/14/79	5:12 pm	3/23/78	5:23 pm
Personnel Review Form	3 Pages	3/18/79	2:49 pm	12/12/78	12:52 pm
Travel Expense Form	2 Pages	8/1/79	11:47am	7/10/76	3:10 pm
Travel Expense Form (part. filled in)	2 Pages	8/6/79	9:51 am	8/6/79	10:02 am

Figure FIL-2 - Open Folder within an Open File Drawer



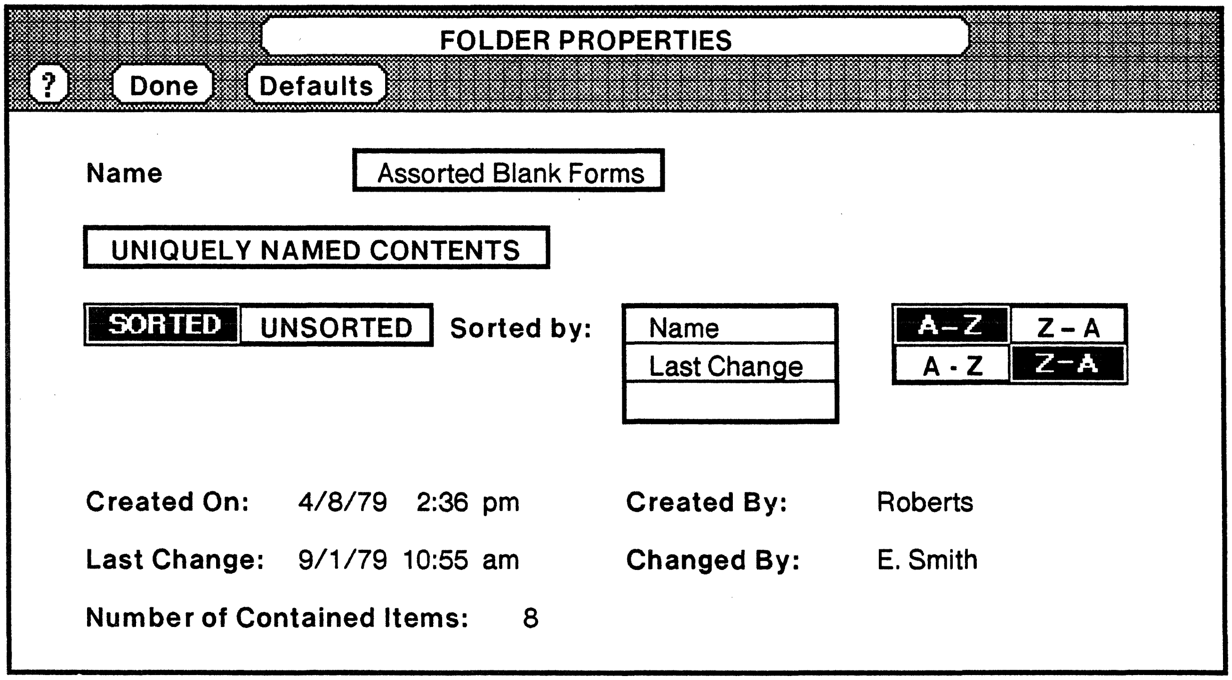
Name: Memo to Wheeler

Created On: 2/1/79 10:56 am Created By: E. Smith

Last Change: 3/6/79 4:04 pm Changed By: Phyllis Johns

Size as of Last Paginate: 2 Pages

Figure FIL-3 -- Document Property Sheet



Name: Assorted Blank Forms

UNIQUELY NAMED CONTENTS

[✓ SORTED] UNSORTED Sorted by:

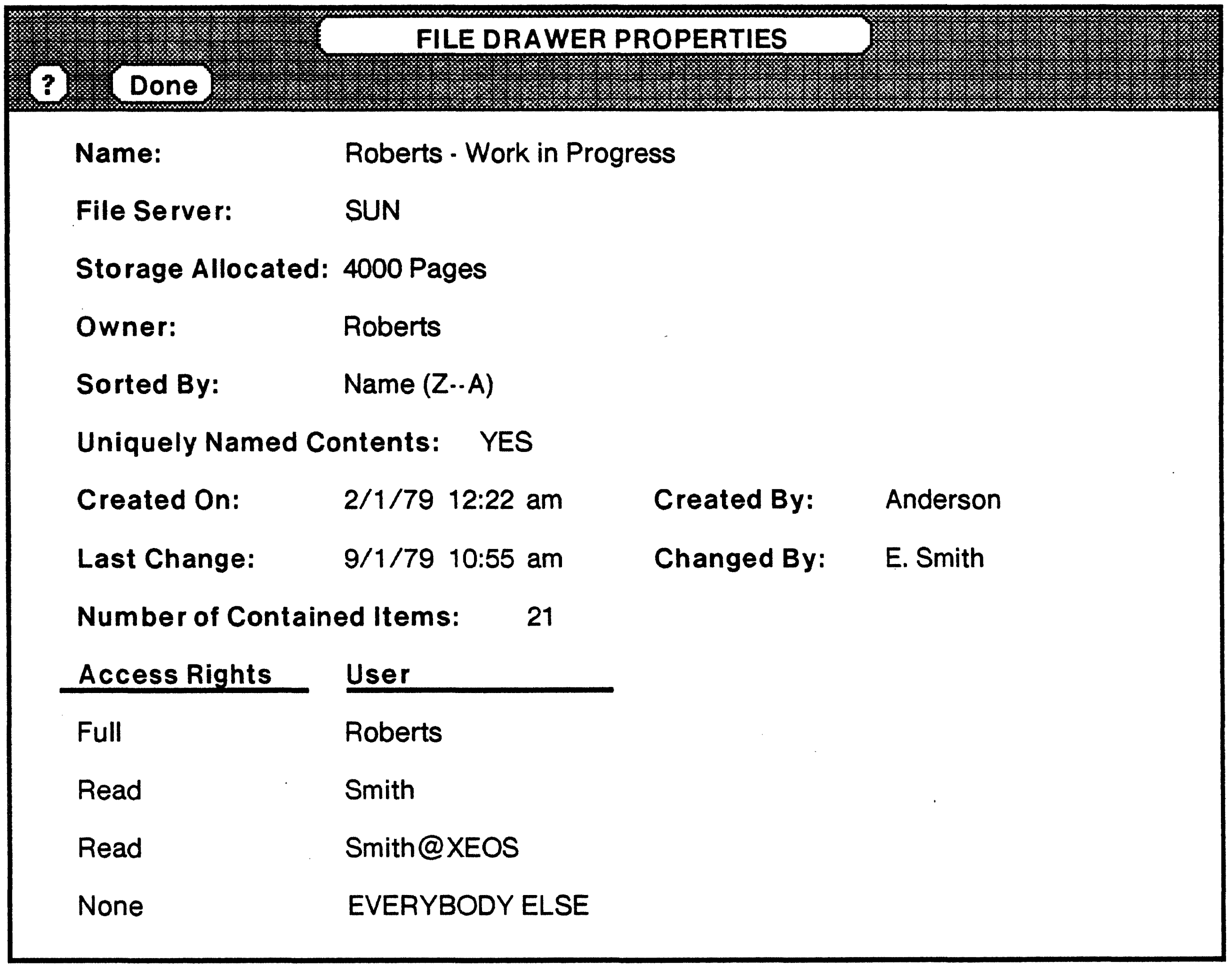
Name	[✓ A-Z]	Z-A
Last Change	A-Z	[✓ Z-A]

Created On: 4/8/79 2:36 pm Created By: Roberts

Last Change: 9/1/79 10:55 am Changed By: E. Smith

Number of Contained Items: 8

Figure FIL-4 -- Folder Property Sheet



Name: Roberts - Work in Progress

File Server: SUN

Storage Allocated: 4000 Pages

Owner: Roberts

Sorted By: Name (Z--A)

Uniquely Named Contents: YES

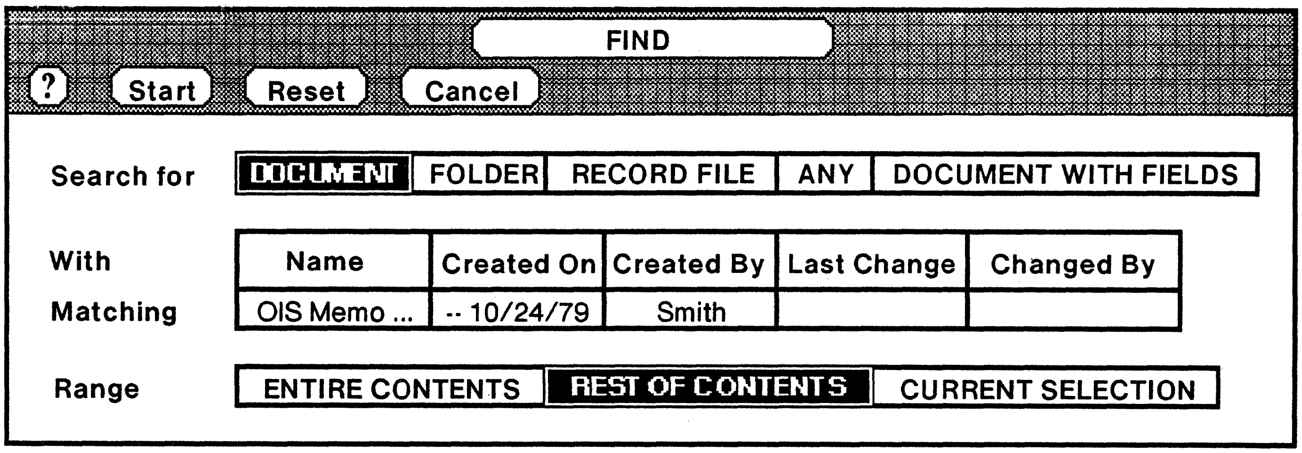
Created On: 2/1/79 12:22 am Created By: Anderson

Last Change: 9/1/79 10:55 am Changed By: E. Smith

Number of Contained Items: 21

Access Rights	User
Full	Roberts
Read	Smith
Read	Smith@XEOS
None	EVERYBODY ELSE

Figure FIL-5 -- File Drawer Property Sheet

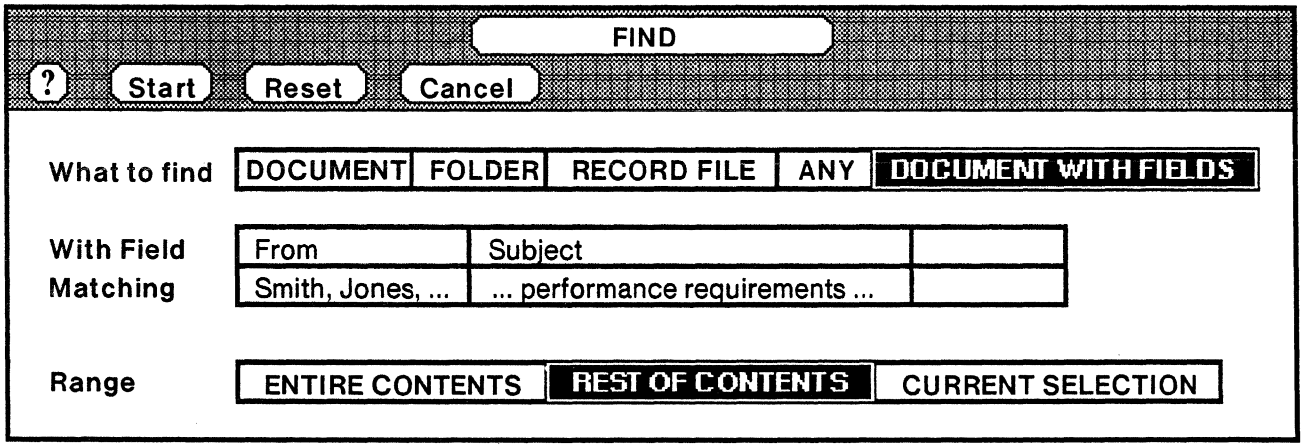


Search for [✓ DOCUMENT] FOLDER RECORD FILE ANY DOCUMENT WITH FIELDS

	Name	Created On	Created By	Last Change	Changed By
With Matching	OIS Memo ...	-- 10/24/79	Smith		

Range ENTIRE CONTENTS [✓ REST OF CONTENTS] CURRENT SELECTION

Figure FIL-6 -- Find Option Sheet for folder and file drawer
(looking for an OIS memo by Smith created on or before October 24, 1979)



What to find DOCUMENT FOLDER RECORD FILE ANY [✓ DOCUMENT WITH FIELDS]

With Field	From	Subject	
Matching	Smith, Jones, ...	... performance requirements ...	

Range ENTIRE CONTENTS [✓ REST OF CONTENTS] CURRENT SELECTION

Figure FIL-7 -- Find Option Sheet for folder and file drawer matching field contents.
(looking for a document from Smith, Jones, and others
containing the words "performance requirements" in a Subject field)

(THIS PAGE INTENTIONALLY BLANK)

(THIS PAGE INTENTIONALLY BLANK)

20. DIRECTORIES

A *directory* provides each Star user with a view of the network configuration in which he is included and of the resources available to him. It provides information about users, devices, facilities, services, and other directories. It is the source of new Desktop function icons. The principal information contained in a directory is derived from the Clearinghouse.

Configuring an installation involves, among other things, specifying to the Clearinghouse Service the names of software services and users of the system, allocating file drawers on file servers, and allocating mailboxes on mail servers. Configuring an installation is the specialized function of a system administrator and is generally not of concern to normal users. It is accomplished almost exclusively at the control terminals of the servers in the internetwork.

USER OVERVIEW

The Ethernet is a high-bandwidth, local communications network to which Star workstations and servers are connected. Two Ethernets can be interconnected via an Internetwork Routing Service, possibly using phone lines, thereby permitting Star workstations and servers on an Ethernet to access servers on other Ethernets. Such an interconnection of Ethernets is called an *internetwork*. Stand-alone workstations, running their own Internetwork Routing Service, can also access servers on an Ethernet via phone lines.

Directories provide access to the resources on the internetwork. Resources are represented in a directory as icons, which are grouped in "dividers" according to their uses, e.g. Filing, Mailing, Printing. The icons in a directory can be copied to the Desktop. For example, in the Printing divider appear printer icons corresponding to printer services available on the internetwork. To access a printer service, the user simply copies a printer icon to his Desktop and uses it in the normal fashion.

Most objects whose names appear in a directory have been *created* by a system administrator at a Clearinghouse Service, File Service or Mail Service and *registered* at the Clearinghouse Service. The *Clearinghouse Service* is a distributed agent which (among other things) maintains a data base of named users and services in the internetwork, such as file servers, printers, users, and public distribution lists. The contents of a directory are derived mostly from the Clearinghouse. A directory also contains information related to devices and facilities located on a Star workstation, such as the local floppy disk drive(s) or the calculator.

The Directory also contains basic Xerox-provided data icons, such as a blank document, a blank record file, a blank folder, a document containing graphics symbols, and places to put user defined abbreviations. These enable users to get started creating their own documents, folders, etc.

A *domain* is a logical grouping of resources on one or more Ethernets, e.g. within an organization or at a given location. There may not be more than one domain on a single Ethernet. This section assumes that only resources on a user's local network appear in his local directory. Resources on other networks appear in remote directories. This enables users to form expectations for the response times of various types of resources. (Remote resources will likely have longer access times.) However, this is only a convention; nothing prevents a system administrator from registering in a local domain Clearinghouse a resource accessed through an Internetwork Router.

By controlling the contents of the Clearinghouse (and therefore of the various dividers in the directory), system administrators can limit access to the internetwork's resources. This affects *all* users in a domain. For example, if the Printing divider contains no printer icons, users cannot print. If the Mailing divider contains no in- or out-baskets, users cannot use electronic mail. If the Remote Directories divider is empty, users cannot access resources in other domains. And so forth. Access controls may also be put on individual icons in some dividers -- i.e., file drawers in a File Server divider and in-baskets in the Mailing divider -- as an additional way to control their access on a per-user basis (see *Document Filing* and *Electronic Mail*). Directories themselves have no access controls; anyone can open a directory or a divider.

DETAILED SPECIFICATION

OBJECTS

System Administrator

A *system administrator* is a person at a customer site who performs administrative tasks at the Communication, File, and Print Servers. There may be more than one such person at any given site, but there is always at least one. He works with Xerox personnel during the installation of new equipment and during system (re)configuration. He registers, modifies, and deletes users, servers and services, and creates and deletes file drawers and mailboxes. All such interaction occurs at server control terminals. See the *Network Administration* specification listed in Section 2.

Clearinghouse

The *Clearinghouse* is a decentralized agent for supporting the naming and locating of a distributed set of objects. Objects may be machines or workstations or file servers or people; all that is assumed about each object is that it has a *name* and a *location* (typically a network address). The role of the Clearinghouse is to maintain a data base that maps names into locations, and to support at least the following primitive operations that use or modify this data base:

locating named objects;

creating, deleting and *changing* the names of objects;

creating, deleting and *changing* the locations of objects;

passing the name of an object from one user to another so that others can access the object.

The mappings maintained by the Clearinghouse are more general than just name-location mappings; the Clearinghouse also supports aliases (alternate names for objects) and various other mappings on names. See the Clearinghouse documentation listed in Section 2.

Network Object

Names for the following types of objects (among others) are registered in the Clearinghouse:

file server -- see *Document Filing*
mail server -- see *Electronic Mail*
print server (i.e. printer) -- see *Printing*
help server -- see *Help and Training*
user (including aliases) -- see below
public distribution list -- see *Electronic Mail*
remote Clearinghouse domain (remote directory) -- see below.

These are called *network objects* because their names are known on a network-wide basis; they are addressable without having to access other objects on a network (as opposed to a file drawer, say, which needs the name of its file server to be complete). Whenever Star *stores* or *manipulates* one of these names, it does so in the name's "fully qualified" form: local name @ domain name @ region name (see *System Overview*). However only as much of the name is *displayed* as is necessary; redundant information is suppressed. Thus the region is suppressed if it is the same as the user's own region; the domain is suppressed if both the domain and region are the same as the user's. (Note: It is possible that the user's domain or region may not be the same as the workstation's domain or region -- see *System Overview*.)

Directory

A *directory* provides access to the network-based resources (among other things) in a domain. The data base that describes these network-based resources is managed by the Clearinghouse Service and the File/Mail Services registered therein. The data is changed via the server control terminals. A directory merely reflects the information in this data base.

There is always a local directory on every Desktop, which lists the resources available in the user's domain. This directory has an icon which can be opened or moved on the Desktop, but which cannot be copied or deleted.

The name of a directory, e.g. "Palo Alto @ Xerox", corresponds to its domain (Palo Alto) and region (Xerox) within the Clearinghouse data base.

A directory is similar to a file drawer. It contains "dividers" (see below) that in turn contain icons or other dividers (but not both), much like folders.

Directory Window (see Figures DIR-1, DIR-2 and DIR-3)

A directory window shows the objects (resources) in the directory's domain. Each object is represented by a descriptor composed of a small icon followed by fields that describe the object. None of the textual information in the window can be edited, although dividers in the window can be opened. A divider appears as a dotted square surrounding a small icon of the type contained in the divider. The only field shown for dividers is the type of contents. For objects inside a divider, various object-specific fields are shown. The contents of each divider are sorted by name. The directory may query the Clearinghouse or other services *when a divider is opened.* Thus the contents of dividers are always up to date. For example, when the Filing divider is opened, the directory queries the Clearinghouse and displays a list of the file servers in the domain. When a File Server divider is opened, the directory queries the file server and displays a list of the file drawers it contains. [Eventually it will display only those file drawers to which the user has access.]

The title of the directory window is composed of a small directory icon, followed by the name of the directory (the domain and region). If a divider is open, a divider icon and the divider name are appended to the title. The directory window contains scroll and thumb areas as described in *System Overview*. The window may be scrolled in the normal fashion. The window menu commands are ?, Close, and Redisplay. Make Document and Set Window appear in the auxiliary menu. When a divider is open, there is also a Close All command as in filing windows.

Directory Property Sheet

Directories have no properties.

Divider

A *divider* is analogous to a folder in a file drawer, except that it may contain function icons rather than data icons. The only purpose of dividers is to provide intermediate structure to the contents of a directory. Dividers may contain other dividers, serving to further subdivide the contents. All the dividers except for the Typing Aids divider are read-only; they cannot be moved, copied or deleted. They may be opened and their windows manipulated in the usual way (e.g. scrolled), but except for the Typing Aids divider, their contents cannot be changed in any way. Dividers have no properties.

There is a fixed set of dividers in a directory. The following are the dividers provided:

Filing contains

- zero or more **File Server** dividers, each of which contains
  - zero or more **File Drawer** icons (see *Document Filing*)
  - fields displayed are: "File Drawers" (the name of each drawer) and "Owner"
- field displayed is: "File Servers" (the name of each file server)

Mailing contains

- one **Mail Baskets** divider, which contains
  - zero or more **In Basket** icons (see *Electronic Mail*)
  - zero or one **Out Basket** icon (see *Electronic Mail*)
  - fields displayed are: "Mail Baskets" and "Owner"
- one **Users and Distribution Lists** divider, which contains
  - one or more **User** icons (see below)
  - zero or more **Public Distribution List** icons (see *Electronic Mail*)
  - fields displayed are: "Name" and "Aliases"
- field displayed is: "Type"

Printing contains

- zero or more **Printer** icons (see *Printing*)
- field displayed is: "Printers" (the name of each printer)

Office Aids contains

- zero or one **Calculator** icon (see *Calculator*)
- zero or one **Spelling Checker** icon (see *Spelling Checking*)
- field displayed is: "Type"

Local Workstation Devices contains

- one **Floppy Disk Drive** icon representing the local floppy drive (see *Removable Storage Media*)
- one **Printer** icon representing the local character printer (see *Printing*)
- field displayed is: "Type"

Device Emulators contains

- zero or one **TTY Emulator** icon (see *TTY Emulation*)
- zero or one **3270 Emulator** icon (see *3270 Emulation*)
- field displayed is: "Type"

Remote Directories contains

- zero or more **Remote Directory** icons (see below)
- field displayed is: "Remote Directories" (the name of each directory)

Basic Documents, Folders, and Record Files contains

- one **Blank Document**
- one **Blank Folder**
- one **Blank Record File**
- one **Basic Graphics Transfers Document**
- field displayed is: "Name"

Typing Aids contains

- one **Abbreviations** folder, initially blank, which may contain
  - zero or more user defined **Abbreviations Documents** (see *Text Editing*)

        field displayed is: "Name"
    field displayed is: "Name"
    zero or more **J-Star Dictionaries** (J-Star only)
    field displayed is: "Name"

The name of each icon in a divider is unique with respect to all the other names in that divider.

Remote Directory

At times it is necessary to communicate between a Star workstation and a service located in another domain on another Ethernet. An example is when a user wishes to use a remote printer. This situation is handled by the Internetwork Routing Service. It is completely transparent to the user, except that the communication may take longer because it may be accomplished through phone lines.

A *remote directory* is a reference to the directory of another domain. It allows users in one domain to access resources in a different domain. For a user in domain A to access an object in domain B, directory B must be present as a remote directory in directory A (i.e. be registered as a domain in A's Clearinghouse).

The Remote Directories divider lists the directory icons for the remote domains that are registered in the user's domain Clearinghouse. A remote directory icon can be copied to the Desktop, opened, and used just like the local directory. From the user's perspective the only difference is that the response of remote resources (e.g. displaying window contents) may be slower.

Remote directories are registered in the Clearinghouse data base only by a system administrator.

Function Icon

A *function icon* represents a resource available from a user's workstation. All function icons are listed in the directory. They may only be *copied* from a directory, never moved, deleted or otherwise altered. The properties of the icons enable them to be used without additional specification once they are on the Desktop.

All function icons are *references* to their resources. They are not equivalent to tangible objects. They serve only to provide access to their associated resources. For example, a file drawer icon on the Desktop provides access to an actual file drawer and its contents on a file server; a printer icon provides access to a physical printer. Deleting a function icon from the Desktop does not affect the corresponding resource, but simply removes the ability to access it. To regain the use of the resource, the user need only copy a new icon from a directory to his Desktop. Opening a function icon performs an automatic "dereference," showing the contents of the corresponding object.

Some function icons, once on the Desktop, may have local data or a queue of objects waiting to be processed, e.g. out-baskets, printers, the calculator, and the spelling checker. (Early versions of Star will not implement background processes.) Attempts to delete these icons from the Desktop when they are non-empty results in a warning message; confirmation is required to actually delete the icon and its data.

Properties of function icons are initially specified when the corresponding resource is registered at a server. These properties can be changed only through the control terminal attached to the corresponding server, and then only by a system administrator. Properties of directory objects cannot be modified via their function icons on a Desktop; their property sheets are information-only, if they exist at all.

Directory Data Icon

The Directory contains a carefully chosen set of data icons in the Basic Documents, Folders and Record Files and Typing Aids dividers. The data icons in the Basic Documents, Folders and Record Files divider may only be copied to the Desktop. They may not be moved, nor deleted, nor may other icons be moved to this divider. Within the Typing Aids divider, the unique folder named Abbreviations will be used for abbreviation expansion, as described in *Text Editing*. The data icons in the Abbreviations Folder may be moved, copied, and deleted.

User

Users are registered with the Clearinghouse Service by a system administrator. Descriptive information about the user is defined at that time. In addition, a certain amount of work space may be allocated on various file servers for his file drawer(s), mailbox(es) and Desktop. Users are represented iconically in the Users and Distribution Lists divider -- contained in the Mailing divider -- in the directory. User icons can reside in a directory, on the user's Desktop, or (eventually) in a document. In a document, the user icon is used primarily for addressing electronic mail.

Each user may have a password.

An *alias* is a shortened form of the permanent user name. For example, the user John Paul Jones might have an alias of "Jones". Aliases have all the status of permanent names. Users can log in as aliases, send mail to aliases, etc. The difference is that aliases may be removed from the system if they are no longer unique. (In the example above, if another Jones joins the domain, the "Jones" alias would -- or should -- be removed.) The system translates aliases into permanent names at log-on, so that a canonical form is always stored by the system. Only permanent names appear in property sheets and filing windows, although aliases can appear in in-basket windows.

ACTIONS

Select

A directory icon is selected like any other icon. Icons in a directory window are selected as described in *Document Filing*.

Open Directory/Divider

A user opens a directory by selecting it and invoking OPEN. This displays the directory's window, showing its dividers. The user can open a divider by selecting it and invoking OPEN again, displaying the icons within it. Only divider icons can be opened in a directory window; other icons must first be copied out to the Desktop.

Close/Close All

A user closes a directory window by invoking the Close menu command, returning the directory to its iconic form. If a divider is open within the window, Close returns the divider to its iconic representation in the list of dividers shown when the divider was opened. When a divider is open, the Close All command becomes visible in the window menu, similar to other container windows. Close All causes the directory window to be reduced to its iconic form on the Desktop regardless of how many levels of divider were open.

Move Directory/Directory Contents

There are several restrictions on the Move operation when applied to directories:

> **Move Directory** - A directory icon can be moved from place to place on the Desktop. It cannot be put down on any other icon.
>
> **Move Icon in Directory** - Except for the Typing Aids divider, icons contained in a directory cannot be moved. All icons in the Typing Aids divider may be moved within the divider or moved to the desktop.
>
> **Move Icon on Desktop** - Once an icon has been copied from a directory to the Desktop, it can be moved from place to place on the Desktop.

Copy Directory/Directory Contents

There are several restrictions on the Copy operation when applied to directories:

> **Copy Directory** - The local directory cannot be copied.
>
> **Copy Icon in Directory** - A user may copy any non-divider icon from a directory to his Desktop. This is the normal way to obtain access to resources in the internetwork.

Copy Icon on Desktop - Once an icon has been copied from a directory to the Desktop, it cannot be copied again, except for data icons obtained from the Basic Documents, Folders and Record Files divider and the Typing Aids divider. The restriction on copying function icons is to avoid confusion as to the state of the icon's internal data and queues. if any.

Delete Directory/Directory Contents

A user cannot delete his local directory. Nor can he delete the contents of any directory window, local or remote, except for the Typing Aids divider. Icons in this divider may be deleted at will. (Only a system administrator can delete function icons from a network, which he does at a sever control terminal.) At all times, a user may delete any icon which he has copied to his Desktop.

Show/Change/Copy Properties of Directory/Directory Contents

Directory and divider icons have no properties. A user can display the properties of an icon which has been copied to the Desktop from a directory by selecting it and pushing the PROPERTIES key. The properties displayed for function icons are information-only, if the icon has any at all; users cannot modify them either directly or by copying properties. The properties for data icons obtained from the directory are in general writable, as described elsewhere in this specification.

Create Directory/Directory Contents

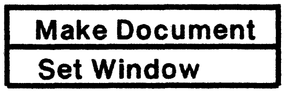
Every Desktop permanently contains a local directory. It cannot be copied or deleted. It is a built-in part of the system. Remote directories can be created by system administrators as with other directory contents, as described below.

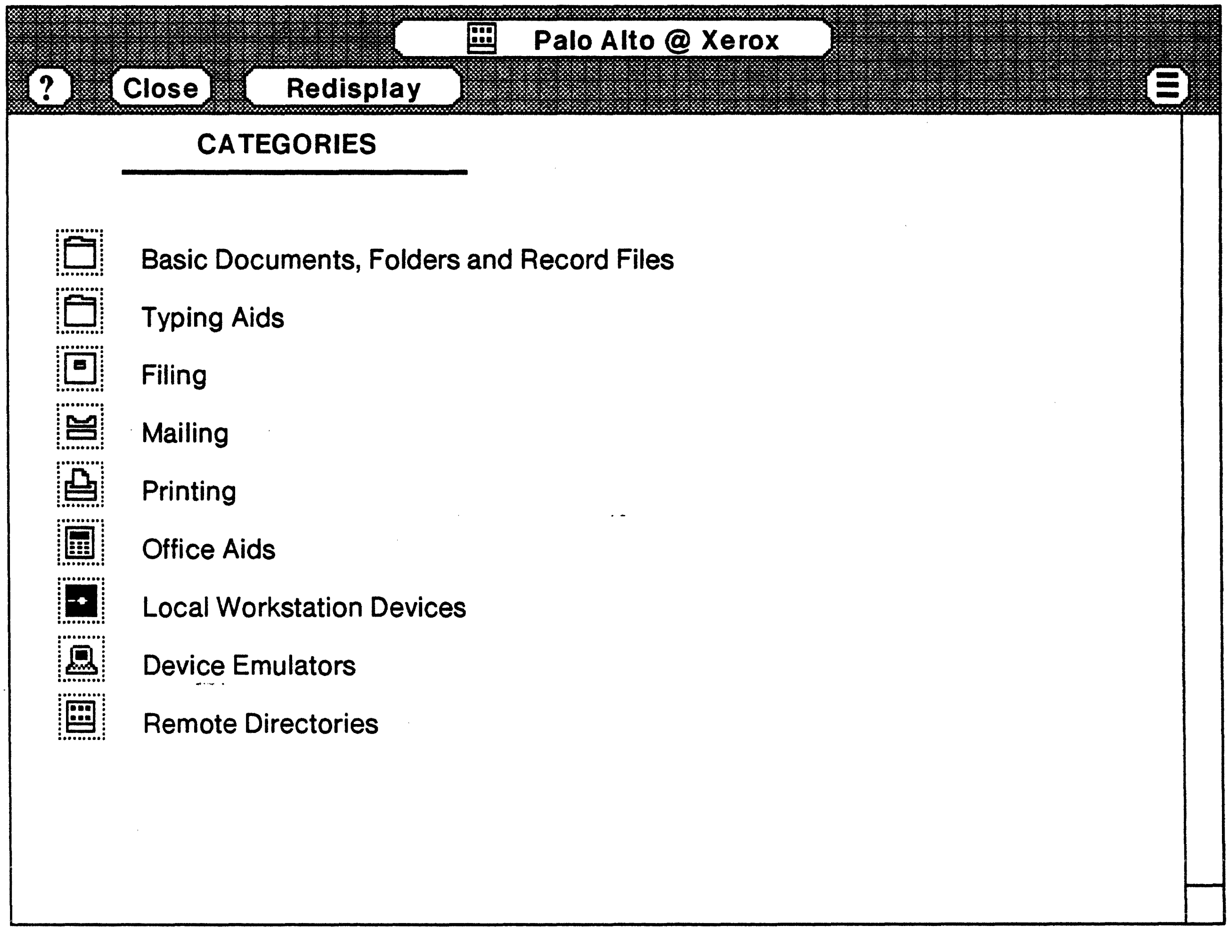
Since a directory merely reflects the information in the domain's Clearinghouse and the File/Mail Services registered therein, icons cannot be inserted into a directory directly, except for data icons placed into the Typing Aids divider. Note that data icons placed into the Typing Aids divider are private to the particular user, unlike function icons. A system administrator registers function icons at the appropriate server control terminal. They automatically appear in the directory the next time it is opened.

Refresh Directory Window

The contents of a directory window can be refreshed (brought up to date with any changes at the Clearinghouse or other services) via the Redisplay menu command. As with all Redisplay commands, the current scroll position is not maintained.

auxiliary menu

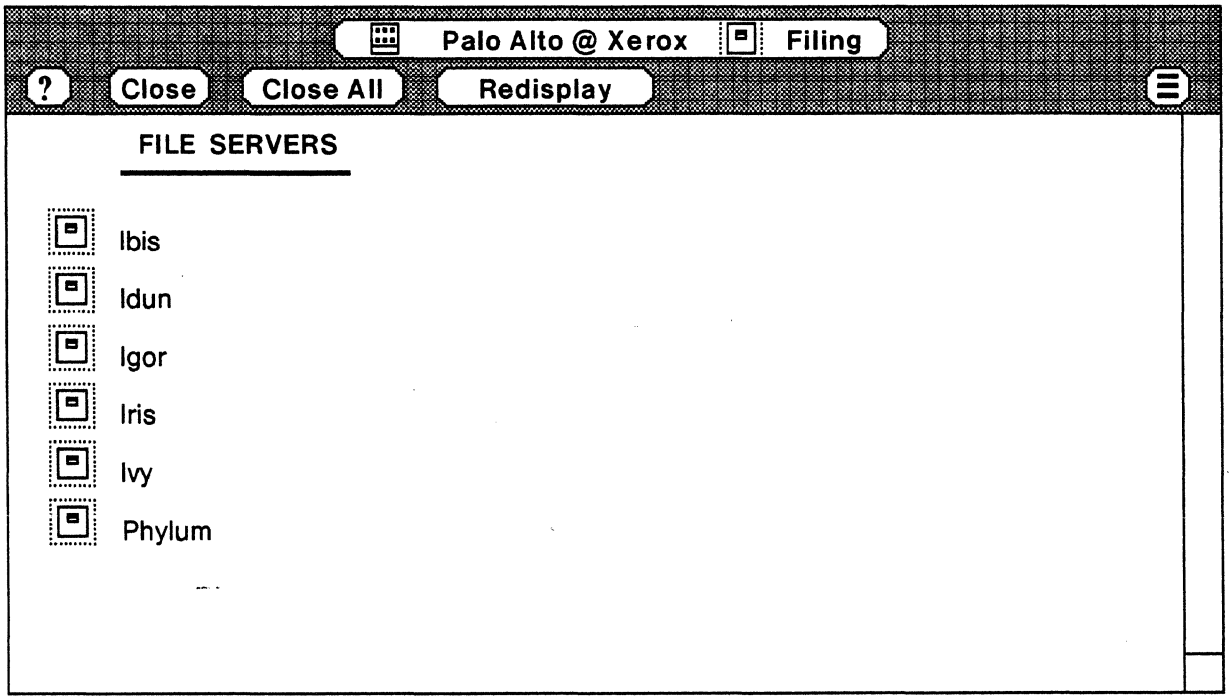




CATEGORIES

- Basic Documents, Folders and Record Files
- Typing Aids
- Filing
- Mailing
- Printing
- Office Aids
- Local Workstation Devices
- Device Emulators
- Remote Directories

Figure DIR-1 -- Directory window for the Palo Alto domain in the Xerox region



FILE SERVERS

Ibis

Idun

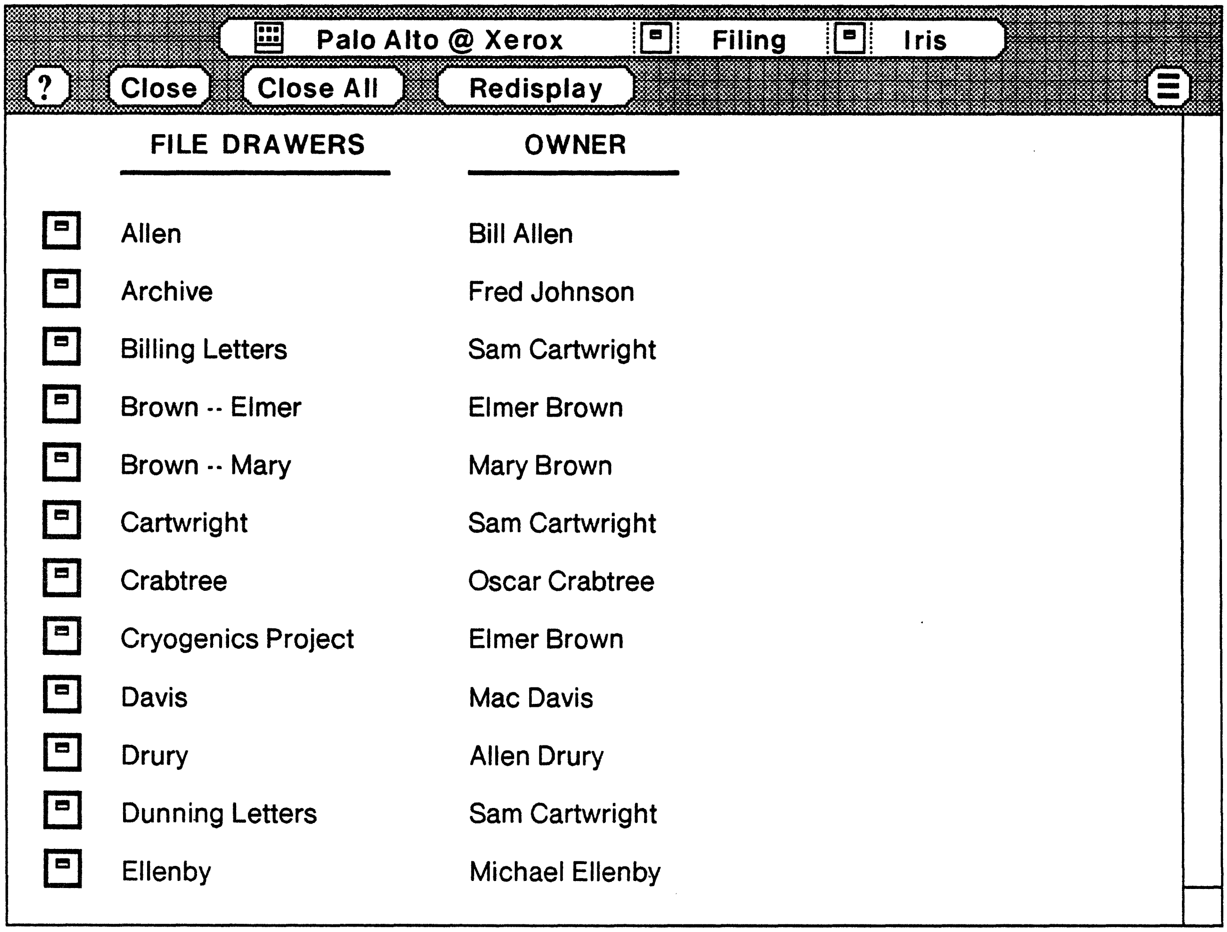
Igor

Iris

Ivy

Phylum

Figure DIR-2 -- Open Filing divider showing six file servers



FILE DRAWERS	OWNER
Allen	Bill Allen
Archive	Fred Johnson
Billing Letters	Sam Cartwright
Brown -- Elmer	Elmer Brown
Brown -- Mary	Mary Brown
Cartwright	Sam Cartwright
Crabtree	Oscar Crabtree
Cryogenics Project	Elmer Brown
Davis	Mac Davis
Drury	Allen Drury
Dunning Letters	Sam Cartwright
Ellenby	Michael Ellenby

Figure DIR-3 -- Open File Server divider for the Iris file server

(PAGES 251 THROUGH 256 INTENTIONALLY BLANK)

(PAGES 251 THROUGH 256 INTENTIONALLY BLANK)

21. ELECTRONIC MAIL

The electronic mail functions allow Star users to transfer documents, folders, record files, non-Star documents, and non-Star record files to and from users at Star workstations, Xerox 860 IPS workstations, and various communicating foreign devices. Conversion to the correct format is handled automatically in most cases.

[This section describes Star-1 electronic mail. In many ways it is constrained by mail services available on the *OIS File Server*. Subsequent versions will include more robust mailing and distribution features.]

USER OVERVIEW

Electronic Mail functions make use of communication hardware, namely the *Ethernet* and *RS232C ports*, an *OIS Mail Server*, and the user registration features described in the *Directories* section. These facilities must be present to support the features described in this section.

All Electronic Mail functions are invoked using the In Basket and Out Basket objects. Documents, folders, record files, non-Star documents, and non-Star record files (hereafter referred to as *mailed* or *mailable objects*) can be moved and copied to and from In/Out Baskets in much the same way as with File Drawers. The major difference is that objects placed in an Out Basket are mailed and then automatically deleted after they have been sent, and objects appear automatically in an In Basket as they are received.

Mailing is controlled by a set of options that appear whenever a mailable object is dropped on an Out Basket. The options specify the names of the recipients. A recipient name may be either a user name or a public distribution list name. Names can also be generated by expanding private distribution lists, using abbreviation expansion.

Mailed objects are delivered to In Baskets by the mail service. Received mail is classified as "new mail" until it has been opened, at which point it becomes "old mail." Normally the contents of an In Basket are accessible only to its owner and are not shared. They may, however. be shared in the same way as File Drawers (see *Document Filing*). In Baskets have the same access restrictions as File Drawers; Out Baskets have no access restrictions.

If a document or record file is sent to a user at a non-Star communicating word processor such as the Xerox 860, it is automatically converted to the correct format. A document or record file is not automatically converted, however, when it is received by a Star user from a non-Star user; the Star user explicitly asks to have it converted if he wishes. This permits Star to store objects in non-Star format and to forward them to other destinations, thus acting as a "switch" or "interface" to a variety of foreign devices.

DETAILED SPECIFICATION

OBJECTS

Mailable Objects

Star documents, folders, record files, reference icons for any of these types, non-Star documents, and non-Star record files can be electronically mailed between Star users. (When a reference icon is mailed, the reference itself, *not* the object referred to, is sent.) In addition documents and record files in both Star and non-Star format can be electronically mailed between Star users and non-Star users, such as Xerox 860 or IBM OS/6 users; facilities are available for converting to and from the various foreign formats.

In Basket

An *In Basket* object provides users with the ability to receive electronic mail in Star. It can exist as an icon in a Directory or on the Desktop, or as a window on the Desktop. It is similar to a File Drawer that contains mailed objects. Each In Basket communicates with exactly one mail server. Whenever a user is logged on, each In Basket on his Desktop periodically "polls" its mail server and retrieves any mail that has arrived since last checking.

An In Basket may be physically stored on a mail server or on a workstation's local disk. The user-perceived behavior is the same in either case, although mail opening may be slightly faster in the local case. The System Administrator declares where each In Basket is to be physically stored when he creates it.

An In Basket can be shared among users or private to a single user, just like File Drawers. It may have *access controls* associated with it; the access controls for an In Basket are the same as for a File Drawer.

[In Star-1.0 all In Baskets will be stored on the Desktop, will be private, and will have no access controls.]

In Basket Icon

A user can have multiple *In Basket icons* on his Desktop if he chooses, designating the same or different mail, just as he may have multiple File Drawer icons designating the same or different files. In Basket icons are copied out of the Directory. Normally a user has a single In Basket for his personal mail. When new (unopened) mail exists, the In Basket icon is displayed with a letter in it.

In Basket Window (see Figures ELM-2 and ELM-3)

An *In Basket window* shows objects received by the In Basket. It is ten inches wide. Each object is represented by a descriptor composed of a small icon followed by fields that describe the object. None of the textual information in the window can be edited, although objects in the window can be opened. The fields are "Name", "From", "Date", and "Recipients".

The Recipients field always contains all the recipients, valid or invalid, to whom the object was addressed; but note that only the *names* of public distribution lists appear in the Recipients field, not their contents. The conventions for the Recipients field are as follows (see Figure ELM-2):

> The "to" recipients are preceded by the string "To:".
>
> The "copies" recipients are preceded by the string "Cc:".
>
> Invalid recipients are preceded by the string "Invalid To:" or "Invalid Cc:" as appropriate. Invalid recipients are always individual users, even if the user was in a distribution list. A distribution list appears only if the distribution list name itself is invalid.
>
> If more than one of the above categories occurs in a Recipients field, they are separated from one another by a semicolon and a blank line.
>
> Parentheses are used to indicate who did and did not actually receive the object, and to enclose descriptive text. The system *did not* deliver the mail to any name that is enclosed in parentheses. The system *did* deliver the mail to any name that is not in parentheses. Besides being a help to the user, this information is used by Send Mail, Answer, and Forward (q.v.).

An In Basket window displays objects in the order in which they were received, oldest at the top, newest at the bottom. The objects are considered to be "new mail" (indicated by the word NEW at the left edge of the descriptor) until they are Opened, whereupon they become "old mail" -- the NEW is removed. Items which are sent to an In Basket by the system because there were errors in their recipient names (see "Send Mail" below) are indicated by the word ERROR instead of NEW. ERROR is equivalent to NEW as far as the In Basket window is concerned. [Implementation note: NEW and ERROR can just be *bitmaps* in the window; they do not require a new font.]

The title of an In Basket window contains a small In Basket icon, followed by the name of the In Basket (corresponds exactly to a user's name). If an object is open in the window, its icon and name are appended to the title. An In Basket window contains scroll and thumb areas as described in *System Overview*. The window may be scrolled in the normal fashion. The window menu commands are ?, Close. New Form, Answer. Forward and Redisplay. Make Document, Set Mail Form, and Set Window appear in the auxiliary menu. If an object is open in the window, four additional commands appear in the menu: Close All, Discard, Show Next and Show Previous.

In Basket Property Sheet

In Baskets have no properties.

Out Basket

An *Out Basket* object provides users with the ability to send electronic mail in Star. It can exist as an icon on the Desktop or in a Directory, or as a window on the Desktop. It is similar to a Printer in that both involve queues of objects waiting to be processed. The objects in an Out Basket are waiting to be sent to its associated mail server, are in the process of being sent, or have failed to send for some reason (e.g. the mail server is down). The mail server is responsible for actually transmitting the objects to their intended recipients.

Out Basket Icon

A user can have multiple *Out Basket icons* on his Desktop if he chooses, although he normally has only one. Out Basket icons are copied out of the Directory. Each Out Basket communicates with exactly one mail server. Each icon has its own separate mail queue. Whenever the mail queue is non-empty, the Out Basket icon is displayed with a letter in it.

Out Basket Window (see Figure ELM-4)

An *Out Basket window* shows objects waiting to be mailed, i.e. its "mail queue". Each object is represented by a descriptor composed of a small icon followed by fields that describe the object. None of the textual information in the window can be edited, nor can any of the objects in the window be opened. The fields are "Name" and "Status". The status is one of "Waiting" or "Being sent".

An Out Basket window displays objects in the order in which they occur in the mail queue. They are sent in this order, top one first.

The title of an Out Basket window contains a small Out Basket icon and the Out Basket name (simply "Out Basket" in Star-1). An Out Basket window contains scroll and thumb areas as described in *System Overview*. The window may be scrolled in the normal fashion; this does not change the mail queue or the object being sent. The window menu commands are ?, Close and Redisplay. Make Document and Set Window appear in the auxiliary menu.

Out Basket Property Sheet

Out Baskets have no properties.

Mail Server

A *mail server* is one of Star's "integrated network services." In Star-1 it is co-resident with an OIS *file server*. It is a machine connected to a (large) disk store and an Ethernet. It provides the processes which resolve recipient names into mail addresses and transmit electronic mail over the network, local or remote, to those addresses. The address translation is accomplished using the "Clearinghouse" (see *Directories*). The collection of facilities supplied by mail servers, the Clearinghouse, and the directory is called the "mail service". See the file server and Clearinghouse documentation listed in Section 2.

Recipient Name

A *recipient name* is a string of characters that identifies a user or a public distribution list to which a mailable object is to be sent. A user name may be the full name or an alias for the user (see *System Overview*). A public distribution list name designates a distribution list on a mail server. User and public distribution list names share the same name space.

A recipient name must include a "domain" and an "organization" if they are different from the sender's (see *System Overview*). An "alias" for a user is a valid recipient name. For example, one could type "Jones" instead of "John Paul Jones" if "Jones" is registered as an alias. Aliases are preserved in the In Basket window; they are not expanded to full names.

[Eventually Star will permit *icons* referring to users and public distribution lists to be placed inside documents (such as Mail Forms) and inside the Mailing option sheet. This will enable users to unambiguously designate recipients without having to type their names.]

User

See *Directories*.

Public Distribution List

A *public distribution list* is a list of recipient names. It can contain the names of users and the names of other public distribution lists. It is stored on a mail server and is created and updated via the mail server operator interface. It allows a shorthand, error-free way of specifying groups of recipients. [This feature is subject to its availability on the OIS File Server.]

Users must be careful to include enough qualification in recipient names in public and private distribution lists. All names in a distribution list should include the domain if the list is to be used in inter-domain communication. All names in a distribution list should include both the domain and the region if the list is to be used in inter-region (inter-company) communication.

Private Distribution List

A *private distribution list* is a list of recipient names local to a workstation. It is stored as an abbreviation which the user can expand when he places it in a Mailing option sheet or in a document (see *Abbreviation Expansion*). For example, the abbreviation "managers" might expand to "Tom Jones, Pete Smith, Joe Hoover". Users can "share" private distribution lists by sending the abbrevation and its expansion to each other in the normal way. Each user would then define the abbreviation on his own workstation.

Mail Form

Every In Basket has a document associated with it to assist creating, answering and forwarding mail. Called a "Mail Form", it is an ordinary Star document that typically contains one or more Mailing Fields. Whenever the New Form, Answer or Forward command is invoked (described below), a copy is made of this document, the copy is opened, and its Mailing Fields are automatically filled in as appropriate. This automates some of the routine aspects of mail sending.

Users are free to modify an In Basket's Mail Form or replace it entirely. This is done with the Set Mail Form command. An example of a Xerox Mail Form is shown in Figure ELM-1.

Mailing Fields

There are four *reserved mailing field names* in Star documents: "Mail To", "Mail From", "Mail Copies" and "Mail Subject". If any fields with these names occur in a document, the In and Out Basket can process them as follows:

> The In Basket will automatically fill in any Mailing Fields that occur in its Mail Form during answering and forwarding mail.
>
> The Out Basket can automatically extract recipient names from the "Mail To" and "Mail Copies" fields when a document is mailed, freeing users from having to type them in. (This is a Mailing option that users can enable or disable. In either case, additional recipient names can be entered directly in the Mailing option sheet.)

Mailing Option Sheet

The *Mailing option sheet* controls to whom mailable objects are sent (see Figure ELM-5). It appears (with the default settings) whenever a mailable object is dropped on an Out Basket icon or Out Basket window. The options are:

> **To** - The names of the "to" recipients of the object are entered in this repeating text parameter (empty by default). The caret is automatically placed in the first entry, ready for typing. Each entry may contain a single name or multiple names separated by commas. The user may step through the parameters, creating new blank ones as necessary, using the NEXT and SKIP keys as described in *System Overview*. The names entered here are preceded by "To:" in the In Basket Recipients column.
>
> **Copies** - The names of the "copies" recipients of the object are entered in this repeating text parameter (empty by default). It behaves exactly like the To parameter. The names entered here are preceded by "Cc:" in the In Basket Recipients column.
>
> **Additional recipients** - This is a state parameter, IN DOCUMENT, which controls whether the Out Basket looks for additional recipient names in the document. The default is off. If it is on and the document contains the mailing fields "Mail To" and/or "Mail Copies", the contents of those fields are added to the list of recipients. If the mailed object is not a document, this parameter is ignored.
>
> **Deliver** - This is a choice parameter with the choices ONLY IF ALL NAMES ARE VALID (the default) and TO ALL VALID NAMES. This controls what happens if the mail service

determines that any recipient name is invalid (not a registered user). In the former case, the mail is not delivered to anyone if even one name is invalid. In the latter case, it is delivered to all the names that are valid regardless of whether or not invalid ones are present. In either case, if any name is invalid, the mail is returned to the sending user so that he can correct the errors. It appears in his In Basket window with all the names who did *not* receive the mail *enclosed in parentheses.* Thus, depending on this parameter, either all the names will be parenthesized or just the invalid ones.

The Out Basket will not attempt to send mail to any name or other text that is enclosed in parentheses. The system uses this feature in displaying invalid recipient names. Users may also employ it to include descriptive text in their recipient lists. The parenthesized text will appear in the Recipients column in the In Basket, but it will not be interpreted as recipient names.

The Tσ and Copies parameters are editable in the normal way until the user invokes Start. Thereafter the recipient names cannot be changed unless the user removes the object from the Out Basket and starts over. (See "Send Mail" below.)

ACTIONS

Send Mail

There is no explicit Send command in Star. Objects are mailed as a result of their being moved or copied to an Out Basket icon or Out Basket window. This causes the Mailing option sheet to be displayed, enabling the user to specify the recipients. Invoking Start confirms the send action and places the object in the mail queue.

Objects are transmitted to the mail server in the order in which they appear in the Out Basket window (mail queue), top one first. Once an object has been transmitted to the mail server successfully, the Out Basket deletes it. All further handling of the object is the responsibility of the mail service. The mail service converts recipient names into user addresses, performs the (inter)network communication necessary to transmit the object to the destination mail server(s), and stores the object until retrieved by an In Basket.

If the mail service determines that a recipient name is invalid (not a registered name), then the object is delivered to the sending user, where it appears in his In Basket with the invalid recipients parenthesized and prefaced by "Invalid To:" or "Invalid Cc:" (see Figure ELM-2). The user may try sending the object again by selecting it in the In Basket window and moving or copying it to an Out Basket. When an object is moved or copied *straight from an In Basket to an Out Basket*, all parenthesized recipient names are automatically entered in the appropriate parameters in the Mailing option sheet, ready for editing and re-sending. If the user had specified Deliver ONLY IF ALL NAMES ARE VALID, then *all* recipients are entered into the option sheet, since the mail was not delivered to anyone. If he had specified Deliver TO ALL VALID NAMES, then only the invalid names are entered in the option sheet. The user can correct the errors and invoke Start. The object will be sent to the new recipients in the normal fashion.

A Star user may send an object to a user on a foreign workstation in the same way he would send it to a Star user. Conversion happens automatically as described below under "Convert to Foreign Format."

The "Created On"/"Created By"/"Last Changed"/"Changed By" properties are not modified when an icon is mailed. They remain what they were before the icon was dropped on the Out Basket.

[In Star-1.0 sending mail will be *synchronous*, i.e. will not run in the background. To account for this, the behavior of the Out Basket is temporarily modified as follows. When a user drops an icon on an Out Basket, Star does *not* return control to the user until it has attempted to deliver the mail. This may take seconds or minutes. The user can press the STOP key to cancel the sending if he chooses.]

Receive Mail

There is no explicit Receive command in Star. Objects are received automatically by In Baskets. Each In Basket on a user's Desktop periodically asks the mail service if anything is available for it. If so, it retrieves the mail into its contents. The interval between queries is TBD but will probably be a small number of minutes.

[In Star-1.0 receiving mail will be *synchronous*, i.e. will not run in the background. To account for this, Star *does* temporarily have a Receive Mail command. It appears in the In Basket menu. Invoking it causes the In Basket to retrieve any waiting mail from the mail server into its contents. However the In Basket will still periodically poll the mail server to see if new mail exists; if so, it will display an envelope in the In Basket icon, as before.]

Read Mail

When an In Basket icon is opened, its window appears showing the mailed objects it contains. The window automatically scrolls to the first "new mail" item, displaying it at the top of the window (or as high as necessary to show the last item -- see Figure ELM-2). The first "new mail" item is automatically selected, ready to be opened. If a document is opened in the window, its contents are displayed and the window becomes, in effect, a Document window. Similarly for folders and record files; non-Star documents cannot be opened. As in other windows, an icon and the name of the object are appended to the title of the window. When an object in an In Basket window is open, two additional commands appear in the window menu: Show Next and Discard. The user can close the currently open object and open the next one by invoking the Show Next command. The Discard command is similar, except that it also deletes the currently open object. These commands permit rapid reading of mail.

Mail, once opened, can be edited in the normal fashion while it resides in an In Basket.

If a piece of mail is in non-Star format, it must be converted to Star format before it can be read. This is described below under "Convert to Star Format."

Create In/Out Basket

A user may create In/Out Basket icons by copying them from the Directory to his Desktop. He may not copy In/Out Basket icons already on his Desktop, because they may have non-empty contents. There may be any number of In and Out Basket icons on a Desktop designating the same or different mail.

Only a System Administrator can add an In Basket to an installation. This is done at the mail server. See the *Network Administration* guide listed in Section 2.

Select

Selecting In and Out Baskets icons is the same as selecting other icons (see *Desktop*). Selecting objects in an In or Out Basket window is the same as selecting objects in a File Drawer (see *Document Filing*).

Open

A user opens an In Basket by selecting it and invoking Open. This allows him to see the mail that he has received. The objects in an In Basket window can also be opened.

Similarly a user opens an Out Basket by selecting it and invoking Open. This allows him to see the mail that is waiting to be sent. Objects cannot be opened in an Out Basket window; they must be moved out first.

Move and Copy

In general, In and Out Baskets obey the same Move and Copy rules as File Drawers (see *Document Filing*). There are, however, several differences:

Moving or Copying to an In Basket - Mailable objects can be moved or copied to an In Basket. They are treated as "old mail" with no "From", "Date" or "Recipients" entry.

Moving or Copying to an Out Basket Icon - Mailable objects can be moved or copied to an Out Basket icon. They are placed at the bottom of the mail queue.

Moving or Copying into an Out Basket Window - Mailable objects can be moved or copied to a specific location between objects in an Out Basket window; they are placed in the mail queue at that position. They cannot be inserted in a closed object (e.g. a folder) in the window.

Moving out of an Out Basket Window - Moving objects out of an Out Basket window causes them to be removed from the mail queue. The object currently being sent, if any, cannot be moved.

Moving within an Out Basket Window - Objects can be moved within an Out Basket window in order to rearrange the mail queue. The object currently being sent, if any, cannot be moved.

Delete

Users can delete objects in an In Basket or Out Basket window in the normal fashion. The object currently being sent, if any, cannot be deleted.

In and Out Basket icons can be deleted only from the Desktop, not from the Directory. Deleting an In Basket is analogous to deleting a File Drawer -- its contents are not deleted, just the icon which accesses those contents. A user may always get another copy of the icon from the Directory. Deleting an Out Basket is analogous to deleting a Printer -- it and all its contents (i.e., its mail queue) are deleted. Therefore attempting to delete a non-empty Out Basket causes a warning message and requires the user to confirm the deletion.

Convert to Foreign Format

A Star document or record file may not be converted to some other format explicitly. Rather conversion is done automatically as part of the mailing service. A Star user sends an object to a user on a foreign workstation in the normal way, namely by putting it in an Out Basket and addressing it appropriately. The mailed object may be in Star or non-Star format, without regard to the receiving user; for example, an OS6 document could be sent to an 860 user.

Convert to Star Format

A user may convert a non-Star document or record file to Star format by selecting it and invoking the Convert command in the Desktop auxiliary menu. The non-Star object is converted to Star format, and the converted object is attached to the cursor, as if it had been copied. The user may put it down anywhere he wishes, and he may deal with it like any other Star object. The original remains where it was and is unchanged. The unconverted and converted objects both have the same name. The two are distinguished by their icon appearance, the unconverted form showing its type, e.g. 860 or OS6 (see *Desktop*).

Note that unlike conversion to foreign format, conversion to Star format does not happen automatically. This permits Star to act as an intermediary, receiving objects from one foreign workstation and transmitting them to another, with a minimum loss of information.

Redisplay Window Contents

The Redisplay command updates the contents of the In/Out Basket window. Since background processes are potentially changing the contents of In and Out Baskets, this command must be executed to restore the correspondence between the window display and its underlying contents. [Subsequent to Star-1 we may synchronize the display with changes.]

New Form

The New Form command is used to create a new message, automating some actions and making it easier to do the rest.

A user invokes the New Form command in an In Basket window. The system makes a copy of In Basket's Mail Form and opens it on the Desktop. The new Mail Form occupies a separate window; it does not obscure the In Basket window. The system then automatically fills in any mailing fields that exist in the Mail Form, as follows:

Mail To - empty

Mail From - the user's own name

Mail Copies - empty

Mail Subject - empty

The system places the caret in the first field in the document, whether or not it is a mailing field, just as it does with other forms (see *Field Fill-In*). The user may edit the Mail Form like any other document. When he is finished, he closes it and moves it to an Out Basket. If the More Recipients IN DOCUMENT option is chosen, the document can be delivered to the "Mail To" and "Mail Copies" recipients automatically.

Answer Mail

The Answer command is used to reply to a message, making sure the reply goes to the appropriate people with the appropriate identification.

A user selects or opens a mailed object in an In Basket window and invokes Answer. The system makes a copy of the In Basket's Mail Form and opens it on the Desktop. The new Mail Form occupies a separate window; it does not obscure the In Basket window. The system then automatically fills in any mailing fields that exist in the Mail Form, as follows:

Mail To - the person who sent the object -- the contents of the From field in the In Basket window

Mail From - the user's own name

Mail Copies - the user's own name, followed by the other recipients of the object -- the contents of the Recipients field in the In Basket window

Mail Subject - the text "In reply to <sender>'s message of <day and time sent>", for example "In reply to John Smith's message of Wednesday, October 4, 1980, 3:14 pm"

Note: The *range* and/or *domain* parts of a recipient name are *displayed* in an In Basket, Out Basket, option sheet, etc., only if they are different from the sending user's. Also the sending user must *enter* the domain and region when typing a recipient name only if they are different from his own. However the system always *stores* and *transmits* the domain and region information with every recipient name; internally it always passes around recipient names in their "fully qualified" form.

Finally, the system names the new Mail Form "Re: <mailed object name>" (unless the mailed object name already begins with "Re: ", in which case a second "Re: " is not added). For example, if a document named "Staff meeting Thursday" is Answered, the reply would have the name "Re: Staff meeting Thursday".

The user may edit the body of the answer in the normal way. When he is finished, he closes the copy of the Mail Form and moves it to an Out Basket. If the Mail Form contains "Mail To" and/or "Mail Copies" fields, the recipients will already be specified.

Forward Mail

The Forward command also automates some of the routine aspects of message sending. It is used to pass on a received object to other recipients.

The user selects or opens a mailed object in an In Basket window and invokes Forward. The system creates a folder containing the following objects: (a) a copy of the In Basket's Mail Form, and (b) a copy of the mailed object being forwarded. As with the Answer command, the system opens the new folder on the Desktop. The folder occupies a separate window; it does not obscure the In Basket window. The system then opens the copy of the Mail Form in the folder window and proceeds to fill in its mailing fields, if any, as follows:

Mail To - the "to" recipients of the original message

Mail From - the sender of the original message

Mail Copies - the "copies" recipients of the original message

Mail Subject - the text "Forwarding <sender>'s message of <day and time sent>", for example "Forwarding John Smith's message of Wednesday, October 4, 1980, 3:14 pm"

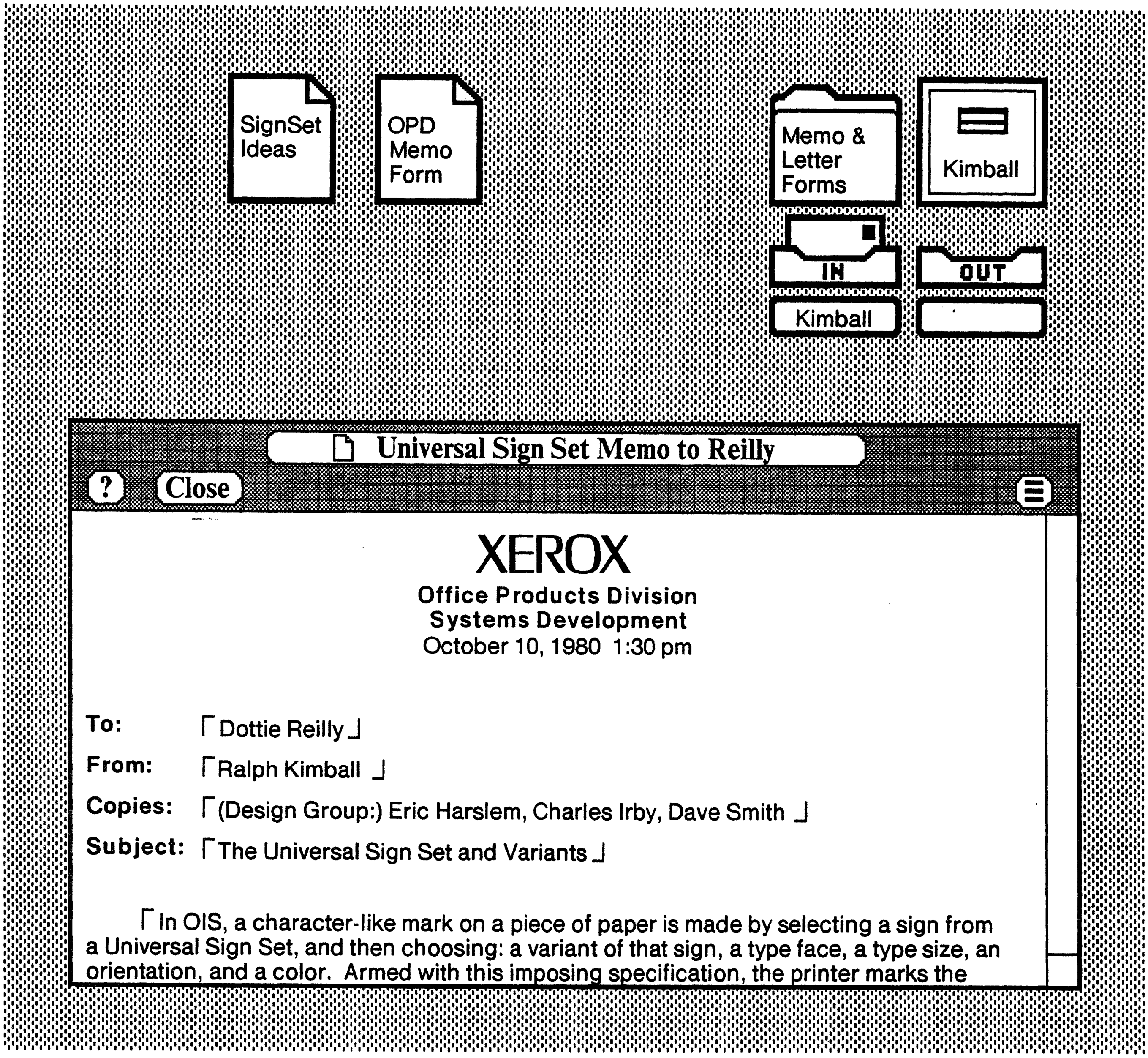
Essentially this copies the information in the In Basket record into the Mail Form in order to preserve a record of the original recipients. Note that the forwarded object is *not* sent to these recipients. Both the new Mail Form and the original mailed object are placed in a folder, and it is the *folder* that is mailed. The recipients of the folder are whatever the user declares them to be when he mails the folder.

Finally, the system names the new folder "Forwarding: <mailed object name>" (unless the mailed object name already begins with "Forwarding: ", in which case a second "Forwarding: " is not added). For example, if a document named "Staff meeting Thursday" is Forwarded, the folder would have the name "Forwarding: Staff meeting Thursday".

The user may add any comments he wishes to the body of the Mail Form in the normal way. When he is finished, he closes the Mail Form and folder and moves the folder to an Out Basket. He specifies the recipients when the Mailing option sheet appears.

Change Mail Form

A copy of the In Basket's existing Mail Form can be made by invoking its New Form command. The copy can then be edited in the normal way, printed, mailed, filed, etc. It can be associated with an In Basket by selecting it and invoking the In Basket's Set Mail Form command. Any document (containing the appropriately named fields) can be used as a Mail Form.



XEROX

Office Products Division
Systems Development
October 10, 1980 1:30 pm

To: ⌜Dottie Reilly⌟

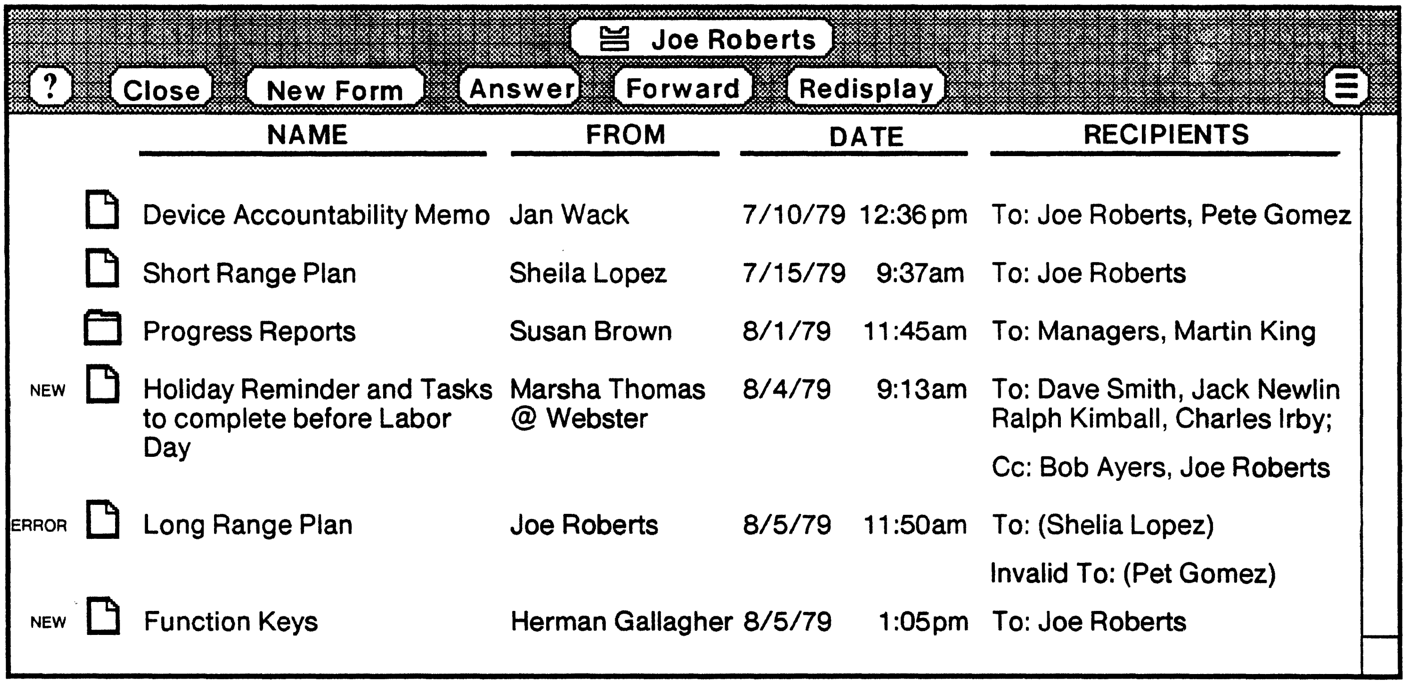
From: ⌜Ralph Kimball ⌟

Copies: ⌜(Design Group:) Eric Harslem, Charles Irby, Dave Smith ⌟

Subject: ⌜The Universal Sign Set and Variants⌟

⌜In OIS, a character-like mark on a piece of paper is made by selecting a sign from a Universal Sign Set, and then choosing: a variant of that sign, a type face, a type size, an orientation, and a color. Armed with this imposing specification, the printer marks the

Figure ELM-1 -- Desktop with Mail Form and In/Out Baskets



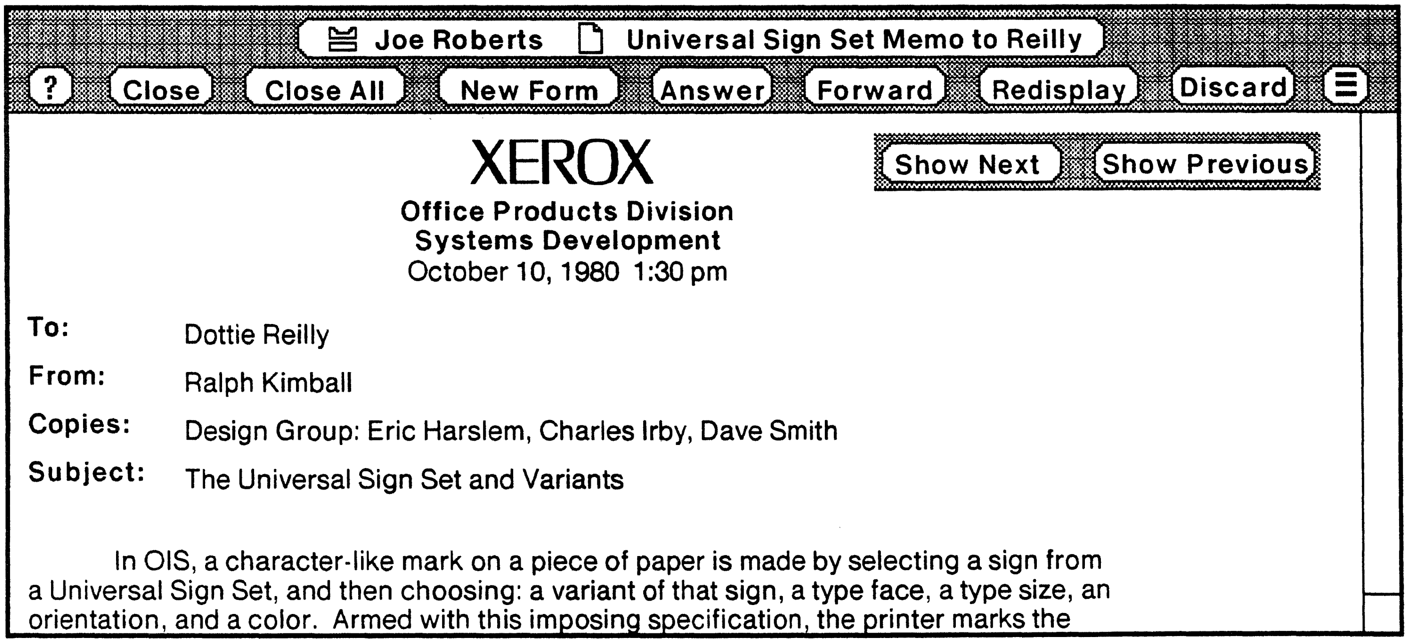
		NAME	FROM	DATE	RECIPIENTS
	□	Device Accountability Memo	Jan Wack	7/10/79 12:36pm	To: Joe Roberts, Pete Gomez
	□	Short Range Plan	Sheila Lopez	7/15/79 9:37am	To: Joe Roberts
	▭	Progress Reports	Susan Brown	8/1/79 11:45am	To: Managers, Martin King
NEW	□	Holiday Reminder and Tasks to complete before Labor Day	Marsha Thomas @ Webster	8/4/79 9:13am	To: Dave Smith, Jack Newlin Ralph Kimball, Charles Irby; Cc: Bob Ayers, Joe Roberts
ERROR	□	Long Range Plan	Joe Roberts	8/5/79 11:50am	To: (Shelia Lopez) Invalid To: (Pet Gomez)
NEW	□	Function Keys	Herman Gallagher	8/5/79 1:05pm	To: Joe Roberts

Figure ELM-2 -- In Basket window

The normal width of this window will be ten inches.

In Basket transient menu

Make Document
Set Mail Form
Set Window



XEROX

Office Products Division
Systems Development
October 10, 1980 1:30 pm

To: Dottie Reilly

From: Ralph Kimball

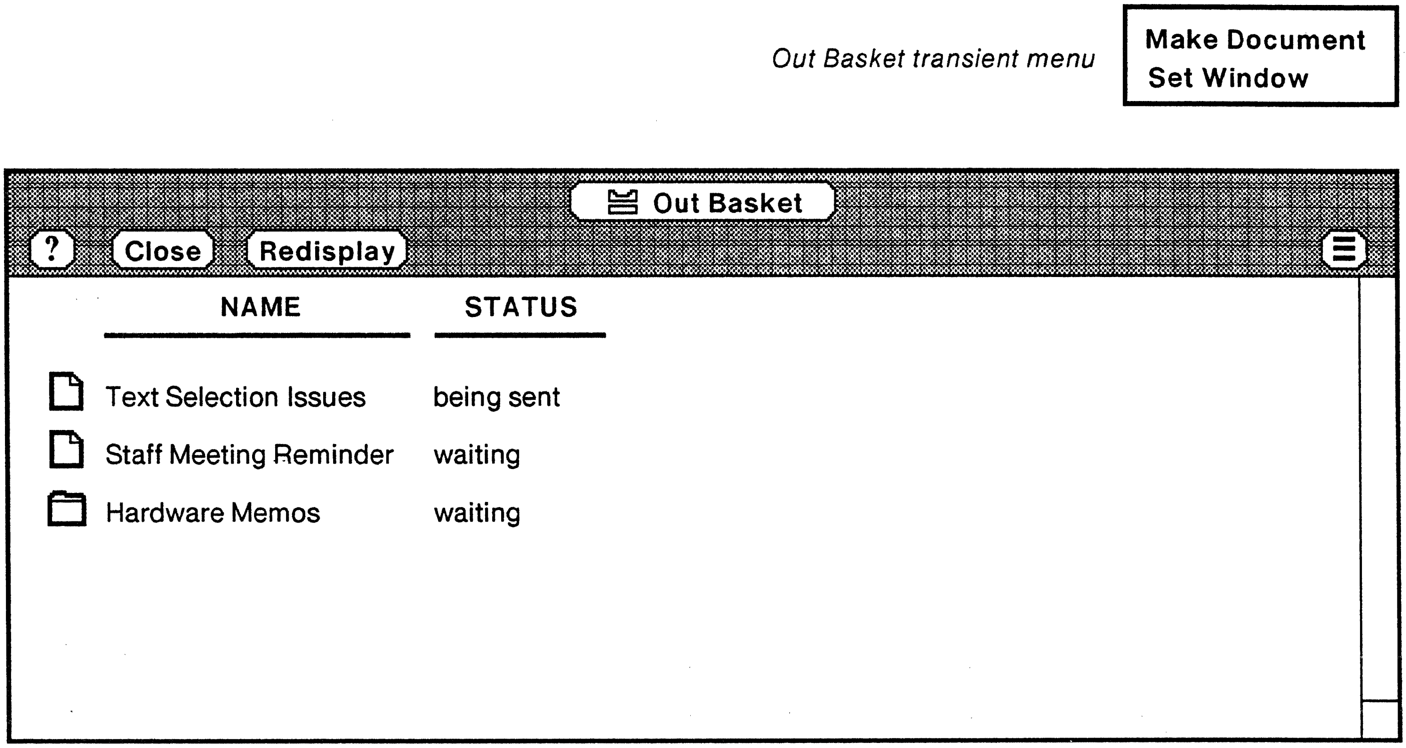
Copies: Design Group: Eric Harslem, Charles Irby, Dave Smith

Subject: The Universal Sign Set and Variants

In OIS, a character-like mark on a piece of paper is made by selecting a sign from a Universal Sign Set, and then choosing: a variant of that sign, a type face, a type size, an orientation, and a color. Armed with this imposing specification, the printer marks the

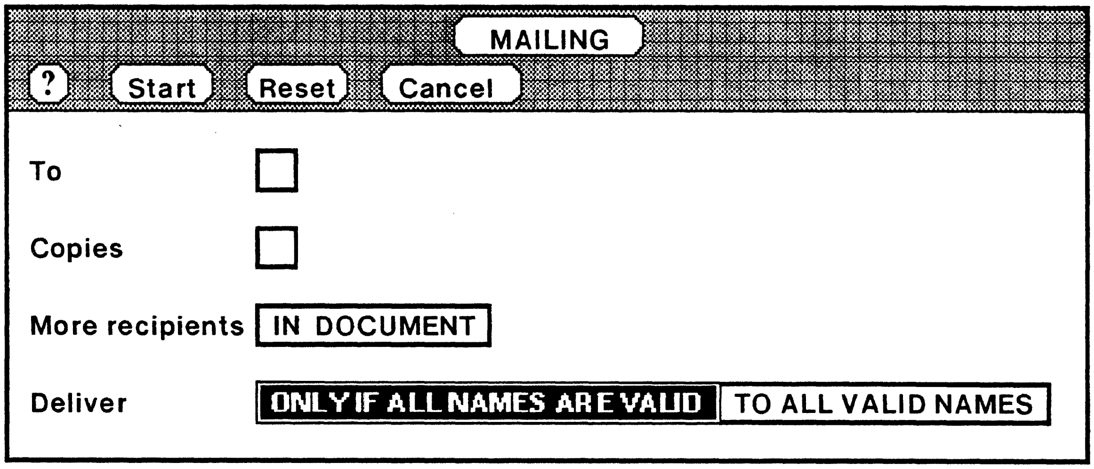
Figure ELM-3 -- Document open in an In Basket window

Note: There will be room in the menu of the 10" wide window for Show Next and Show Previous.


Out Basket transient menu

NAME	STATUS
Text Selection Issues	being sent
Staff Meeting Reminder	waiting
Hardware Memos	waiting

Figure ELM-4 -- Out Basket window



To □

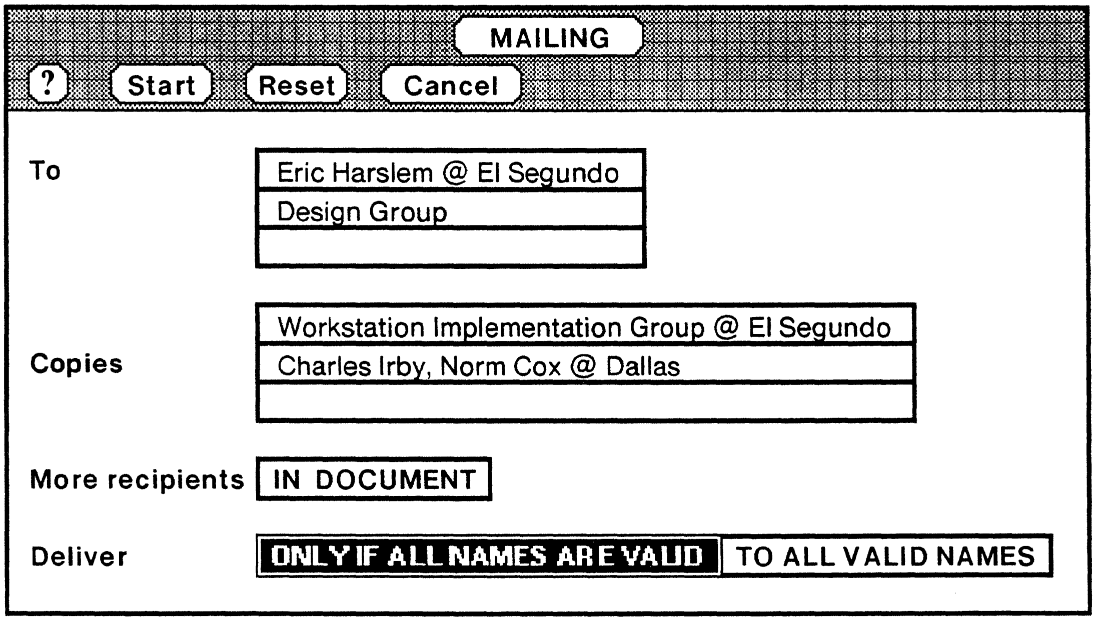
Copies □

More recipients: IN DOCUMENT

Deliver: [✓ ONLY IF ALL NAMES ARE VALID] TO ALL VALID NAMES

Figure ELM-5 -- Mailing option sheet

This appears whenever a mailable object is dropped on an Out Basket. These are the mailing defaults.



To: Eric Harslem @ El Segundo
Design Group

Copies: Workstation Implementation Group @ El Segundo
Charles Irby, Norm Cox @ Dallas

More recipients: IN DOCUMENT

Deliver: [✓ ONLY IF ALL NAMES ARE VALID] TO ALL VALID NAMES

Figure ELM-6 -- Sample Mailing option sheet

(THIS PAGE INTENTIONALLY BLANK)

22. PRINTING

The printing functions allow Star users to render documents, folders and record files in hardcopy form. Printing can occur at local printers on the user's network or at remote printers on other networks.

USER OVERVIEW

Printing functions make use of communications hardware, namely the *Ethernet* and an *OIS Print Server*. In general this hardware must be present to support the features described in this section, although printing on the character printer attached to a workstation can be done even in the absence of an Ethernet.

Two types of printers are available: the impact character printer and the raster output scan (ROS) printer; see *Hardware Overview*. The character printer is primarily limited to text, including multiple-font and tabular text, although limited graphics may be provided. The raster printer can print anything that can be put in a document, including text, graphics and equations; furthermore if the document has just been paginated, the displayed and printed pages will correspond closely. In Star-1 only a relatively low speed ROS is available, called the Low Speed Electronic Printer (LSEP). (Higher speed ROS printers will be available in later releases.)

All printer functions are invoked using Printer icons. Documents, folders and record files (hereafter referred to as *printed* or *printable objects*) can be moved and copied to and from Printers in much the same way as with File Drawers. The major difference is that objects placed in a Printer are printed and then automatically moved back to where they came from.

Printing is controlled by a set of options that appear whenever a printable object is dropped on a Printer. The options include the number of copies, which pages to print, whether to repaginate before printing, etc.

Printers may be used by anyone; there are no access restrictions.

DETAILED SPECIFICATION

OBJECTS

Printable Objects

Documents, folders and record files can be printed in Star. Reference icons for any of these types can also be printed, in which case the Printer will print the contents of the object referred to. Documents and record files in the format of some other system (e.g. Xerox 860 or IBM OS/6) may not be printed directly; they must first be converted to the equivalent Star object.

Printer

A *Printer* object provides users with access to physical printers in Star. It can exist as an icon in a Directory or on the Desktop, or as a window on the Desktop. It is similar to an Out Basket in that both involve queues of objects waiting to be processed. The objects in a Printer are waiting to be processed, are being formatted into OIS Print File Format, are being transmitted to a print server, or have failed to format or transmit correctly for some reason (e.g. the print server is down). The print server is responsible for actually producing marks on paper.

Printer Icon

A user can have multiple *Printer icons* on his Desktop if he chooses, designating the same or different physical printers. Printer icons are copied out of the Directory. Each icon has its own separate print queue.

The Directory contains a special Printer icon named "LOCAL" that always refers to the character printer attached to the workstation *at the time of printing*, hereafter called the "workstation's-local-character-printer" icon. This icon looks and acts like other Printer icons, with the exception that it is not bound to a physical printer until a print request is actually made. Thus with this icon a user can always print an object on the character printer (if any) attached to the machine he is currently on. *Users cannot print on a character printer attached to another workstation, but they can print on character printers connected to a print server.*

Printer Window (see Figure PRT-1)

A *Printer window* shows objects waiting to be printed, i.e. its "print queue". Each object is represented by a descriptor composed of a small icon followed by fields that describe the object. None of the textual information in the window can be edited, nor can any of the objects in the window be opened. The fields are "Name", (number of) "Copies", and "Status". The status is one of "Waiting", "Being formatted" or "Being sent to printer". There may also be various status error conditions, TBD. For the workstation's-local-character-printer, there are the additional status states of "Printing" and "Interrupted".

A Printer window displays objects in the order in which they occur in the print queue. They are sent to the print server in this order, top one first. A Printer window shows only objects placed in that Printer icon by that user; the print server may contain other objects submitted by other users.

The title of a Printer window contains a small Printer icon and the Printer name. A Printer window contains scroll and thumb areas as described in *System Overview*. The window may be scrolled in the normal fashion; this does not change the print queue or the object being printed. The window menu commands are ?, Close and Redisplay. Make Document and Set Window appear in the auxiliary menu.

Printer Property Sheet

Printers have no properties.

Print Server

A *print server* is one of Star's "integrated network services." It is a machine connected to a physical raster or character printer, a disk store, and an Ethernet. It provides the processes which spool print requests in OIS Print File Format over the network, drive the physical printer, and handle exception conditions. See the print server documentation listed in Section 2.

Printing Option Sheet

The *Printing option sheet* makes various printing options available (see Figures PRT-2 and PRT-3). They direct the printing of individual documents, all objects contained in a folder, and all documents generated from a record file. The Printing option sheet appears (with default settings) whenever a printable object is dropped on a Printer icon or Printer window. The options are:

> **Number of Copies** - This text parameter indicates how many copies are to be printed. The default is one. A cover sheet is printed only once, even if the number of copies is greater than one.
>
> **Pages** - This choice parameter determines whether all pages or a range of pages in a document are to be printed. The choices are ALL (the default) and SPECIFIED. If SPECIFIED is set, two text parameters appear in which the user can enter the numbers of the first and last pages to print (see Figure PRT-3). These contain page numbers as they would appear on the printed page, e.g. "Page 7 of 22"; if the number occurs more than once in a document, the first number encountered is used. A single page can be printed by specifying the same number in both parameters. If SPECIFIED is set and no numbers are entered, an error message is displayed when the user invokes Start.
>
> **Print** - This choice parameter determines whether everything in a document or only the contents of filled-in fields are to be printed. (See *Field Fill-in*.) The choices are EVERYTHING (the default) and FIELD CONTENTS ONLY. FIELD CONTENTS ONLY is used primarily when printing on preprinted forms. When it is set, the contents of *form fields* (fields embedded in graphics) will be printed in the positions that they would normally occupy in the document; however fields embedded in text (e.g. in a form letter) may not print in their correct locations.
>
> **Before Printing:** REPAGINATE - This state parameter, if it is on, indicates that a document is to be repaginated before it is printed. The default is on. Note that if a document is copied to a printer, only the copy is repaginated, not the original. If a document is not repaginated before printing, some of its pages may be longer or shorter than the paper size because of editing changes that have occurred since the

most recent pagination. In that case long pages are broken into run-on pages that may not have the correct page numbers; short pages may have excessive white space.

The following options are present only for character printers.

Page Mode - This choice parameter determines whether a document is printed one page at a time or continuously. The choices are ONE PAGE AT A TIME (the default) and CONTINUOUS. If ONE PAGE AT A TIME is on, an interrupt occurs after each page is finished. The user inserts a new sheet of paper and presses the BREAK button on the printer to resume printing. If CONTINUOUS is on, printing will not be interrupted for page changes.

Font Mode - This state parameter determines the handling of font changes during printing. If STOP FOR FONTS is on (the default), printing will be interrupted at each occurrence of a new font. The user changes the print wheel and presses the BREAK button on the printer to resume printing. If STOP FOR FONTS is off, printing will not be interrupted for font changes.

ACTIONS

Print

There is no explicit Print command in Star. Objects are printed as a result of their being moved or copied to a Printer icon or Printer window. This causes the Printing option sheet to be displayed, enabling the user to specify the printing options that he wants. Invoking Start confirms the print action and places the object in the print queue.

For raster printers and character printers connected to a print server, objects are formatted and sent to the print server in the order in which they appear in the Printer window (print queue), top one first. This may or may not be the order in which they are actually printed, since the print server has its own queuing algorithm. After an object has been sent to the print server successfully, the Printer moves it back to where it came from, available for further user actions. If it was copied to the Printer, and thus does not have a previous location, it is placed on the Desktop. [In Star-1.0 all objects may be put on the Desktop. If there is no room on the Desktop, they will be left in the Printer.] Actual hardcopy may not be produced until some time later, when the print server gets around to it. Once an object has been sent to a print server, the user has no further control at his workstation over its printing. He must enter commands *at the server* if he subsequently wants to alter its printing, e.g. cancel it.

For the workstation's-local-character-printer, objects are also formatted and printed in the order in which they appear in the Printer window. The top object remains in the window until it has been completely printed. The printer control commands (see below) can be used to alter the printing of this object. When the top object has been printed, The Printer moves it back to where it came from as described above, and the next object down (if any) begins printing.

The following is printed for each type of printable object:

> *document:* the contents of the document as controlled by the printing options -- for example, only certain pages may be printed. Line numbers and revision marks, if present, are printed in the margins as specified by their properties (see *Formatting and Layout*).
>
> *folder:* all the contents of the folder recursively (depth first) in the folder's sort order.
>
> *record file:* all the documents generated by the record file according to its current filter, sort order, and formatting document.
>
> *reference icon:* the contents of the object referred to, just as if it were printed directly.

A cover sheet is printed at the end containing information about the print request: user name, printed object name, date and time, number of pages printed, etc. A cover sheet is also printed for each object contained in a folder. For example, suppose the following folder is printed:

```
* folder
    * document
    * document
    * record file
        document
        document
    * document
    * folder
        * document
        * document.
```

A cover sheet would be printed for each of the starred (*) objects.

[In Star-1.0 printing will be *synchronous*, i.e. will not run in the background. To account for this, the behavior of the Printer is temporarily modified as follows. When a user drops an icon on a Printer, Star does *not* return control to the user until the icon is completely formatted and sent to the print server. This will typically take minutes. The user can press the STOP key to cancel the printing if he chooses.]

Create Printer

A user may create a Printer icon by copying it from the Directory to his Desktop. He may not copy Printer icons already on his Desktop, because they may have non-empty queues. There may be any number of Printer icons on a Desktop designating the same or different printers.

Only a System Administrator can add a Printer to an installation. This is done by registering it in the Clearinghouse (see *Directories*).

Select

Selecting Printer icons is the same as selecting other icons (see *Desktop*). Selecting objects in a Printer window is the same as selecting objects in a File Drawer (see *Document Filing*).

Open

A user opens a Printer by selecting it and invoking Open. This allows him to see the objects that are waiting to be printed. Objects cannot be opened in a Printer window; they must be moved out first.

Move and Copy

In general, Printers obey the same Move and Copy rules as File Drawers (see *Document Filing*). There are, however, several differences:

Moving or Copying to a Printer Icon - Printable objects can be moved or copied to a Printer icon. They are placed at the bottom of the print queue.

Moving or Copying into a Printer Window - Printable objects can be moved or copied to a specific location between objects in a Printer window; they are placed in the print queue at that position. They cannot be inserted in a closed object (e.g. a folder) in the window.

Moving or Copying a Reference Icon to a Printer - Reference icons (see *Document Filing*) can be moved or copied to a Printer, or deposited there directly as part of the Make Reference command. The Printer will track down the object referred to, if possible, and print its contents. This may save the user time and disk space, since a reference icon is a small object.

Moving out of a Printer Window - Moving objects out of a Printer window causes them to be removed from the print queue. If the object currently being formatted or transmitted is moved out of the window, the processing of that object is immediately cancelled. The system issues a warning message to that effect.

Moving within a Printer Window - Objects can be moved within a Printer window in order to rearrange the print queue. As above, if the object currently being formatted or transmitted is moved to a lower position in the queue, the processing of that object is immediately cancelled. The system issues a warning message to that effect.

Delete

Users can delete objects in a Printer window in the normal fashion. The deleted objects are removed from the window and the print queue. As above, if the object currently being formatted or transmitted is deleted, the processing of that object is immediately cancelled. The system issues a warning message to that effect.

Printer icons can be deleted only from the Desktop, not from the Directory. Deleting a Printer is analogous to deleting an Out Basket -- it and all its contents (its print queue) are deleted. Therefore attempting to delete a non-empty Printer causes a warning message and requires the user to confirm the deletion. A user may always get another copy of a Printer icon from the Directory as described above.

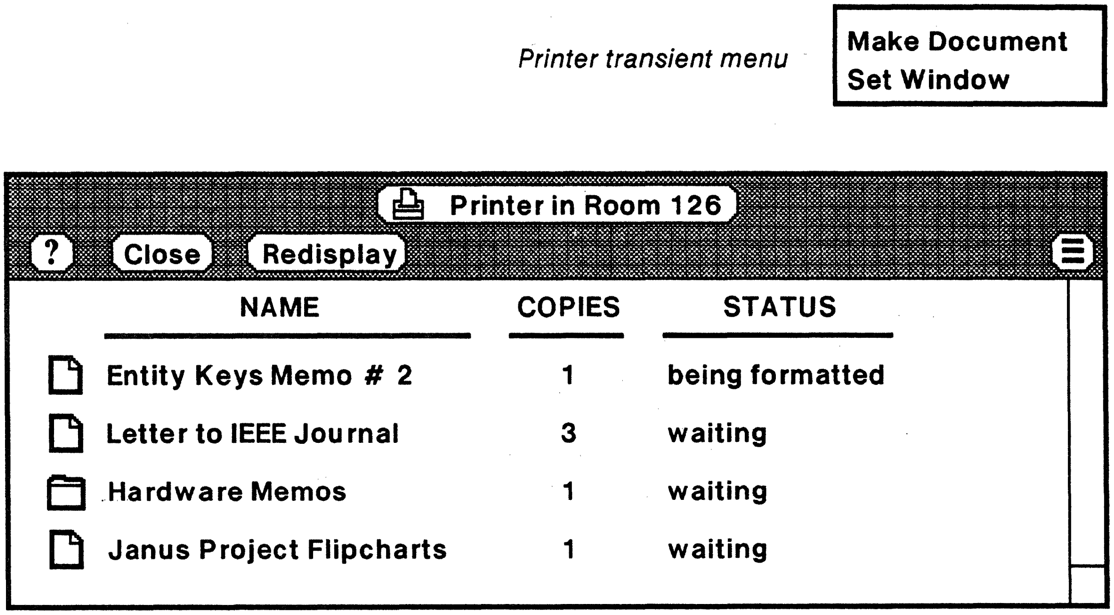
Printer Control Commands

Printer control commands affect the actual printing in progress. *These apply only to the workstation's-local-character-printer.* The commands are Interrupt Printing, Continue Printing, Repeat Page, and Start Over.

> **Interrupt Printing** - temporarily interrupts the printing of an object. The user invokes it by pushing the BREAK button on the printer. Printing stops immediately. The status of the currently printing object changes from "Being printed" to "Interrupted".
>
> Printing on the workstation's-local-character-printer may also be interrupted when a different font is required and STOP FOR FONTS is on, the end of a page is reached and the page mode is ONE PAGE AT A TIME, a STOP code is encountered (see *Text Editing*), or the printer malfunctions in some way. The printer remains in the interrupted state until one of the following printer control commands is invoked, or until the pending print request is deleted from the Printer window. Printing then continues normally.
>
> **Continue Printing** - resumes printing of an interrupted object. The user invokes it also by pushing the BREAK button on the Printer.
>
> **Repeat Page** - reprints the page that was in progress when a printer was interrupted. The user invokes it from the window menu. It appears in the menu only for the workstation's-local-character-printer.
>
> **Start Over** - reprints the interrupted object from the beginning. The user invokes it from the window menu. It appears in the menu only for the workstation's-local-character-printer.

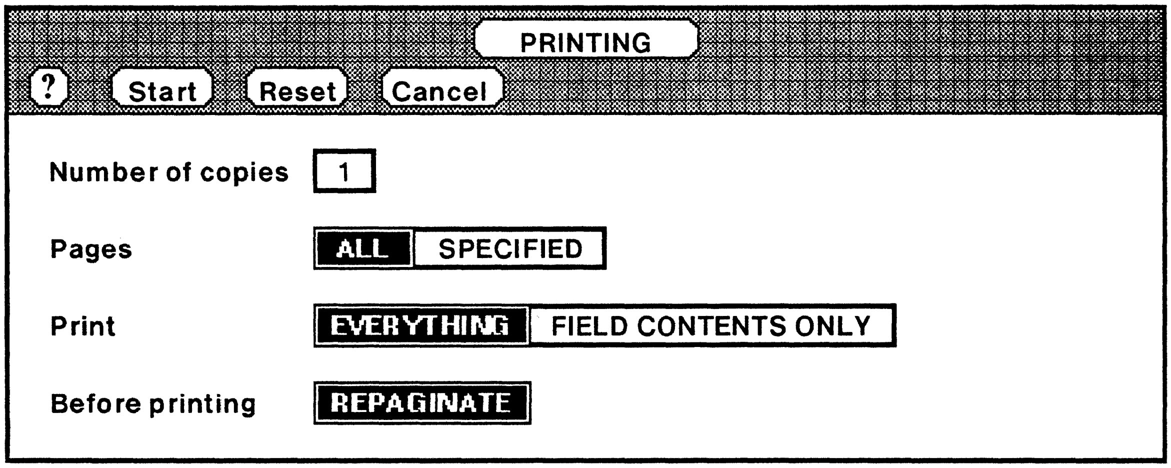
Redisplay Window Contents

The Redisplay command updates the contents of the Printer window. Since background processes are potentially changing the contents of Printers. this command must be executed to restore the correspondence between the window display and its underlying contents. [Subsequent to Star-1 we may synchronize the display with changes.]



NAME	COPIES	STATUS
Entity Keys Memo # 2	1	being formatted
Letter to IEEE Journal	3	waiting
Hardware Memos	1	waiting
Janus Project Flipcharts	1	waiting

Figure PRT-1 -- Printer window



Number of copies 1

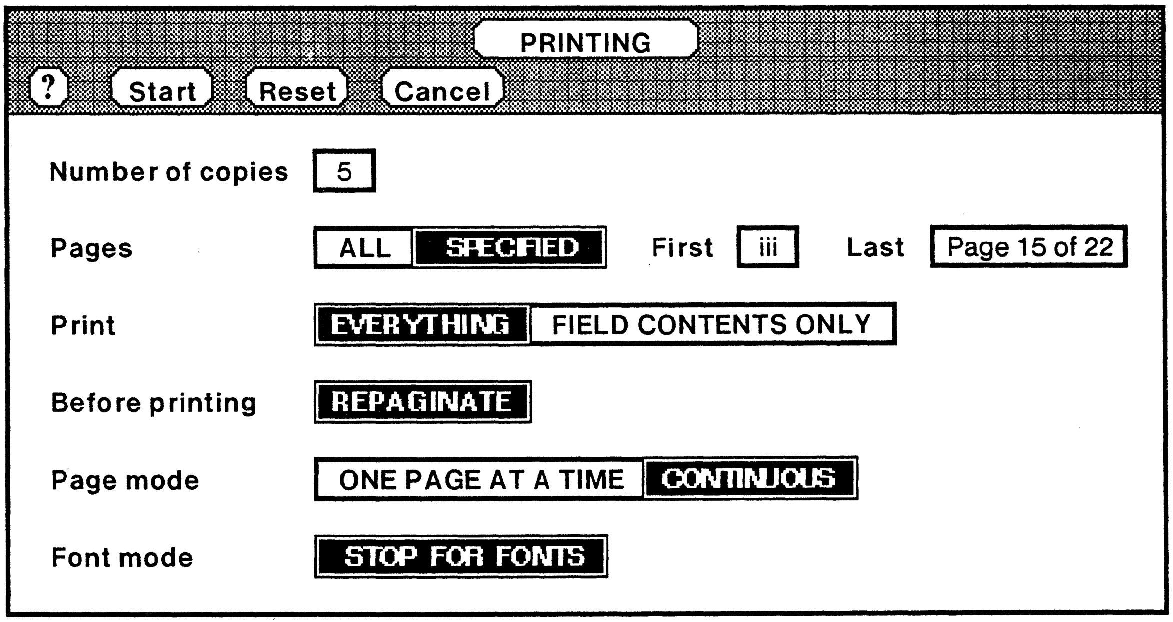
Pages [✓ ALL] SPECIFIED

Print [✓ EVERYTHING] FIELD CONTENTS ONLY

Before printing [✓ REPAGINATE]

Figure PRT-2 -- Printing option sheet

These are the printing defaults.



Number of copies 5

Pages ALL [✓ SPECIFIED] First iii Last Page 15 of 22

Print [✓ EVERYTHING] FIELD CONTENTS ONLY

Before printing [✓ REPAGINATE]

Page mode ONE PAGE AT A TIME [✓ CONTINUOUS]

Font mode [✓ STOP FOR FONTS]

Figure PRT-3 -- Printing option sheet for a character printer

(PAGES 284 THROUGH 290 INTENTIONALLY BLANK)

23. REMOVABLE STORAGE MEDIA

The removable storage media facility provides mechanisms to store, retrieve, and organize documents on floppy disks. The objects and actions used to perform these operations are described in this section. For the most part, removable storage media facilities are similar to the document filing facilities. However, because non-Star documents are being manipulated, there are some necessary departures from the standard way of doing things in Star.

Sections that are of special value in helping the reader understand the material presented in this section are *Desktop* and *Directories*.

USER OVERVIEW

One type of removable storage device is supported in Star-1, the floppy disk drive attached to each Star workstation. The floppy disk drive is accessible only at the workstation to which it is physically connected.

Three floppy disk formats are supported, utilizing the standard SA-850 floppy disk drive:

Star format
Xerox 860 IPS format
IBM Office System 6 format [Not supported in Star Release 1.0]

To access a floppy disk, the icon representing the floppy disk drive must be *opened* by selecting it and pressing the OPEN key. If a disk has been properly inserted, the icon becomes a window and the user may effect information transfer through system commands and commands that are available in the window menu. When access to the device is no longer needed, its window may be *closed*, at which point its representation on the display is reduced to iconic form.

The basic mechanisms for transferring information to and from removable storage media are the *move* and *copy* operations. When transferring documents between Star and a non-Star format floppy disk, format and function codes and text may be converted (optionally in some cases) to/from Star format.

DETAILED SPECIFICATION

OBJECTS

Removable Storage Devices

Removable storage devices supported by Star-1 are limited to floppy disk drives. Each Star workstation is equipped with a single SA-850 drive that is able to read/write standard size, single- or double-sided floppy disks, in single- or double-density formats. Access to a user's local floppy disk drive is by means of the floppy disk object, which can exist on the Desktop in either iconic or window form.

Note that terms specific to other systems (e.g., Office System 6) are introduced in the following descriptions. These terms are important since the user accesses information on foreign media using the same terms as he would if using the foreign systems.

Floppy Disks

Each Star format disk can contain about 300 hardcopy pages of information. A document/record file is not allowed to extend over more than one disk. Only the Star format floppy disk contains information that maintains the Star relationship between folders and documents. Floppy disks may be protected from being written upon by Star by having the write protection hole or notch open. It must be covered with tape to allow writing.

Floppy Disk Icons

The floppy disk icon resembles a floppy disk; however, the icon is essentially a reference to the local floppy disk drive, rather than to individual disk volumes. A floppy disk icon can be used only after it has been copied from the Local Directory to the user's Desktop. (See the *Directories* section.)

Floppy Disk Windows

The floppy disk window provides a listing of the contents of a floppy disk that has been loaded into the local floppy disk drive and OPENed. The window is basically the same as a file folder window, but the exact format and content of the main display area of the window depends upon the type of floppy that has been loaded.

The window title indicates the name and format of the floppy disk whose contents are being displayed. For example, if a Star format floppy has been opened, the window title includes a small (16x16 pixel) Floppy Disk icon, followed by "Star Floppy:," followed by the user-assigned disk identifier. The menu contains the standard commands "?" and Close, plus Redisplay and one floppy-specific command, Initialize Disk. The Make Document and Set Window commands appear in the auxiliary menu. The floppy disk window contains a scroll area, as described in *Desktop*.

The window format and volume content information displayed for the different floppy disk types is described in the following paragraphs.

Star format

The window contains the same type of information as that displayed in a folder window (see *Document Filing*). An example of a Star Floppy Disk Window is provided in Figure RSM-1.

Xerox 860 IPS format

The 860 window is identical to the Star format window, with the following exceptions:

For each document, the window contains an 860 document mini-icon

Document name is limited to 20 characters in length, and subject to 860 naming conventions

IBM Office System 6 format

For each document or segment of a document, the window contains an OS-6 document mini-icon and the following information: a "job" code (A-Z, AA-FF) in alphabetical order, the descriptive title of the information (one to 36 characters), and the size of the document (in hardcopy pages). All job codes may be displayed although some of them may be scrolled out. For job codes that are not in use, there is no document icon and the descriptive title area contains the word UNUSED. Job codes that are UNUSED should be thought of as slots available for documents. They are not objects, nor are they null documents. See Figure RSM-2.

Property Sheets

Floppy disks have no properties.

ACTIONS

Open and Close

A user accesses a floppy disk by selecting the floppy disk icon on the Desktop and invoking Open. If the device is operational and a disk has been inserted properly, a floppy disk window is created. The disk contents are listed in the window, in the format appropriate to the type of disk loaded.

Exception conditions result if:

The floppy disk drive is not operational

A floppy disk has not been inserted in the drive

The floppy disk is not in standard format (e.g., a blank disk to be initialized)

In each case, an empty floppy disk window is created, and an appropriate message is displayed in the Desktop message area.

Close is invoked from the floppy disk window menu. This function reduces the window to iconic form.

Move and Copy

In general, Move and Copy are the actions that perform storage and retrieval operations. They are basically the same as the Move and Copy operations for document filing.

Move/Copy from a floppy disk

When information is to be moved or copied from a floppy disk, the user selects one or more documents (or perhaps folders in the case of a Star format disk) from those displayed in the floppy disk window, and invokes Move/Copy just as in document filing. In the case of foreign format floppies, no conversion is performed by the Move/Copy command.

Move/Copy to a floppy disk

When information is to be moved or copied to a floppy disk, the user makes any valid document or folder selection. The Move or Copy command is invoked. The user then selects the destination for the move/copy to the disk by pointing at the appropriate location in the floppy disk window. If the selection cannot be stored on the disk in its entirety, the user is notified with the message "Disk cannot hold document. <Move/Copy> request terminated."; if more than one document is in the selection, the message "Disk cannot hold all documents. <Move/Copy> request incomplete." The user must load a new disk with more storage space (or make a selection requiring less space on the disk). In the case of non-Star formats, folders will be decomposed into a set of documents. This may result in only some of the documents of the folder being moved or copied to the floppy disk before it is unable to hold another document. In any case, no partial document is written on the disk.

If a Star document is Moved/Copied to a foreign format disk, conversion is performed automatically before the document is recorded on the disk.

For Office System 6 disks, the document is stored using the job code selected by the user. (Only an UNUSED job code can be selected.) The source document name (if any) is used as the descriptive title on the disk. If more than one document is being transferred with a single command, the remaining documents go to the next available (UNUSED) job codes. As in filing, the window arrow moves vertically, but only from Unused Slot to Unused Slot.

If an object document has a name that is too long, the message "Document name <document name> is greater than <20/36> characters. The name has been truncated." The user can modify the name as appropriate.

Delete

The Delete command is used to perform two functions related to floppy disk operations.

First, Delete is used to eliminate entities such as documents, folders and record files that are stored on floppy disk. It is invoked by making a selection from the floppy disk contents that are displayed in the window and pushing the DELETE key. Note that this is a "hard" delete -- Star does not observe the 860 IPS backup index convention, for example.

Second, Delete can be used to delete a floppy disk icon from the Desktop. This deletes the reference only. At any time, the user can create another Desktop reference to the device by making a copy of it from the local directory.

Change Disk

This operation is performed manually by the user. If the floppy disk window is already open, the user invokes Redisplay in the window menu to update the window with the contents of the new disk. If the user invokes a command that would require a disk access before invoking Redisplay, Star will abort that command *but will update the window at that time.* If the aborted command is, by chance, meaningful in the context of the newly displayed disk contents, the command can then be reissued by the user.

Initialize Disk

The Initialize-Disk command appears in the floppy disk window menu, and is used to initialize a blank disk or a disk that has been used previously but contains only information that is no longer useful. Initialization is performed by loading the disk in the drive and invoking the Initialize Disk command (after first opening a floppy disk window). A Floppy Disk Initialization Option Sheet appears (see Figure RSM-3). This option sheet has two parameters, a choice parameter which allows the user to specify the format for which the disk is to be initialized (Star format is the default) and a text parameter which allows the user to specify the disk identifier.

The initialization process begins when the user invokes the Start command in the option sheet menu. If Start is invoked before a disk identifier has been specified, the message "Specify a disk identifier." is displayed and the Start command is ignored. If a disk identifier is specified incorrectly, a message stating that the identifier is in the incorrect format and a description of the correct format will be displayed. The disk identifier for Star is any string of text up to 100 characters in length. A Xerox 860 IPS disk has an identifier of up to 12 characters. For IBM Office System 6, the identifier is one to six characters in length and consists of any combination of capital letters and numbers. The STOP key may be used to interrupt initialization, but will probably leave the disk in an unreadable state.

Reconstruct Disk

TBD

Undo

The Undo command is meaningful for all move/copy operations to or from floppy disks. Delete operations from floppy disks are undoable. [Note that Undo is not available in Star Release 1.0.]

Exceptional Conditions

If a floppy disk is damaged, Star will ask the user if he wants the system to attempt to reconstruct the disk index as described under *Reconstruct Disk* in this section.

If the user tries to perform any operation requiring writing upon or erasing of a disk that is write protected, Star will abort the operation and display the following message: "This disk is protected from <writing/deleting>. Please cover the protection notch if this operation is intended."

Conversion

A facility for converting information from one format to another is provided. This feature is useful whenever the removable storage media has a format different from that of Star. However, if the information in a foreign format is not to be used in Star (e.g., it is to be transferred *as is* to some other workstation), the information can be transmitted without conversion.

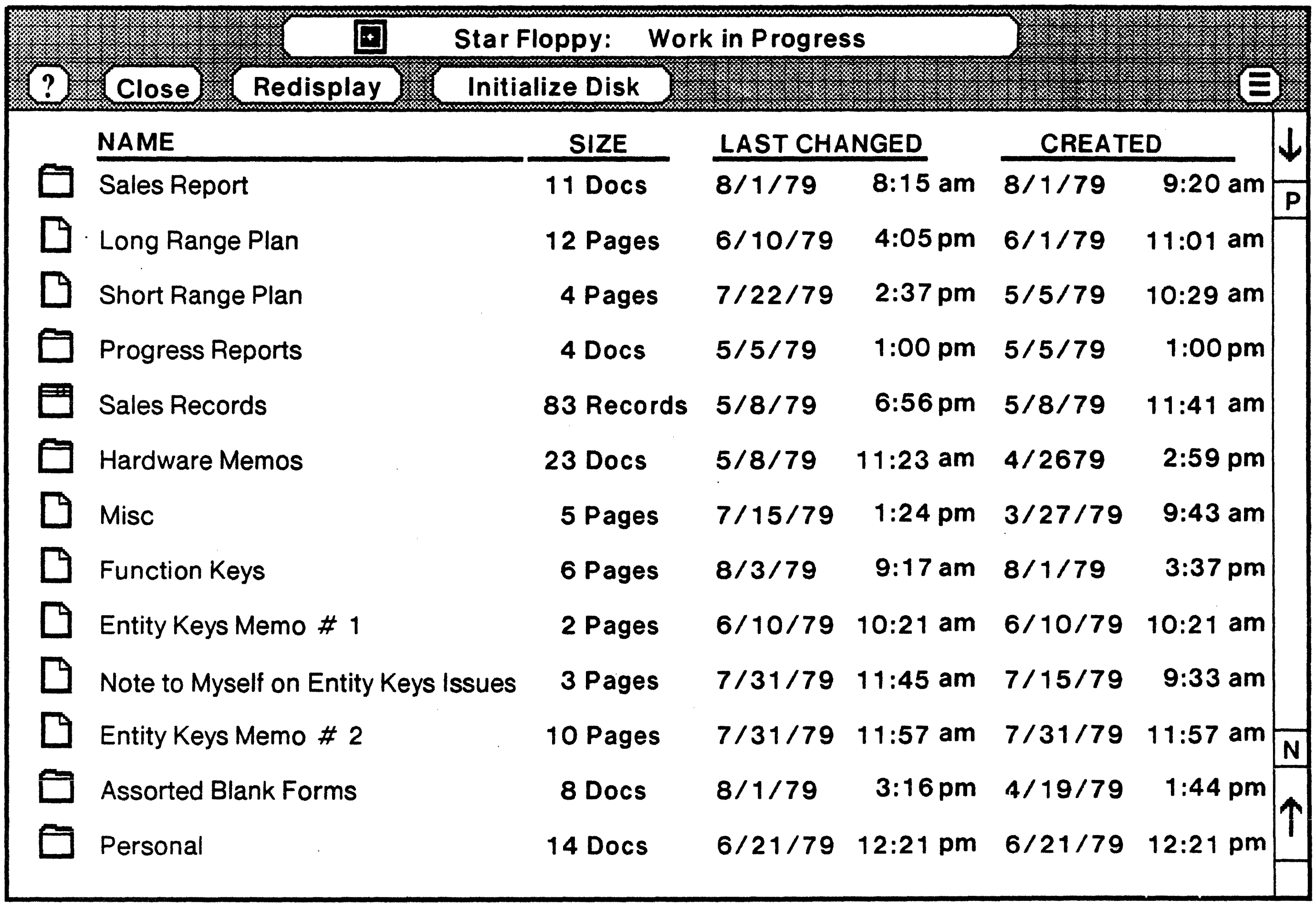
The conversion facility is generally employed for converting between differing document formats. The one exception is that the facility can convert a record file between Office System 6 format and the Star record file format. Because the Star record file format allows more complex record file structures, it is not always possible to convert any Star record file to Office System 6 record file format. [Note that Star Release 1.0 does not support OS-6 media or conversion of record file structures between Star and foreign formats.]

For each removable storage format, the details of conversion are system-defined and are TBD. *Examples* of things that must be considered in designing the conversion routines are tab stops, margin settings, embedded stop codes, required hyphens, required carrier returns, type style changes within documents, underscores, and subscripts. If conversion beyond that which is system-defined is required, the user must edit the document while it is in Star format.

A foreign format document is not converted at the time it is moved/copied into the Star system from a floppy disk. If the user wants the document converted to Star format, he first moves/copies it to the Desktop, then invokes the Convert command in the Desktop auxiliary menu (as described in *Electronic Mail*). This results in the document being converted in place (on the Desktop), indicated by the foreign document icon changing to a normal Star document icon.

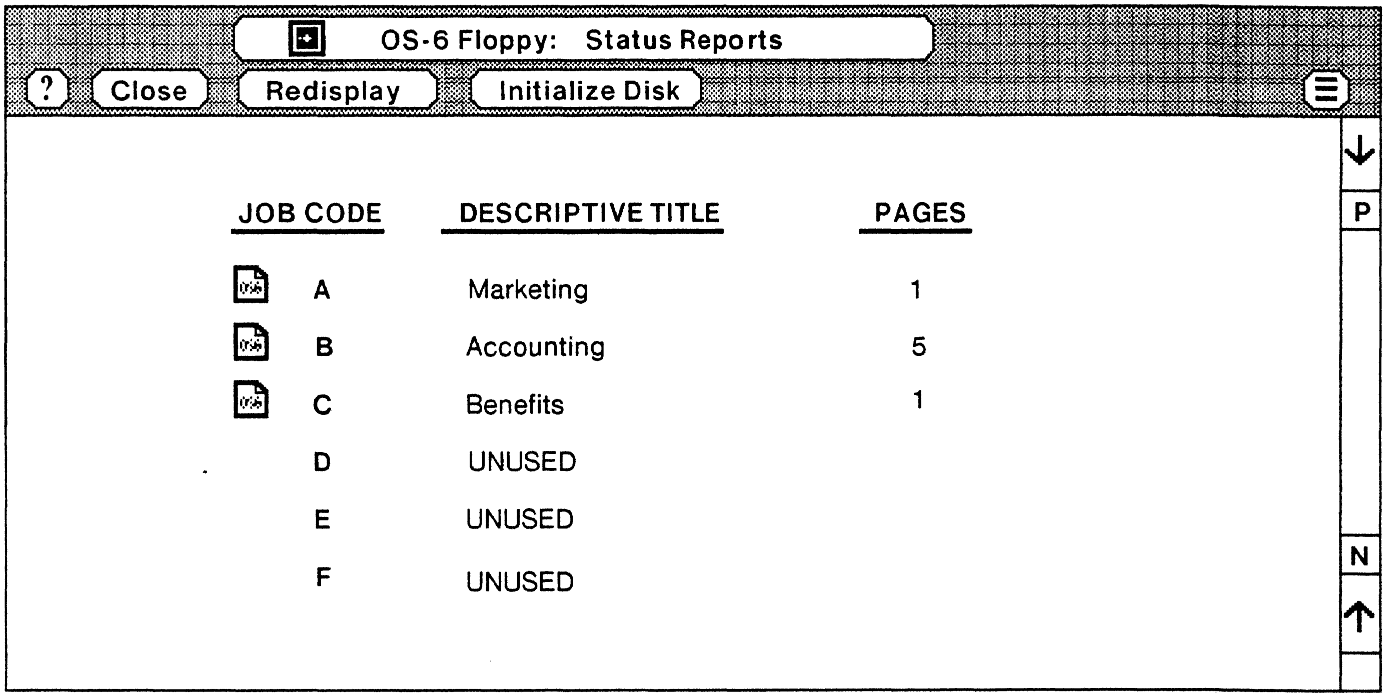
When a Star document is moved/copied to a floppy disk, the document is converted to the appropriate format automatically. These conversion properties are system defined for each supported foreign format.

If both the source and destination are in non-Star format, the document will be converted automatically from one format to the other via an intermediate Star format.



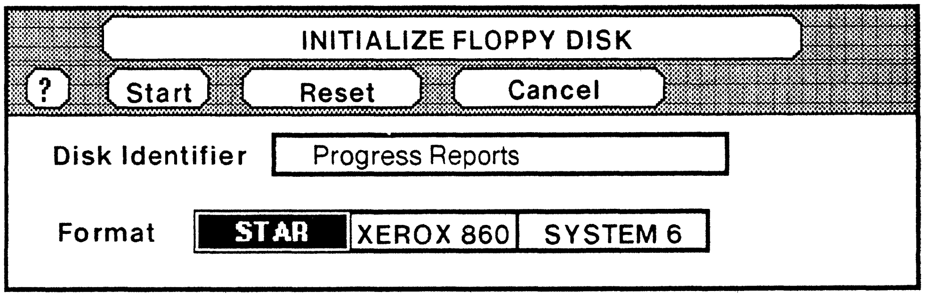
NAME	SIZE	LAST CHANGED		CREATED	
Sales Report	11 Docs	8/1/79	8:15 am	8/1/79	9:20 am
Long Range Plan	12 Pages	6/10/79	4:05 pm	6/1/79	11:01 am
Short Range Plan	4 Pages	7/22/79	2:37 pm	5/5/79	10:29 am
Progress Reports	4 Docs	5/5/79	1:00 pm	5/5/79	1:00 pm
Sales Records	83 Records	5/8/79	6:56 pm	5/8/79	11:41 am
Hardware Memos	23 Docs	5/8/79	11:23 am	4/2679	2:59 pm
Misc	5 Pages	7/15/79	1:24 pm	3/27/79	9:43 am
Function Keys	6 Pages	8/3/79	9:17 am	8/1/79	3:37 pm
Entity Keys Memo # 1	2 Pages	6/10/79	10:21 am	6/10/79	10:21 am
Note to Myself on Entity Keys Issues	3 Pages	7/31/79	11:45 am	7/15/79	9:33 am
Entity Keys Memo # 2	10 Pages	7/31/79	11:57 am	7/31/79	11:57 am
Assorted Blank Forms	8 Docs	8/1/79	3:16 pm	4/19/79	1:44 pm
Personal	14 Docs	6/21/79	12:21 pm	6/21/79	12:21 pm

Figure RSM-1 -- Floppy Disk Drive Window (Star Format)



JOB CODE	DESCRIPTIVE TITLE	PAGES
A	Marketing	1
B	Accounting	5
C	Benefits	1
D	UNUSED	
E	UNUSED	
F	UNUSED	

Figure RSM-2 -- Floppy Disk Drive Window (Office System 6 Format)



Disk Identifier: Progress Reports

Format: [✓ STAR] XEROX 860 SYSTEM 6

Figure RSM-3 -- Floppy Disk Initialization Option Sheet

(PAGES 300 THROUGH 308 INTENTIONALLY BLANK)

24. TERMINAL EMULATION

Star includes facilities which enable a Star workstation to emulate an interactive terminal, and to communicate in that terminal's native protocol. A Star user may employ these facilities to establish communications with a host computer, and interact with data bases and application programs resident on that computer.

Each of the terminal emulations available in Star is described in a separate subsection, within this section.

[*Non-Star Systems*, which previously occupied this section, has been subsumed by Section 21, *Electronic Mail,* and by the Gateway Services, described in separate specifications. See *Derivation and References* for applicable documents.]

(PAGES 310 THROUGH 322 ARE INTENTIONALLY BLANK)

25. RECORDS PROCESSING

The Star records processing facility is designed to store, retrieve and coordinate changes to structured information. The basic unit of information is the *record*. Whereas the document preparation features of Star are oriented towards producing information to be printed, distributed and filed for occasional future reference, the records processing features are oriented towards maintaining information for on-going processing.

USER OVERVIEW

A record file is both a collection of views and a collection of uniformly structured records. The notion of *view* is unique to the records processing facility, and represents the marriage of forms processing with records processing. In Star, a view is a form together with a sort order and a filter, applied to the data in the record file. Through the view's form, the view may look like a table of data, a sequence of filled-in form letters, or a conventional records processing report. The user can view the same data through a variety of forms, each suited to a particular need. Conversely the user can use the same form to view successively altered subsets of the records data, such as all the delinquent customer records, sorted by customer name, or all new customer records sorted by zip code.

Records underlie views in the record file. Each record is simply a collection of fields, and all the records in a record file have the same structure. Since the records of a record file can always be mapped directly onto an appropriately defined *table*, Star simplifies basic record manipulations by presenting them in terms of table row and column operations. In fact, the definition of a new record file's structure will commonly be done by copying the structure of a table in a document. [The initial release of Star may not provide for tables in documents. This document will generally assume the existence of tables, but make explicit note of ramifications of the restriction where applicable. The one relevant point here is that, in the absence of a table facility, a document containing fields may be used to define a record's structure.]

The fields that comprise records are similar to the fields that occur in documents. (See Field Definition and Field Fill-in.) Both have the same legal contents, except that "ANY" fields and computed fields are not allowed in the records themselves. The information in record fields may be both stored and retrieved through the fields in the documents associated with each view. Any Star document can be used for input to, or output from, a record file; however, only documents whose fields have some correspondence to the fields in the records are generally useful. Computed fields *are* allowed in the documents that render record files, and they are very useful for manipulating the data for summaries and reports.

The Star-1 records processing facility is simultaneously fairly powerful and fairly limited as database systems go. The marriage of forms and records processing in views, as described above, is a unique and powerful Star feature. However, the Star-1 facility lacks certain conventional database capabilities, such as being able to define a *key* in a record file (i.e. a field that is guaranteed to contain unique values), or being able to combine several record files together for single logical operations (i.e. "linking" or "joining"). Finally, Star-1 record files can not be simultaneously shared by more than one user.

DETAILED SPECIFICATION

OBJECTS

The relationships among the objects in records processing can best be understood by referring to Figure RP-1, which is a map of the overall facility. This should be kept in mind when studying the following definitions.

Record File

A record file is one of the three basic Star data icons. Like documents and folders, record files can exist on the Desktop, be stored in folders and file drawers, be mailed, printed, moved, copied, and deleted. A record file is both a collection of *views* and a collection of uniformly structured *records*.

View

A view consists of a form, a sort order, and a filter. (These are discussed in detail below in the sections "Record File Form," "Filter," and "Sort Order" under "Stable View Properties.") Views exist within record files and provide the fundamental organization for all the various modes of examining ("viewing") record file data. A view is the medium through which the data in the record file is accessed.

All views associated with a given record file may be examined and manipulated by opening the record file icon. Views are represented in the record file window by small, distinctive icons. The view icons can be moved, copied, and deleted within the window, but cannot leave the window to reside anywhere else. Thus, the record file window resembles a folder of views.

The Base View

A new, empty record file is provided with a *Base View* by the system when its structure is defined (see "Define Structure"). This view consists of a null filter, a null sort order, and the Full Tabular Form. A null filter passes every record in the file; a null sort order means that the records are provided in some arbitrary order. The Full Tabular Form is described in a separate section below.

The Base View may not be moved, deleted, or changed. It may be copied within the record file window; this is the mechanism for creating other views whose properties then *can* be changed.

The Current View

The current view is the last view selected. When a record file is closed, the state of the current view (form, filters, and sort order) is remembered. When the record file is opened again, the same view becomes selected. This optimizes performance for users who habitually return to the same view. They can select the record file icon and invoke OPEN twice to get to their desired view immediately.

Record

A record is a collection of one or more fields. Every record in a record file must have the same structure, i.e. record files are homogeneous, just as are tables. Star exploits the similarity between record files and tables to the fullest: there is a one-to-one correspondence between records and rows in a table, with the individual fields in a record file corresponding to table entries.

Field

A field is the basic unit of information in a record file. Like fields in documents, fields in record files have names, values, and properties, with the exception that record fields may not be computed. [In Star-1 the implementation restricts record fields to be of type TEXT, DATE, or AMOUNT. The "ANY" option, which would allow record fields to contain frames, graphics, and equations, is deferred to a later release.]

A record field may be subdivided, i.e. it may contain subfields. This gives records a hierarchical structure. There is no logical limit to the depth of this nesting. [Lacking Tables in the first release of Star, there is no way to define a nested structure for a record file, so this feature will be delayed until they are provided.]

Record File Icon

A record file is represented by a record file icon (see Figures DTP-2 and RP-1). Like documents and folders, the record file icon may be moved, copied, deleted, filed, mailed, and printed. A record file is something of a cross between a document and a folder, in that it contains textual data (like a document), and contains other documents (like a folder).

Record files in Star-1 cannot be shared simultaneously by more than one user, although like documents, record files within a shared file drawer can be accessed by several different users in turn.

Record File Window

The record file window shows all the views associated with a specific record file. It appears whenever the record file is opened. It is the same size as an In-Basket window (10 inches). Otherwise, it appears like a conventional folder with view icons instead of documents (see Figure RP-2). The record file window is a *prison* for views: they may not be moved or copied out. When it is closed, the record file icon appears.

In addition to the views, a record file window contains two folders, the forms folder and the error folder, described subsequently.

The window menu commands for the record file window are ?, Close, Set Window, and (initially) Define Structure; Define Structure is removed from the menu after it has been invoked.

Filter

A filter is a means of subsetting a record file. When the user sets a filter, he is, in effect, asking for only *certain records* to be displayed. Applying a filter to a record file results in zero or more records "passing." If nothing is specified in the filter, then all the records pass. Setting a filter is the fundamental information retrieval task in Star.

The user sets a filter by entering *field patterns* into appropriate fields in rows of the filter, which is displayed as a table. The field pattern specifies a condition that a corresponding record's field must satisfy if the record is to pass the filter. Each row of the filter represents a set of simultaneous conditions that the various fields of the records must satisfy in order to pass the filter. Subsequent rows of the filter specify alternate sets of conditions that will also allow records to pass the filter. Star always maintains an empty row at the bottom of the table, to allow the user to specify additional filtering. When the user enters a field pattern in this row, Star generates a new empty row below it.

Thus if the filter has one row, in which the first field is constrained to be a specific part name, and the second field is constrained to be a specific salesman name, then only those records matching *both* the part name and the salesman name will pass the filter. But if a second row specifies the same part name but a different salesman name, then this two-row filter will pass records with the specific part name and either salesman name. In logical terms, the elements of a single row specify an "AND" condition, while the distinct rows specify an "OR" condition.

It is possible for the record file (and thus the filter) to contain a repeating divided column with several embedded rows. When a filter pattern is set in one subrow of such a column, records pass this filter if, for *any* row in the filter field, there is a row in the record field for which all the subfields match. In other words, the separate subrows in a repeating divided filter column specify an "OR" condition, just as with normal rows in the full filter.

For example: a record file represents sales staff and their children. The filter has two subrows in the (repeating) Children column, with "Bill" and "Mary" in the Name fields. Then the records which passs the filter are those representing sales staff with either a child named Bill, or a child named Mary, or both. [This paragraph assumes Tables are provided, since divided and repeating fields cannot be created without them.]

Field Pattern

A field pattern is a predicate that describes a range of legal values for a field. It can be used both in filter fields and in the RANGE property of conventional document fields. In a filter a field pattern specifies the range of values in which a record field's value must lie in order for the record to pass the filter. Field patterns for records processing are identical to those described in *Field Definition* except that an additional Special character is provided to represent a completely empty field.

Forms Folder

The forms folder contains all of the "record file forms" used by the various views in the record file (see Figure RP-8). The user accesses the forms folder via the OPEN key. The forms folder exists only within the record file window; it may not be moved, copied, or deleted as an icon. Otherwise, the forms folder behaves like a conventional Star folder (See *Document Filing*). Its contained forms may be moved, copied, and deleted, both inside the window and across its boundaries. New forms may be moved and copied in from the Desktop or other Star containers, and old forms may be moved and copied out, or deleted. Forms may be opened in place and manipulated. One form, the Full Tabular Form, may not be moved, deleted or changed (see "Full Tabular Form").

The window menu commands for the forms folder are the same as for other Star folders.

Record File Form

A record file form is a conventional Star document containing fields. The field names in the form generally match the names of fields in the record file. This name synchronization is the basis for using forms as the medium of record file input and output.

In this section, the word "form" will be used uniformly to indicate "record file form". The distinction is unimportant, since all record file forms are just standard Star documents (except that the Full Tabular form may not be modified or removed from the forms folder).

The Full Tabular Form

At the time a record file's structure is defined, Star creates a form which contains only a table, whose field structure is exactly the same as the record file's. This form is named the Full Tabular Form, and is inserted in the forms folder. It has a protected status: the user may copy it, but not move it out of the folder, delete it, or modify its contents in any way. [The Full Tabular Form will be provided in the first release, even if tables in documents are not. Pending full tables, it will provide ruling lines in the table header, at least.]

Error Folder

The Error Folder is a special folder that is permanently associated with each record file (see Figure RP-9). It is accessed from the record file window by invoking OPEN. Like the forms folder, it cannot be moved or copied from the record file window. It holds folders of source documents [or eventually, source record files] that have been rejected on input to the record file; i.e. one or more of their field values do not obey the constraints specified for a record field. The error records are contained in the documents in which they were entered (see "Insert Records"). All of the rejected record sources resulting from a single user action are placed in a folder with the following name:

<date> <time>: <n> error records out of <total> records

where

date is the date the documents were entered, in the format MM/DD/YY
time is the time the documents were entered, in the format HH:MM <am/pm>
n is the number of records that were rejected
total is the total number of records specified in the transfer.

This folder is then in turn placed in the Error Folder. Therefore, a sequence of successive user actions may cause the creation of several folders of errors within the Error Folder.

The window menu commands for the error folder are the same as for other Star folders.

View Window

The view window is what the user looks at to see record file data (see Figures RP-3 and RP-4). It is accessed by opening a view within the record file window. The contents of the view window consist of zero or more records rendered though the form associated with the view; which records are rendered depends on the settings in the filter. The order in which they are rendered depends on the sort order.

In rendering records through a form, there are two cases. If the form contains a table with one or more columns whose names are the same as top level field names in the record file, and no field outside the table in the form has the same name as a top level field in the record file, then the table is laded from the record file: a row is generated for each record provided from the record file. Otherwise, each record that passes the filter generates a separate copy of the document, which is filled with data from that record.

Because of the great flexibility afforded by the choice of form in the view, what the user sees in the view window can vary greatly. In the most straightforward case, the form is the full tabular form, showing all the possible record fields in a simple table format. (See Figure RP-3.) There would be one row in the table for each record that passed the filter. If there were many fields in the records, the user might have to scroll the window horizontally to see them all. If many records passed the filter, the user might have to scroll vertically as well.

The user could also view the records through a modified version of the full tabular form (see "Create Form"). Some of the columns could be elided (for easier readability on the display) and computed fields could be added. This view would still retain the basic table-like appearance and would appear to be a single (perhaps long) page of a document. [These documents require the Tables facility, so they may not be available in the first release.]

If the form were a standard form letter, and ten records passed the filter, then in the view window the user would see a sequence of ten copies of the form (displayed one-at-a-time), each filled with data from one record. (See Figure RP-4.) This form could also contain computed fields.

Field matching

A field in a form is filled with data from a field in the record file on output, or provides data to that field on input, if they have *matching names*. In the case of a tabular form, the form's table and the record file are considered to have matching names. For a non-tabular form, the single corresponding record and the document are considered to have matching names. Then those fields which are directly contained in the form and the record, and which have identical names, match. Similarly, fields embedded within other fields match if they have identical names, and so do all of their containing fields.

Since field names are required to be unique within a document or record, a field in a form can match at most one field in a record, and vice versa.

Order of field fill-in

For data input into a record file from a form, record fields are filled in from left to right, and top to bottom within REPEATING subfields, for each record. Fields which have no corresponding source are left empty.

For output to a form, fields are filled in the form's fill-in order (see *Field Fill-in*); any computed fields encountered are computed in order, and fields which have no matching source field have their contents left unchanged.

Field Constraints

It is not necessary that the field constraints in a form be identical to the record file constraints. Users may wish to make document constraints more strict in order to provide additional checking in particular cases (e.g. a personnel update form that does not allow salary changes above $20,000) or less strict in order to make document definition easier (e.g. a salary field used for display may need no constraints, even though the field in the file has a maximum value).

When a field is displayed through a document, no validity checking is performed. In the case of AMOUNT and DATE fields, the FORMAT specified for the document field is used to render the field if the field types match and the value is not too large for the format. Otherwise, a default format appropriate to the source field is used. For DATE fields, the default is the "99/99/99" format; for AMOUNT fields, the default is "-...ZZ.99..." (zero-suppressed, with no commas or currency signs). For example:

Record file value and type	Document format and type	Displayed string
"Smith" (TEXT)	"XXX9" (TEXT)	"Smith"
"Smith" (TEXT)	"$Z9.9" (AMOUNT)	"Smith"
"105.66" (AMOUNT)	"XXX9" (TEXT)	"105.66"
"105.66" (AMOUNT)	"$Z9.9" (AMOUNT)	"105.66"

"4.29" (AMOUNT)	"XXX9" (TEXT)	"4.29"
"4.29" (AMOUNT)	"$Z9.9" (AMOUNT)	"$ 4.3"

On input, all fields in a record being validated must be acceptable according to both the document's and the record file's constraints. Note that the above display rules can cause a field to be displayed in such a way that it is not acceptable for reinput to the record file.

Scrolling

In the first release, the scroll bar will contain regions for jumping to the beginning of the window, jumping just beyond the last data in the window, and scrolling up or down. Because this style of scrolling will never move the window across a record boundary in a form-style (non-tabular) display, two menu commands will be added to the view window for those displays, Show Next and Show Previous. Show Next will display the record following the one currently displayed (unless it is the last in the view, in which case the command will have no effect). Show Previous will "back up" one record, until the first record in the view is displayed. If the record is too long to be displayed in one screenfull, the user may scroll in the standard fashion to see the rest of the form.

In a tabular display, the whole view is considered as one document, and so the user can use the scroll bars to access new records. Rows of the table will scroll in standard fashion, with one exception. If the form contains *only* the table through which the records are displayed (no incidental text or separate computed fields), then the table headers will not scroll with the data. Instead, they will remain at the top of the window, and the table rows will scroll independently underneath them. The Full Tabular Form will behave this way, in particular.

A future release will implement scrolling as described in *System Overview.* At that time, there will be regions in the scroll bar labelled N and P. The semantics of these regions (whether they move the display in units of records, or of screenfulls of data) remain TBD. In the Jumping Region, a number will refer to an approximate percentage within the records in the view (not a page number).

Editing records

Within the view window the user may modify record file data directly using standard text editing commands (see "Change Record") or add new records (see "Insert Record").

The view window menu commands are ?, Close, Close All, and Show View Properties, plus, for a non-tabular form, Show Next, and Show Previous. The Auxiliary Menu commands are Make Document and Set Window, plus Select Row for a tabular form, or Add Record for a non-tabular form.

View Property Sheets

Three property sheets are accessible from the view icon. One specifies relatively stable properties of the view, including its name, its sort order, its basic (*View*) filter, and whether it has an index maintained for it. The second specifies more ephemeral aspects of the view, such as the values and form of display of a secondary (*Retrieval*) filter, and the choice of form through which selected records are displayed. The third is the Record File's Field Summary Sheet. A display choice parameter allows the user to access the two view property sheets or the field summary sheet (for instance, to find out how a rejected data value failed to meet its field constraint). The choices are Transient, Stable, or Field Summary. The Transient sheet is displayed when the user first shows properties for a view.

Stable View Properties

Properties which are considered fundamental to the view, and which may be expensive to change, are stored and displayed on the Stable view Property Sheet (see Figure RP-5). The Stable view property sheet menu commands are ?, Done, Apply, Defaults, and Insert Name. [Apply and Insert Name may not be provided in the first release.] The properties on this sheet are

Name - This text parameter is the name of the view.

State of the View Index - Star can create and maintain an internal data object called an *index* to support each view. Star is willing to maintain any number of these indices. However, each time a record is changed or added to the record file, each index must be updated. Hence, the customer will want to control the number of indices that Star must maintain. Star provides a state parameter called Save Index on the view property sheet to permit this. If this parameter is on, then the view is saved and maintained as just described. If it is off, Star will remember the view description, but will eliminate the index for it whenever it is not the current view. Hence, when such a view becomes the current view and the user issues an Open on that view, there will be a delay while Star reconstructs its index. Such a strategy would be appropriate for seldom-used views.

View Size - When an index is supported for a view, an informational parameter is also provided that indicates how many records are in the current view (as defined by its view filter), as well as the total number of records in the record file. If there is no index, then no size information is displayed.

View Filter - Each view is defined to include some subset (possibly all) of the records in the record file; this subset is defined by the view filter. Star may arrange the underlying records, or maintain auxiliary indices, to support restriction of the view to those records. Records which do not pass the view filter are never seen through the view. (A second, less ponderous filtering operation may be specified via the Retrieval Filter; see Transient View Properties.) The view filter is displayed in the property sheet as a set of rows in a table; fields are edited here as in a standard table. [In the first release, it may be necessary to Confirm Changes on the filter, as on a record in the view window.]

Changes to the View Filter will cause changes to be made in the view index (see below) when Done or Apply is executed. This may be a time-consuming task, and is not expected to be done frequently. The filter is a permanent part of a view: the specific settings of the filter remain in effect until the user changes them on the view property sheet.

Include in View Filter - The user may select which fields of the record file will be displayed in the view filter by this column of state parameters. Each field in the record file is named in a state parameter. If that parameter is on, then that field is included in the filter; if it is off, the corresponding column is omitted from the filter display. This facility is provided for two reasons: 1) usually the user will restrict his interest in filtering to a small subset of all the fields in the record file; and 2) if all the fields were shown simultaneously in the filter, the visual presentation could be cluttered and the manipulations awkward (for some record files, the filter would have to be scrolled horizontally just to see all the entries).

Star rejects an attempt to omit a filter field which contains a pattern. Thus it is not possible to have hidden filters controlling the view.

[This facility will not be available in the first release of Star; the filter will always display the full tabular format, and this column of state parameters will not be displayed.]

Sort Order - A sort order defines the order in which records (and rows of REPEATING columns) are presented to the user. The sort order is a property of a view, and is by no means inherent in the data itself. Changes to the sort order may cause Star to make changes to the view index (see above) when Done is executed. There is a sort order specification for the view as a whole and one for each REPEATING field within the view. All sort orders are specified on the Stable view property sheet.

A sort order specification is a list of rows, one for each field considered in the sort. [In the first release, only one row may be used in each sort specification.] Each row consists of a text parameter which names a field in the record, and a choice parameter which governs the order on that field (A-Z or Z-A). For TEXT fields, sorting is by character position within the character set (ignoring case); for DATE fields, sorting is chronological; and for AMOUNT fields, sorting is by numeric value. A user specifies the sort order by entering the name of the field on which to sort and selecting the desired direction; see "Set Sort Order".

[In some release of Star after the first, it will be possible to define a sort specification on more than one field. Star will then maintain an empty row at the bottom of the list, to allow the user to add rows to the list; when the user enters a field name into this row, another empty row will be generated below it. The field named in the first row of the list is major ordering field; each subsequent sort field determines the ordering within that defined by all previous sort fields - that is, it serves only to break ties on all fields listed above it.]

Any field specified as a sort field must either be directly contained in the record to be sorted or it must be contained in a chain of SINGLE DIVIDED fields, the outermost of which is contained in the record to be sorted. For example, in the following structure:

Salesman Name	(SINGLE DIVIDED field)
Last	(TEXT field)
First	(TEXT field)
Customer Number	(AMOUNT field)

Orders	(REPEATING DIVIDED field)
Part Number	(TEXT Field)
Quantity	(AMOUNT field)

"Salesman Name.Last", "Salesman Name.First" and "Customer Number" are valid sort fields for the records in the file. "Orders.Part Number" and "Orders.Quantity" cannot be used as sort fields for the file as a whole because they are contained in a REPEATING DIVIDED field – there may be several different values of these fields for each record to be sorted. However, these two fields *may* be used as sort fields for the *subrows* in the "Orders" field. Similarly, "Salesman Name.Last" may not be used to sort Orders (even if it were allowed, it would never have any effect, since it has the same value for each collection of Orders), but it is a likely sort field for the view as a whole. [All of this paragraph is moot until provision of tables allows divided fields to be defined in a record file.]

Transient View Properties

View properties which are considered relatively volatile and inexpensive to change are stored and displayed on the transient view property sheet (Figure RP-6). The transient view property sheet menu commands are ?, Done, Apply, and Defaults. [If Apply is not generally implemented in Star, an explicit Apply menu command will be provided for the Transient view property sheet.] The properties on this sheet are:

Retrieval filter - The second layer of filtering on a record file is called the *Retrieval Filter*. It is applied in addition to the current View Filter; that is, it further restricts the set of records which can be seen in a view. The retrieval filter is intended for expressing transitory queries, e.g. for selecting an individual record in the view for display or update. The retrieval filter of a view is reset to null when its view is no longer the current one.

Data Display Form Choice - The user may select among all the forms contained in the record file's forms folder for displaying the records in the view. The names of all the forms are presented as a set of choice parameters.

Include in Retrieval Filter - As with the view filter, the user may select which fields are displayed in the retrieval filter. This facility is exactly parallel to the Include in View Filter parameters on the Stable property sheet: it presents a column of state parameters naming the fields and indicating whether each is to be displayed in the retrieval filter. As with the View Filter, the user is prevented from turning off display of a field for which a filter pattern is currently specified.

[Also as with the View Filter, the initial release will be restricted to a full tabular view of the Retrieval Filter, and this set of parameters will not appear until that restriction is relaxed in a later release.]

Record File Property Sheets

The property sheets for the record file itself allow the user to inspect and modify basic attributes of the record file (and all its views and data simultaneously). The record file property sheets are accessible from the record file icon; the Field Summary sheet may also be accessed from a View Property Sheet. These property sheets include controls for status and statistics, and the basic field structure of the record file.

Status and Statistics Property Sheet - This property sheet contains miscellanous summary information for the record file as a whole. See Figure RP-10. Each parameter has the same meaning as for documents, except that the size parameter is expressed in number of records.

Field Summary Sheet - This property sheet contains information describing the structure of the record file (see Figures RP-11 and RP-12). It contains exactly the same information as the Field Summary property sheet in documents (Figure FLD-7). The manipulations which can be performed on a record file's Field Summary Sheet are the same as on a document's, but in no case may the structure of a record file be modified after data has been stored in it.

Empty Record File

A blank record file is provided in the Directory for creating new record files. It may be copied to the Desktop, where the user may then define its various properties and fill it with data.

ACTIONS

Icon Actions

The same actions apply to record file icons as apply to folder icons: move, copy, delete, print, file, mail, open, close, etc. The semantics are the same as for folders.

When a record file is opened, the list of views is displayed, and the current view is selected.

If a record file icon is moved to a printer, the current view is printed. In addition, a cover page is printed designating the view in effect. This permits a "direct mail" application to be very simply accomplished: the display document is a form letter and the names and addresses are records in the file. Then to print a letter to each addressee, that view is selected, and the record file icon is moved to a printer.

The contents of a record file may also be printed via the "Make Document" command, described below. [Make Document may not be implemented in the first release.]

Select View

The user selects views from within the record file window by pointing with the cursor and clicking the SELECT mouse button. This is exactly analogous to selecting something in a conventional Star folder.

Create View

To create a new view, the user selects an existing view in the record file window and makes a copy. The new view must also reside in the record file window.

Modify View

To modify the properties of a selected view, the user presses PROPERTIES and makes the modifications in the appropriate view property sheet. Note that modifications to a view do not create a new view object. A new view object is created only by copying an existing view object.

If the user already has a view open, the menu command Show View Properties may be used to access the view property sheet in place of closing the view and hitting PROPERTIES. This is particularly convenient for changing the view's retrieval filter.

Move/Copy/Delete View

Views can be manipulated only within the record file window. The privileged view, the Base View, cannot be moved, deleted, or altered. It may be copied to create another view. Normal views may be moved and copied to new locations within the view folder, and may be deleted.

Open Forms Folder

To access the forms folder for the record file, the user selects the forms folder in the record file window and presses OPEN.

Create Form

To create a new form for the record file, the user either copies an existing form within the forms folder, or moves/copies a standard Star document into the forms folder window.

Modify Form

To modify a form for the record file, the user edits the form in the normal Star fashion. This may be done in place within the forms folder, or outside the record file entirely. In this latter case, the form would then have to be moved or copied into the forms folder of the record file in order to be available in views. Note that a form may *not* be changed from within a view.

Move/Copy/Delete Form

Forms can be manipulated either within the forms folder or outside of the record file. The privileged forms, the Defining Form and the Full Tabular Form, cannot be moved out of the record file, deleted or altered. Either of these may be copied to create a normal form. Normal forms may be moved or copied to any legal document location within Star, and they may be deleted.

Open View

A view is opened by selecting a view icon in the record file window and pressing OPEN. When the user opens a record file, the view which was last selected will be selected again. Thus the user may re-access a "standard" view in most cases simply by selecting the record file icon and invoking OPEN twice in succession.

Select Fields in a Record

Since records are always displayed through documents, selecting the contents of fields in a record is the same as selecting the contents of fields in a locked document.

Select Record

If a record is rendered as a row in a table in the view, the user may use conventional row selection to select a record. Multiple records may be selected by extending the selection to include more rows. [Due to implementation restrictions, only records displayed on the user's screen may be selected; scrolling the window will cause records to become de-selected as they move off the screen.] If a record is rendered as an instance of a complete document, the user may select the complete document with multiple clicking in the usual way. Multiple records may not be selected in this case, since only one record's document is displayed at a time. In all cases, only records that pass the filter can be selected, since only those records are displayed on the screen

Insert Record

There are three ways to insert records into a record file: (1) type them in by hand, using Star's field fill-in features; (2) move or copy record data from record sources in an open document or record file window; or (3) move or copy the iconic form of a folder, document or record file to a record file.

(1) Insert records by typing them in

The user may insert a new, blank record in the file. If the view has a tabular form, this is done by first selecting an existing record in the view window and then typing. A new blank record appears immediately after the selected record. The caret is automatically placed in the first field of the new record. The user continues typing data for the fields of the new record. At the end of the record, the new data must be confirmed with Confirm Changes as described in "Change Record" below. As a user convenience, invoking NEXT from the last field of a record confirms changes to that record, and advances the caret to the first field of the next record.

In a document-per-record form, the user is provided with an additional command in the auxiliary menu, Add Record. Invoking Add record causes a new copy of the form to be displayed, with all its fields empty. The caret is placed in the first field, and Confirm Changes and Cancel Changes are added the window menu. The user may now NEXT through the fields of the record entering data, and confirm it when finished.

If the user invokes the NEXT key from the last field of the last record in a view (either kind of form), Star provides the same optimization for entering multiple records as it does for adding rows to a table: Any pending changes are confirmed automatically, and Star inserts a new blank record, positioning the caret in the first field. The user may now enter data in the fields of this record in the normal way. Invoking NEXT in its last field confirms this record and generates a new empty one, as before. This process continues until the user invokes Confirm Changes (instead of NEXT) after completing a record, or invokes Cancel Changes before any data is inserted in the new record. In the latter case, the empty new record is discarded.

If the record file is empty, a record is displayed in which all fields are blank. The user may place the caret in the first field of this record and begin typing, after which the insertion proceeds as described above.

If the window shows an empty record (generated by any of the mechanisms just described) and the user invokes Cancel Changes, the empty record is discarded, and the following record is displayed in its place.

(2) Insert records from an open window

New records may be created in a record file by selecting one or more table rows in an open window and moving or copying them into the destination view window. The table may belong to a simple document, or to a tabular view document for a record file. Multiple records may be generated in a single operation if the selection is extended to include several rows.

Source field values are copied into destination fields with matching field names (see "Field Matching"). Normally a record or row in the source will correspond to a record in the destination record file, but the source might also be a sub-row in a REPEATING DIVIDED column. [The latter facility will not be available in the first release; the sources must be top-level records or rows.] Not all the fields in a source need be present in the destination, and vice versa; only the fields that do correspond will be copied. However, a source is unacceptable if any of its fields has a value that violates the constraints of the corresponding destination field, or if it does not contain a field that, in the destination record file, has the attribute REQUIRED.

If any records are unacceptable to the destination record file, then the entire operation is halted, the first unacceptable record is selected, and the message "The selected row is invalid for the record file." is posted in the Message Area. Records generated before the invalid record was encountered remain in the record file; no subsequent rows are transferred.

(3) Insert records by moving an icon to the record file

One or more records can be inserted from a data icon (document, folder, or record file) in a single operation. The user moves or copies the icon onto the record file (either onto the record file icon if it is in icon form or into a view window). Star will read each source of records contained in the icon in turn, inserting as many records as are indicated by the tables and fields in it. When Star has processed all the contents of the icon, it is returned to its original location. [As with MOVE to the printer, the document is returned to, *not* deleted from, its original location after the data copy is complete.]

A source *document* may contain a collection of fields which correspond to a single record; in this case one record is generated and filled from the document. Alternatively, if a document contains a single table, and no fields outside the table match record field names, then the record structure is matched to the row structure of the table (the document may then generate multiple records).

A source *record file* will generate a destination record for each record in the source's *current view*; the operation is analogous to moving a record file icon to the printer.

A source *folder* may contain documents, other folders, and record files. Each of the objects contained in the folder is processed in turn; nested folders have *their* contents processed, until eventually actual sources of records are encountered, and their contents transferred. [In the first release, folders must contain only documents, and the documents must all have the same field structure.]

Unlike insertion of records from an open window, the occurrence of invalid record sources does *not* interrupt the transfer of data. All acceptable records are transferred to the record file; all unacceptable sources of records are copied into the record file's Error Folder in the following fashion:

1. If the bad source is a document corresponding to a single record, then that document is simply copied into the folder and no corresponding record is inserted in the record file.

2. If the bad source is in a document containing more than one record (i.e. in a table), then a copy of the document is made in which the table contains only those rows which did *not* generate acceptable records. The resulting document, which is a subset of the original document, is then placed in the folder.

3. If the bad source is a record file, it is treated analogously to a table: a copy of the record file is made, containing a copy of the view which was current, and it is filled with exactly those records which could not be transferred.

If a folder containing several different sources of records is moved to the record file icon, and records are rejected, a copy of that folder is placed in the Error Folder. In that copy are placed error containers as described above.

Change Record

Since records are always displayed through a document, editing fields of a record is exactly the same as editing fields of a document (see *Tables* and *Field Fill-in*). The commands NEXT and SKIP operate exactly as they do when applied to tables and fields in documents. When field validation takes place, *both the record file's and the document's constraints are used.* In addition to standard text editing, the user may move, copy, and delete rows in REPEATING DIVIDED fields.

Once a user begins editing a record, he cannot edit any other record in the file or change a view property, such as a filter pattern, until he completes the change. When the user makes his first edit to a record, two additional window menu commands appear: Confirm Changes and Cancel Changes (see Figure RP-7). These commands remain in the window menu until the user invokes one of them. While changing a record, an attempt to edit any other record in that record file causes the error message "A change is in progress. Invoke Confirm Changes or Cancel Changes." to be displayed and the action to be ignored. The same response is given to an attempt to scroll the selected record off the screen, or to close the view window. (If the record is displayed through a form rather than a tabular document, the commands Show Next and Show Previous are removed from the menu, since they are invalid while a change is in progress.)

The user signifies that his changes are complete by means of the Confirm Changes command. At this point, all fields are validated, as described below, and the additional window commands (Confirm Changes and Cancel Changes) disappear. The user may also invoke Cancel Changes, in which case all changes are undone, leaving the record the way it was before the changes were initiated.

While it is being changed, a record remains in its original position on the screen, and no sorting of rows in REPEATING DIVIDED fields takes place. Once Confirm Changes is invoked, the filter and sort order are applied to the record, possibly removing it from the view window. The user may have to modify the view or scroll to the record's new location if he wants to continue to see it.

All field values are validated according to the record's and document's constraints at Confirm Changes time. In addition, the following incremental validations take place:

> An individual field is validated according to the document's and record's constraints whenever NEXT or SKIP is used.
>
> When a row is copied into a REPEATING DIVIDED field, all fields in the new row are validated according to the record's and document's constraints.
>
> Insertion and deletion of rows is prohibited in SINGLE DIVIDED fields.

If any fields are invalid, a message is displayed in the Message Area and the Confirm Changes command is rejected. The first invalid field is selected, so that the user can make the appropriate corrections. No changes are made to the record file; either all of the changes go through or none of them. The user must either correct the invalid fields or cancel the change altogether by means of the Cancel Changes command.

Move/Copy/Delete Record

A record displayed in a tabular form can be manipulated as a unit by selecting it and invoking MOVE, COPY or DELETE. Multiple record selections are also permitted, as discussed above under Select Record. These commands operate exactly as in documents. MOVEs, COPYs and DELETEs of entire records do not have to be confirmed.

For records displayed through a non-tabular form, the user effectively selects a record by multi-clicking to the whole document. MOVE and DELETE will remove the record from the record file; MOVE and COPY will transfer its data to the destination. If the destination is a table or record file, a new row/record is created, and its fields filled from the fields in the record. If the destination is a document, the entire text of the form (both fields and surrounding document filler) is transferred to the destination.

Subparts of a record can also be manipulated, as shown in Figure RP-7. Anything that can be selected can be moved, copied or deleted. However, a user may not move any part of a record to any other record in the same record file. (Changing two records at once is prohibited.) Note that moving an entire record to another part of the same file has no effect, since the ordering is dependent on sort fields.

Whether the source entity is a row of a REPEATING DIVIDED field or an entire record, it may become either in the destination. For example, a destination record file may have the same structure as the row structure of a REPEATING DIVIDED field in the source record file; in this case, rows in the source file may be copied into the destination file, yielding complete records. If the situation were reversed, entire records in the source record file could be copied into a REPEATING DIVIDED field in a record of the destination record file; *this* change would not take effect until Confirm Changes is invoked. [The first release will not provide this general COPY facility; only the following will be provided: (a) move or copy of complete records from one file to another, or within a file, and (b) copy of rows of REPEATING DIVIDED fields within a file.]

Show View Properties

The properties of a view are displayed by selecting a view in the record file window and pressing the PROPERTIES key. The Transient property sheet for the view is shown, as in Figure RP-6. The Stable view property sheet, depicted in Figure RP-5, may then be accessed by selecting its display parameter at the top of the property sheet window. The same effect may be achieved for an open view by invoking Show View Properties in its window menu.

Set Stable View Properties

The following actions are possible within the Stable view property sheet (bearing in mind that the properties of the Base View may never be altered):

Set View Name - The user names the view by typing text into the Name text parameter.

Set View Filter - The user enters field patterns (see *Field Patterns*) directly into the fields of the view filter. The filter takes effect when DONE or APPLY is invoked. DEFAULTS clears the filter, so that all records pass. [Changes to the filter may have to be confirmed, just like changes to the data, in the first release; eventually, no confirmation will be required. Apply may not be available in the first release.]

Set View Filter Fields - The user selects those fields of the record file which he wishes to appear in the filter. Every field in the record file appears in this set of state parameters; setting one on causes inclusion of a corresponding column in the filter table; setting one off deletes the corresponding column from the filter. No column which currently has a filter pattern in it may be turned off; neither may a DIVIDED field if one of its sub-fields is turned on. Inclusion of a sub-field of a DIVIDED field will cause the parent field to be included as well.

Note that there is no requirement that the fields available in the filter be those available in the view form. (If a field's filter pattern is a constant value, there might be good reason not to display that field's data, since it will be the same in all visible records.)

[In the first release, only the full tabular form of filter will be provided, and so this set of parameters will not appear.]

Set Sort Order - As described in "Sort Order" under "Stable View Properties", above, there is one sort order list for the record file as a whole and one for each REPEATING column in the record file. Each such list determines the order in which its corresponding records (or sub-rows) are presented. The same actions are used to set the order of records in the whole view, and of subrows within a REPEATING column of a single record.

Each row in a sort list begins with an text parameter naming a field which is to be used to determine the sort order. Adjacent to the text parameter appears a choice parameter indicating the direction in which to sort values in that field (Ascending or Descending). The user specifies a sort order by entering the name of a field in the text parameter of the row and selecting the desired direction.

The user may enter the field name by typing it in, or by using the Insert Name command in the property sheet menu. Insert Name is described in Section 26, *CUSP*; here is a brief description of its operation: The user selects the column or field whose name is required and invokes Insert Name in the window menu; the cursor changes to its text copy form (see figure HOV-4), and the name of the field is inserted as text at whatever destination is selected by the user. [Insert Name may not be implemented in the first release].

The first row in the sort specification list determines major order for the record file or column. Succeeding rows determine minor orders; that is, they serve only to break ties on all more major orders. Star keeps one empty row at the bottom of the list; if the user inserts a field name into it, another empty row is added below it. (The default choice for direction is A-Z.) The major-minor priority of fields in the sort order is changed by moving their names to different rows in the list. [In the first release, only one field may be used in a sort specification.]

The sort specification is validated and applied when the user invokes Done or Apply in the window menu. The names specified are checked to insure they name fields which are valid sort keys for the view (or REPEATING column, as the case may be), and if so, the sort is performed.

DEFAULTS clears all sort orders, so the order of presentation is arbitrary.

Maintain View Index - The user invokes the SAVE INDEX state parameter. The system then maintains a permanent data structure internally that makes subsequent accesses to the current view settings much faster. This command will be most useful for frequently used views, where the time required to create the index each time the view is opened would be unacceptable.

Defaults - Whenever the user wishes to reset the view property sheet to the default settings, he may invoke the Defaults command in the window menu. This sets the filter and sort specifications to null, includes all fields in the view filter, and sets the display document to the full tabular form.

Done - This is the standard property sheet command that closes the sheet and causes the view itself to be appropriately refreshed.

Apply - This is the same as Done, except the property sheet is not closed.

Set Transient View Properties

The following actions are possible within the transient view property sheet (bearing in mind that the properties of the Base View may never be altered):

Set Retrieval Filter - The user may modify field patterns in the retrieval filter with standard table-editing facilities; invoking Done or Apply initiates updating of the set of records which are displayed in the view window, if it is open.

[If Apply is not implemented in Star in general, then an Apply command will be included in the Transient property sheet menu, with exactly this effect. It is not clear whether this can be provided in the first release.]

Set Retrieval Filter Fields - The user determines which fields are available in the Retrieval Filter in a manner exactly the same as for the View Filter, with the same restrictions and implications. (See "Set View Filter Fields".)

Set Data Form Choice - The user selects from a set of choice parameters which name all the forms in the forms folder; subsequent display of records will be through the selected form.

Defaults - Invoking the Defaults command in the window menu sets the retrievai filter to null, includes all fields in the filter, and sets the data form choice to the full tabular form.

Done - This is the standard property sheet command that closes the sheet and causes the view itself to be appropriately refreshed.

Apply - This is the same as Done, except the property sheet is not closed.

Make Document

[Make Document may not be implemented in the first release.]

A user may generate documents with the same contents as a view window. To accomplish this, the user invokes the Make Document command in the auxiliary menu for the view window. The cursor changes to either a folder or document icon (see below); the user may then indicate the destination of the folder or document in a manner identical to MOVE and COPY. When the destination is indicated, the folder or document is created and moved to the appropriate location.

The document or folder that is generated consists of all records in the file that are available in the current view, as restricted by the retrieval filter, rendered through the view form. In other words, the report will look similar to what is displayed on the screen.

If the current display document contains only a single table and no other fields matching the record file structure, then only a *single document* will be generated, with a *row in the table* for each record in the view; the cursor is in the shape of a *document icon* while the user is indicating the destination. If the current display document does not meet the above restrictions, then the report that is generated is a *folder*, containing a *document* for each record in the view; in this case, the cursor takes the shape of a *folder icon*. This is true even if there happens to be only one record in the view. The document or folder produced by the Make Document command is given the name "Record File <name> -- <date> <time>: <n> out of <total> records", as in the Error Folder. In the folder case, the individual documents are not given names.

Open Error Folder

The user accesses the error folder by selecting the error folder in the record file window, and pressing OPEN. The error folder is never visible as a Desktop icon. Moving or copying objects into the error folder is prohibited. However, the user may move or copy objects out of the folder and may delete objects that are already in the Error Folder. He may also open and close them.

Reorganize

The Reorganize command is used to effect reorganization of the data in a record file (reorganization may include such functions as regeneration of access paths, reclamation of unused space, and modification of file structure to take advantage of observed patterns of use). When invoked, the Reorganize option sheet is displayed, allowing the user to select the nature and extent of reorganization desired. [The types of reorganization to be provided, as well as the content and appearance of the Reorganize option sheet, are TBD. Reorganize will not be available in the first release of Star.]

Create Record File

A new record file is created in one of two ways:

Copy Defined Record File - A record file icon whose structure has already been defined is selected and the user invokes COPY. The result is an exact copy of the first record file. Its structure cannot be changed; it contains copies of all the views, forms, and records in the original. If the user empties this second record file, he just has a record file with no data: its structure still cannot be altered. This procedure is used when what is desired is really a *copy* of an existing record file, for instance for backup.

Copy Undefined Record File - the user selects some record file whose structure has not yet been defined, e.g. the blank record file in the Directory, and invokes COPY. This new object is now a usable record file, containing a single view (the Base view); it has no form in its forms folder; and it contains no records. This record file exists in a special, one-time-only state in which its field structure is not yet defined. As soon as the field structure is defined, it changes its status forevermore into an ordinary non-empty record file whose structure cannot be changed.

While the record file is in this Empty state, the user will encounter an extra menu command in the record file window, labelled Define Structure. Invoking this command with a document selected causes the record file's field structure to be set to correspond to that document. The user may wish to then move or copy that document into the record file's forms folder, for documentation or regular use.

Other objects with field structure (a table or portion of a table within a document, or a record file) may also be used as the model [but not in the first release].

If Define Structure is invoked when the selection is not an object with field structure, no action takes place, and the message "Please select an object whose structure can be copied to this record file; then invoke Define Structure." appears in the Message window.

The following automatic conversions will take place in defining the fields of the record file when Define Structure is invoked.

All type ANY fields are converted to TEXT fields.

Any restriction on the number of rows in a REPEATING DIVIDED field is removed.

All fields with fill-in rules are removed; fields with skip-if specifications are included without those conditions.

If the record definition is of the tabular kind and the table fill-in order is by-column, it is changed to by-row.

Print Record File

Record files are printed directly by moving the record file icon to a printer. The records shown in the current view are rendered on the printer with the view form, and the record file icon is returned to its prior location.

The contents of record files can be printed indirectly by means of the Make Document command. In this case, the new document or folder is created explicitly first, and then is moved to the printer (or mailed, filed, etc).

Mail Record File

Record files are mailed by moving or copying them to the outbasket. In all respects they behave like the other data icons when being mailed.

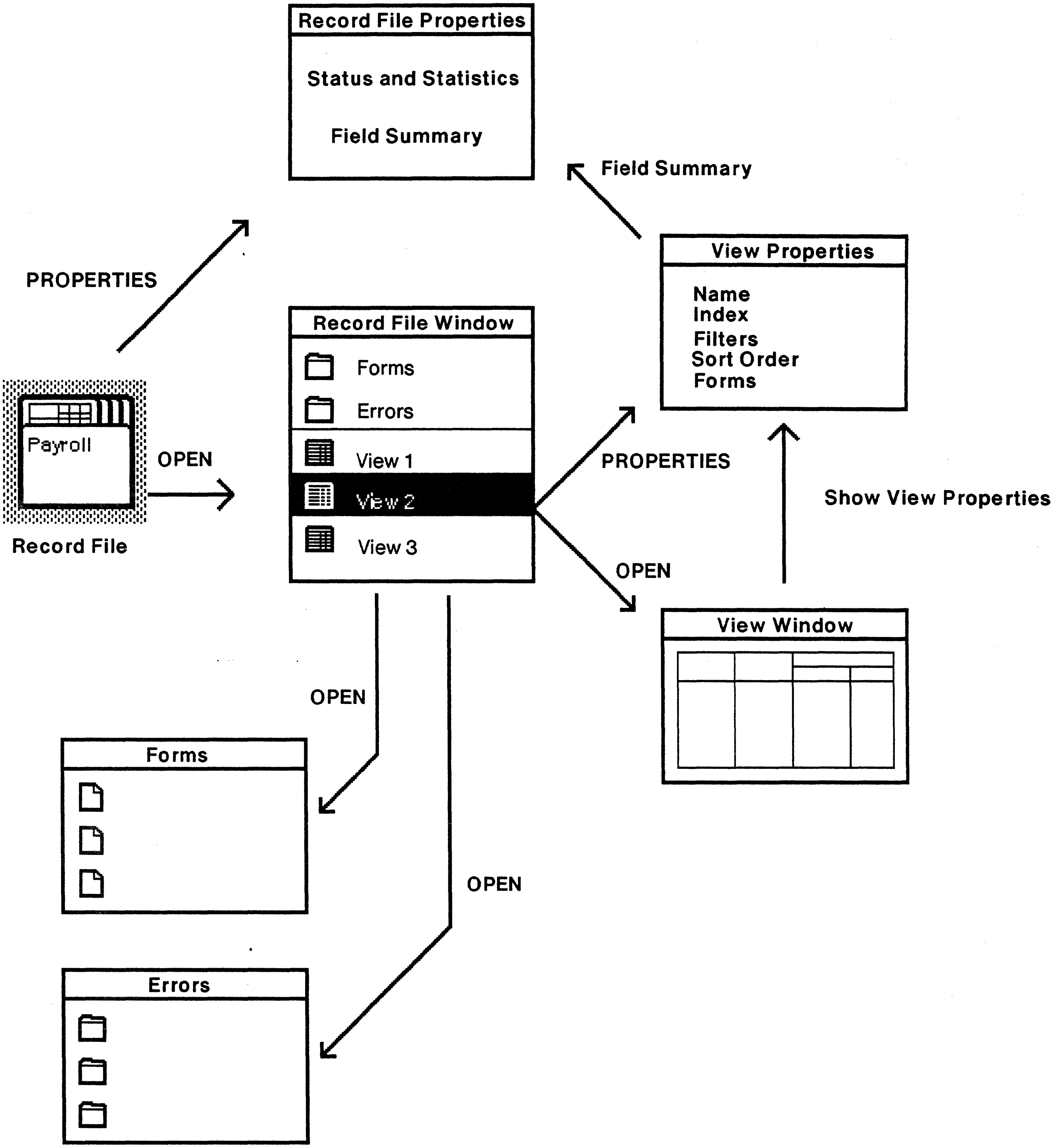
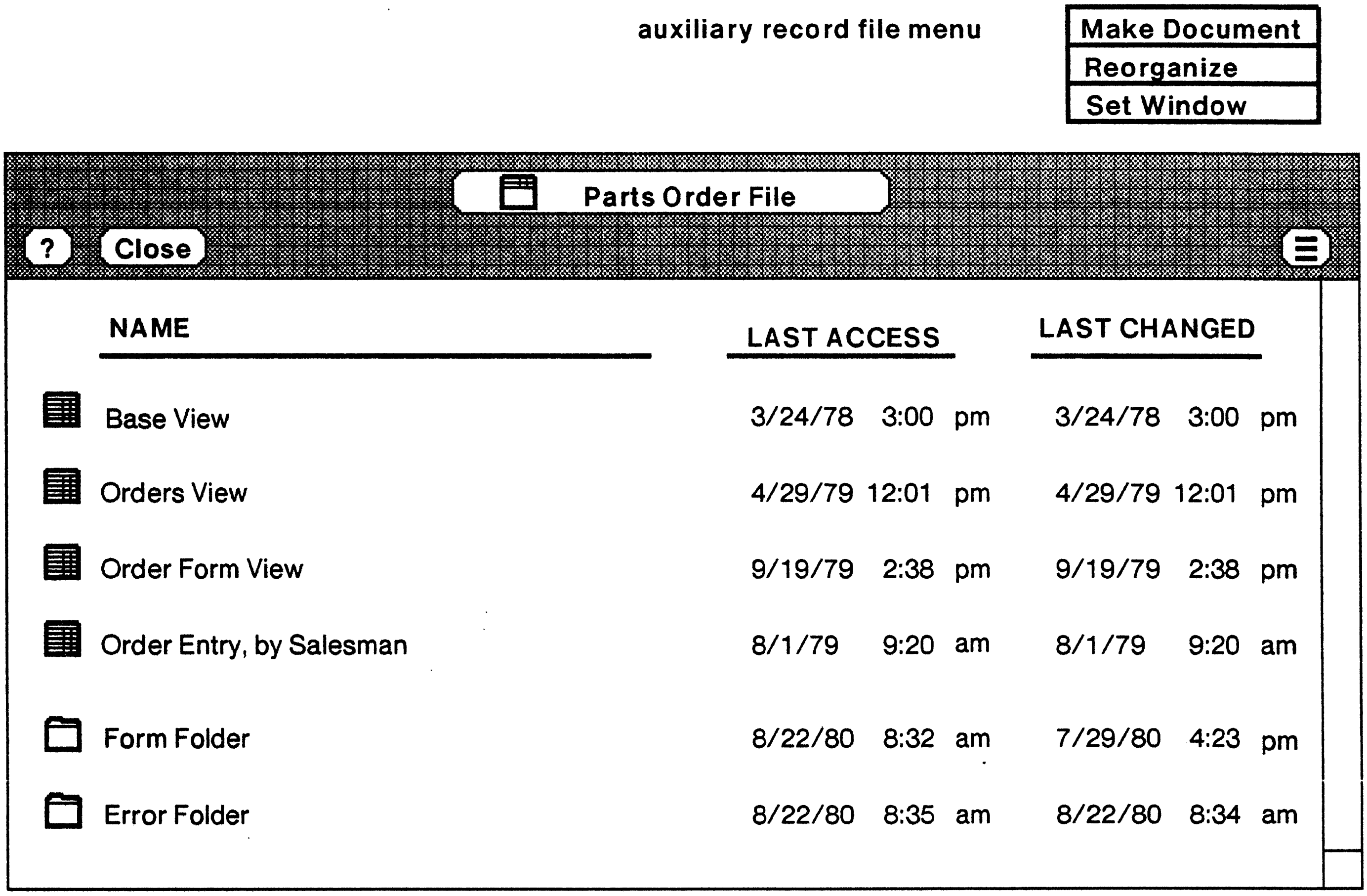
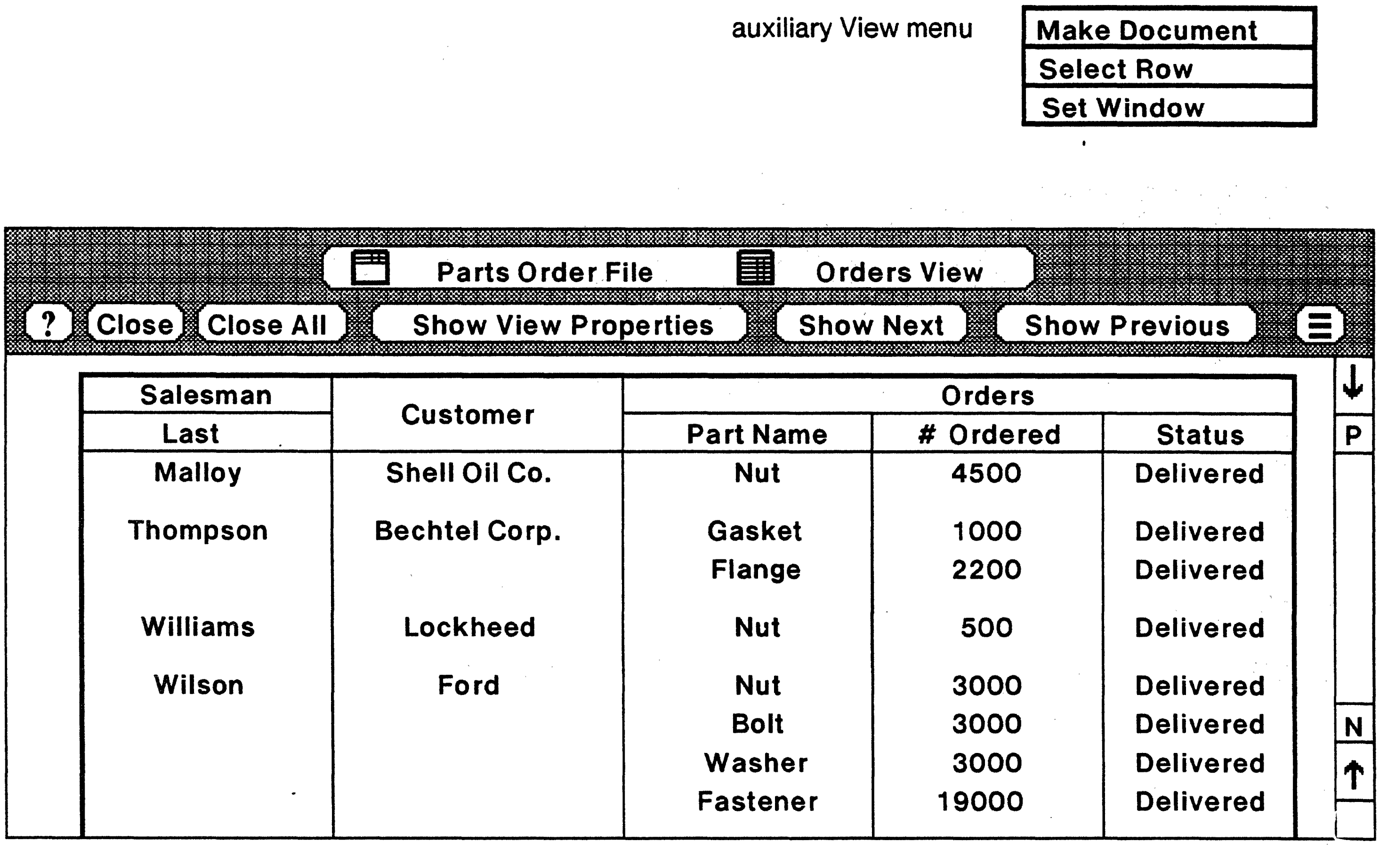


Figure RP-1 -- Relationship Between Objects, Windows and Property Sheets



NAME	LAST ACCESS	LAST CHANGED
Base View	3/24/78 3:00 pm	3/24/78 3:00 pm
Orders View	4/29/79 12:01 pm	4/29/79 12:01 pm
Order Form View	9/19/79 2:38 pm	9/19/79 2:38 pm
Order Entry, by Salesman	8/1/79 9:20 am	8/1/79 9:20 am
Form Folder	8/22/80 8:32 am	7/29/80 4:23 pm
Error Folder	8/22/80 8:35 am	8/22/80 8:34 am

Figure RP-2 -- Record File Window

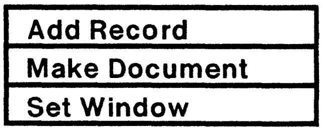


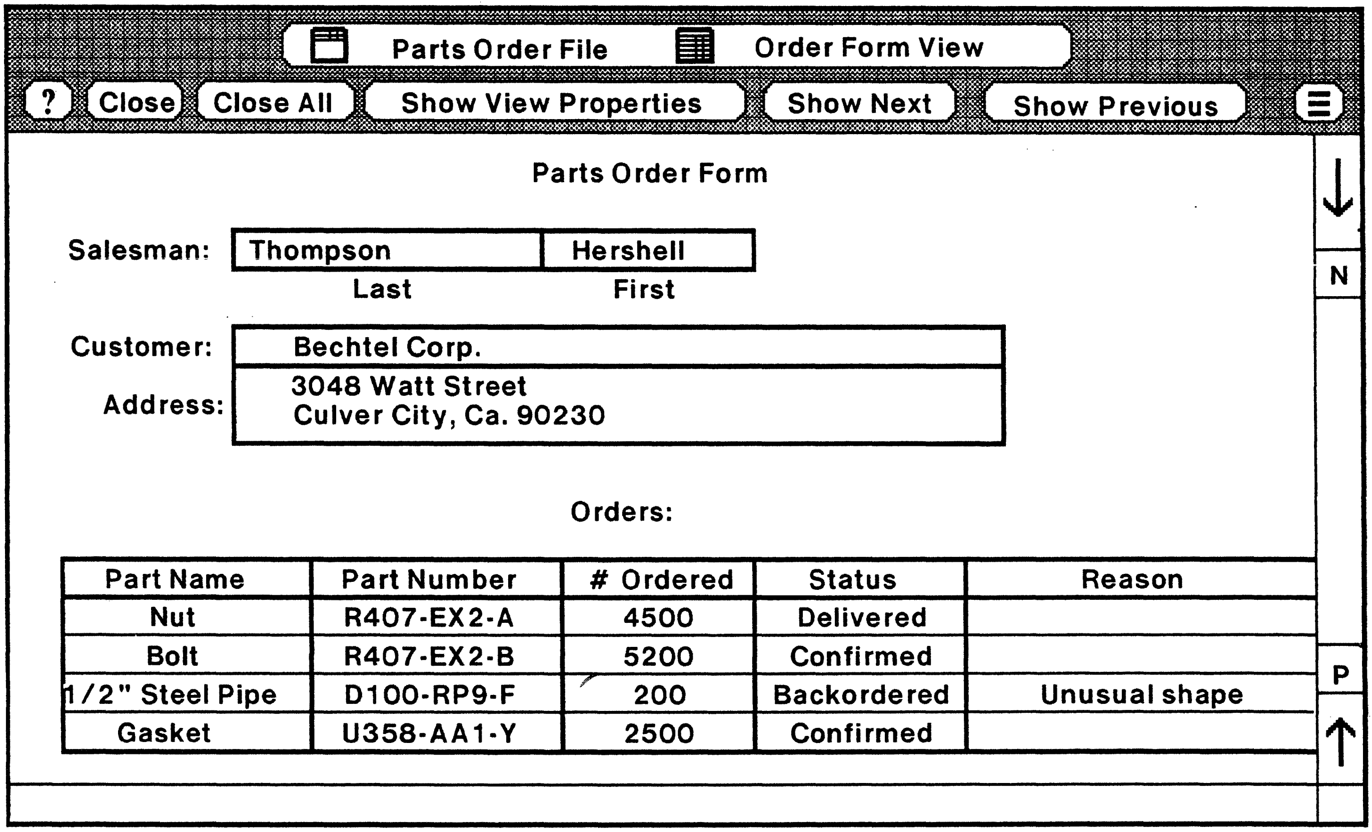
Salesman	Customer	Orders		
Last		Part Name	# Ordered	Status
Malloy	Shell Oil Co.	Nut	4500	Delivered
Thompson	Bechtel Corp.	Gasket	1000	Delivered
		Flange	2200	Delivered
Williams	Lockheed	Nut	500	Delivered
Wilson	Ford	Nut	3000	Delivered
		Bolt	3000	Delivered
		Washer	3000	Delivered
		Fastener	19000	Delivered

Figure RP-3 -- A View Window

(A record file is always displayed through a document.
Here the document contains only a table.)

auxiliary View menu





Parts Order Form

Salesman:	Thompson	Hershell
	Last	First

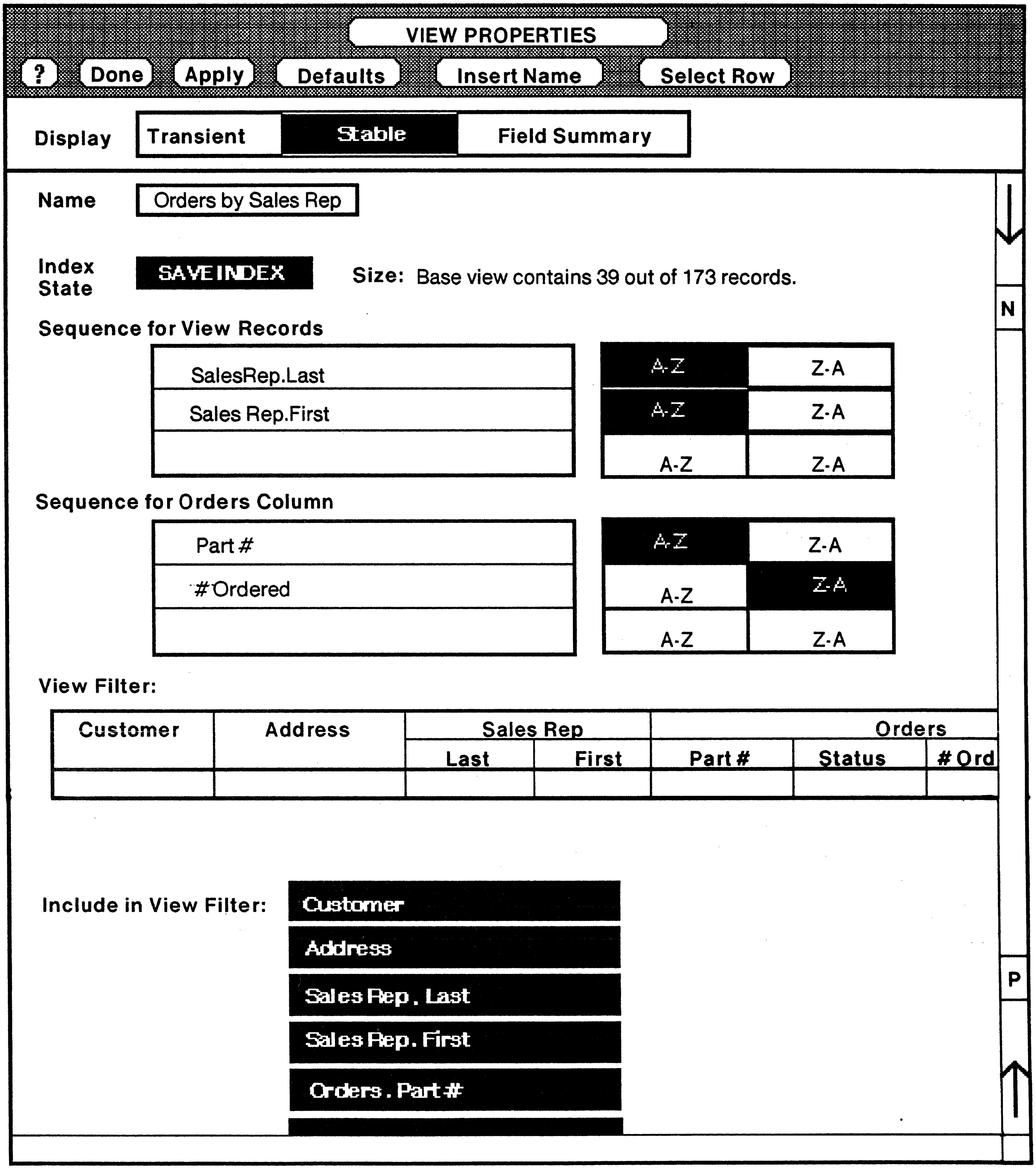
Customer: Bechtel Corp.

Address: 3048 Watt Street
Culver City, Ca. 90230

Orders:

Part Name	Part Number	# Ordered	Status	Reason
Nut	R407-EX2-A	4500	Delivered	
Bolt	R407-EX2-B	5200	Confirmed	
1/2" Steel Pipe	D100-RP9-F	200	Backordered	Unusual shape
Gasket	U358-AA1-Y	2500	Confirmed	

Figure RP-4 -- The Record File Displayed Through a Different View



Display: Transient [✓ Stable] Field Summary

Name: Orders by Sales Rep

Index State: SAVE INDEX

Size: Base view contains 39 out of 173 records.

Sequence for View Records

SalesRep.Last	[✓ A-Z]	Z-A
Sales Rep.First	[✓ A-Z]	Z-A
	A-Z	Z-A

Sequence for Orders Column

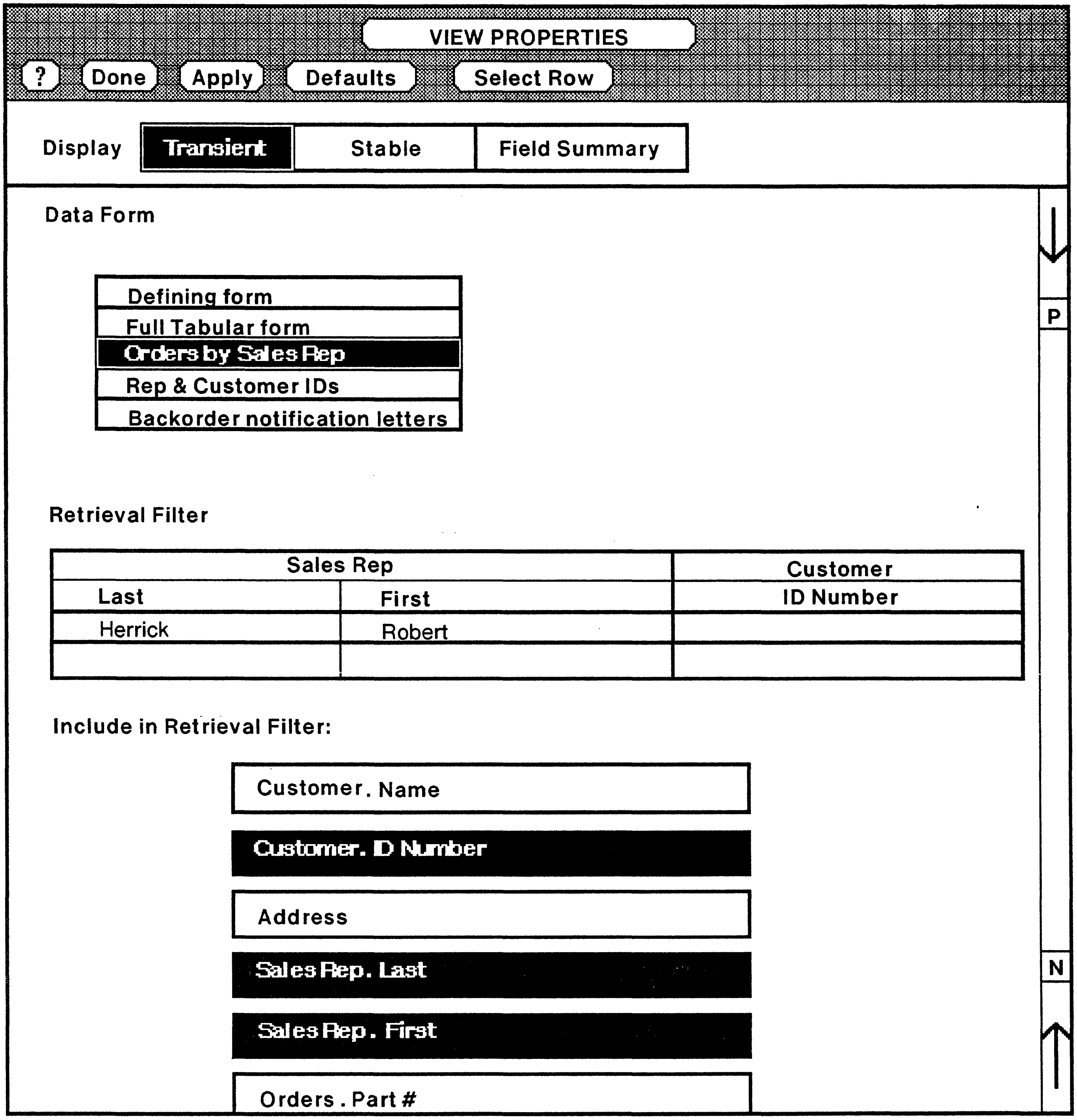
Part #	[✓ A-Z]	Z-A
#Ordered	A-Z	[✓ Z-A]
	A-Z	Z-A

View Filter:

Customer	Address	Sales Rep		Orders		
		Last	First	Part #	Status	# Ord

Include in View Filter:
- Customer
- Address
- Sales Rep . Last
- Sales Rep. First
- Orders . Part #

Figure RP-5 -- View Property Sheet
showing stable view properties
(This property sheet is scrollable.)



Display [✓ Transient] Stable Field Summary

Data Form

- Defining form
- Full Tabular form
- [✓ Orders by Sales Rep]
- Rep & Customer IDs
- Backorder notification letters

Retrieval Filter

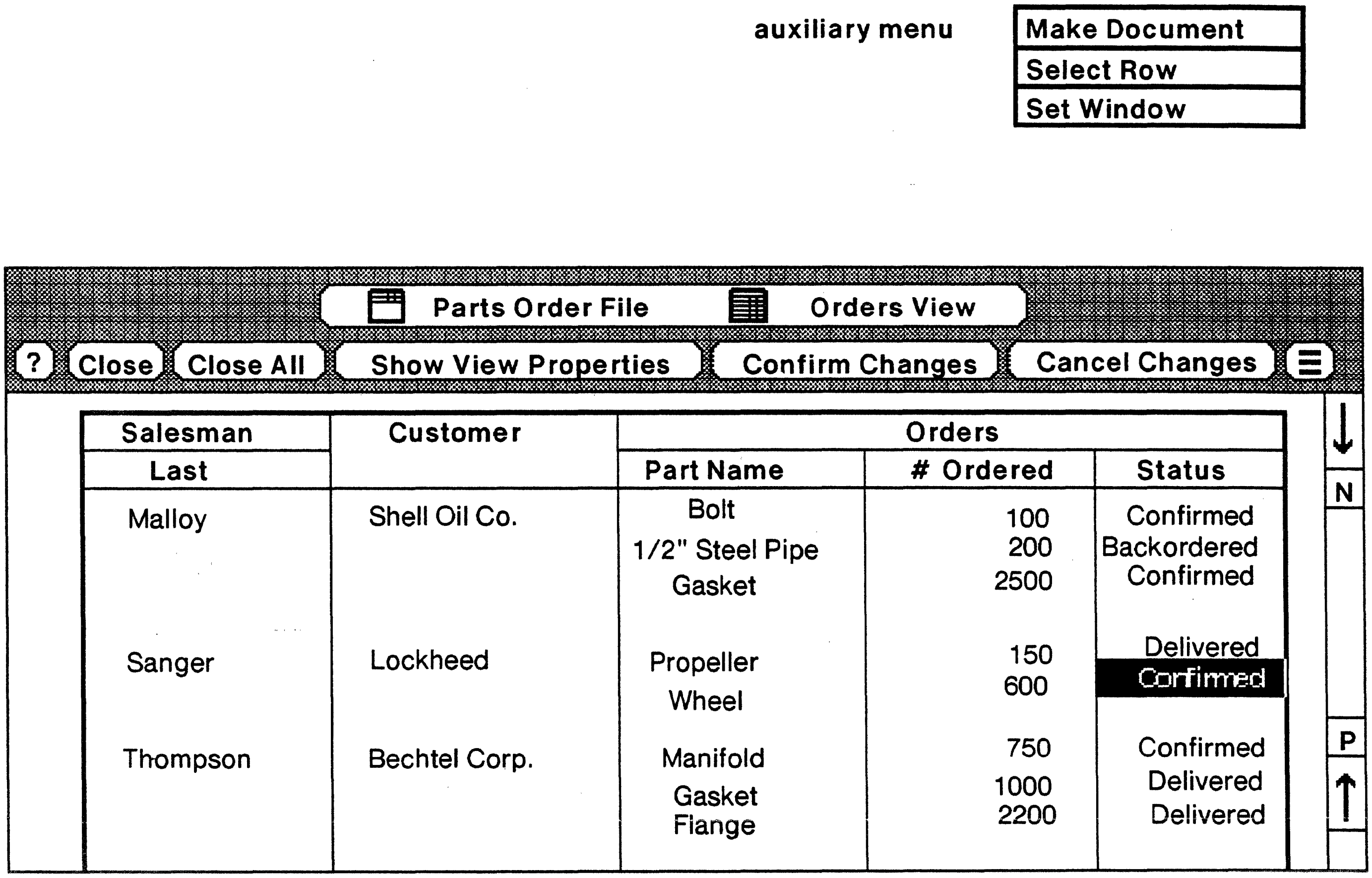
Sales Rep		Customer
Last	First	ID Number
Herrick	Robert	

Include in Retrieval Filter:

- Customer. Name
- [✓ Customer. ID Number]
- Address
- [✓ Sales Rep. Last]
- [✓ Sales Rep. First]
- Orders . Part #

Figure RP-6 -- View Property Sheet
showing transient view properties

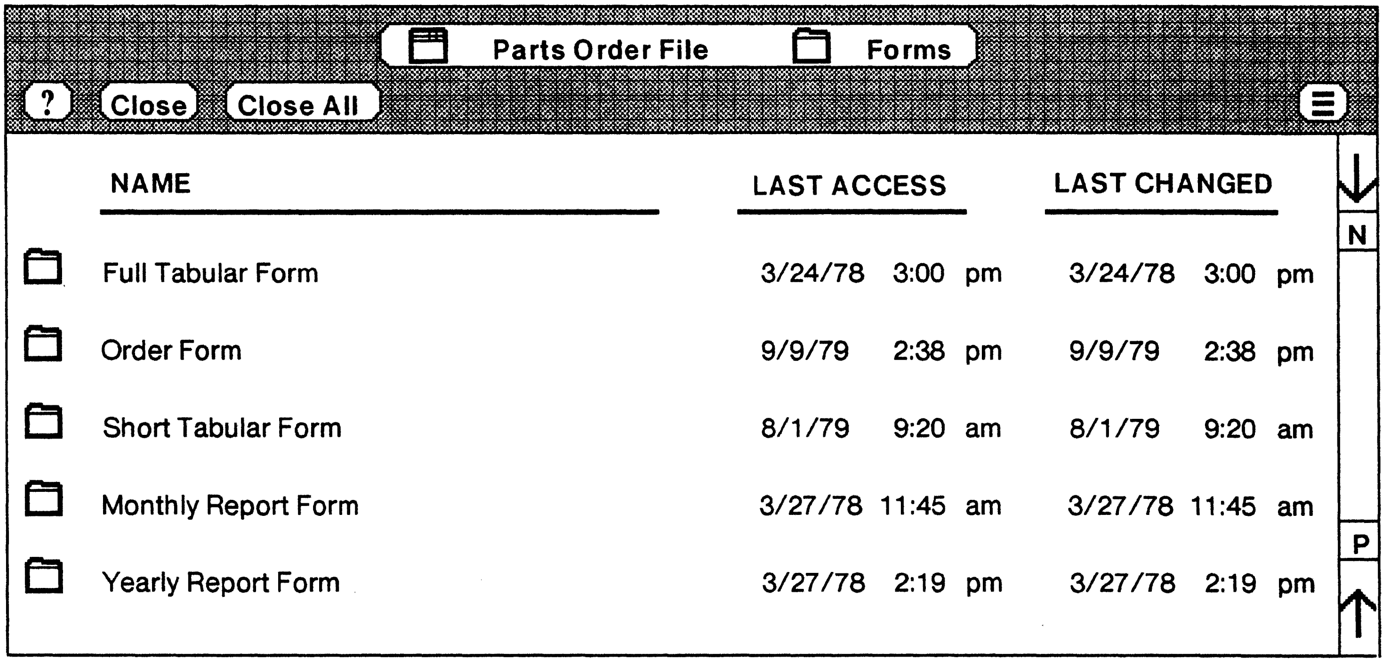
(This property sheet is scrollable.)



Salesman	Customer	Orders		
Last		Part Name	# Ordered	Status
Malloy	Shell Oil Co.	Bolt	100	Confirmed
		1/2" Steel Pipe	200	Backordered
		Gasket	2500	Confirmed
Sanger	Lockheed	Propeller	150	Delivered
		Wheel	600	Confirmed
Thompson	Bechtel Corp.	Manifold	750	Confirmed
		Gasket	1000	Delivered
		Flange	2200	Delivered

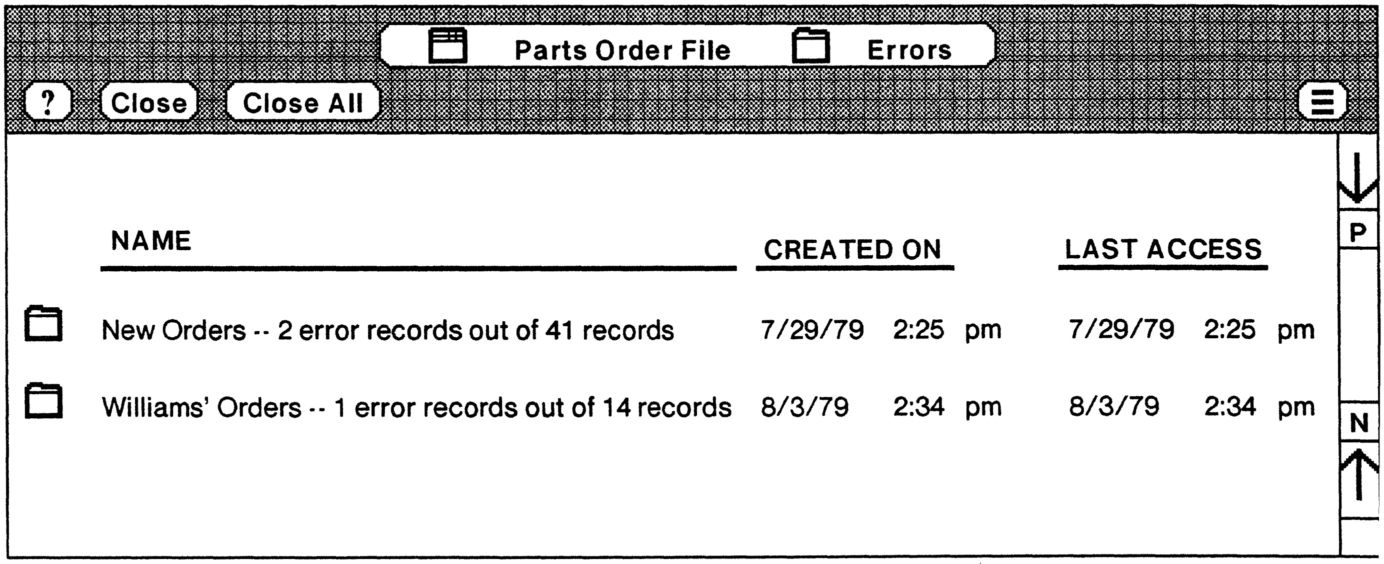
Figure RP-7 -- Changing a record

(Show Next and Show Previous are replaced by Cancel and Confirm Changes, which appear when a record is being editied. The window may not be scrolled or closed while a change is in progress.)



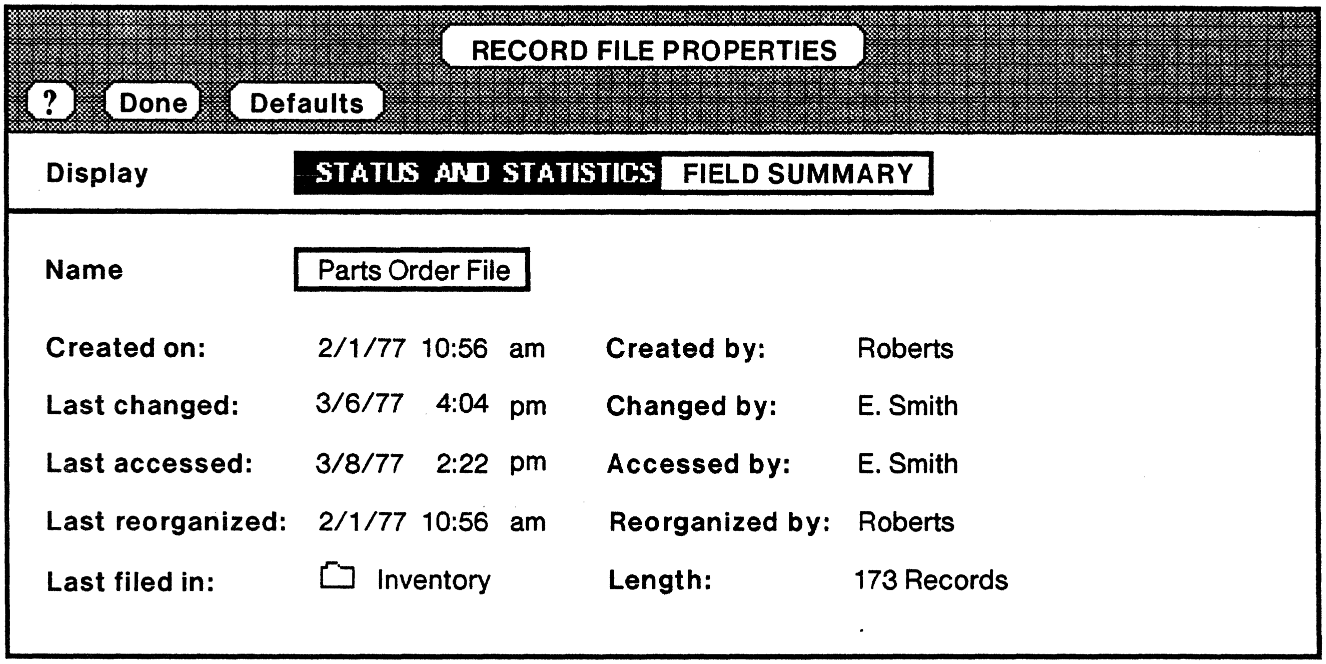
NAME	LAST ACCESS	LAST CHANGED
Full Tabular Form	3/24/78 3:00 pm	3/24/78 3:00 pm
Order Form	9/9/79 2:38 pm	9/9/79 2:38 pm
Short Tabular Form	8/1/79 9:20 am	8/1/79 9:20 am
Monthly Report Form	3/27/78 11:45 am	3/27/78 11:45 am
Yearly Report Form	3/27/78 2:19 pm	3/27/78 2:19 pm

Figure RP-8 -- Forms Folder



NAME	CREATED ON	LAST ACCESS
New Orders -- 2 error records out of 41 records	7/29/79 2:25 pm	7/29/79 2:25 pm
Williams' Orders -- 1 error records out of 14 records	8/3/79 2:34 pm	8/3/79 2:34 pm

Figure RP-9 -- Error Folder

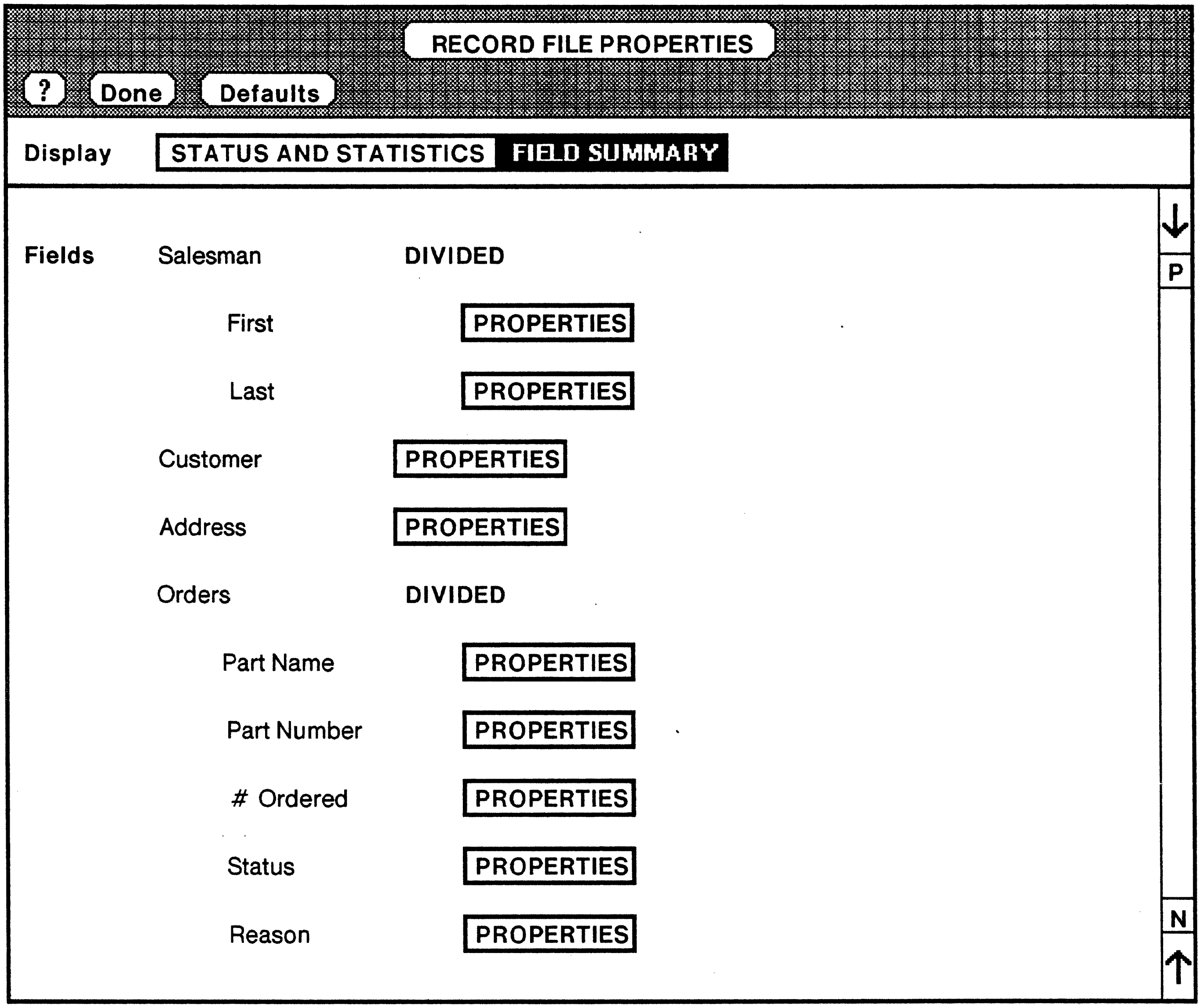


Display: **STATUS AND STATISTICS** | FIELD SUMMARY

Name: Parts Order File

Created on:	2/1/77 10:56 am	Created by:	Roberts
Last changed:	3/6/77 4:04 pm	Changed by:	E. Smith
Last accessed:	3/8/77 2:22 pm	Accessed by:	E. Smith
Last reorganized:	2/1/77 10:56 am	Reorganized by:	Roberts
Last filed in:	🗀 Inventory	Length:	173 Records

Figure RP-10 -- Record File Status and Statistics Property Sheet

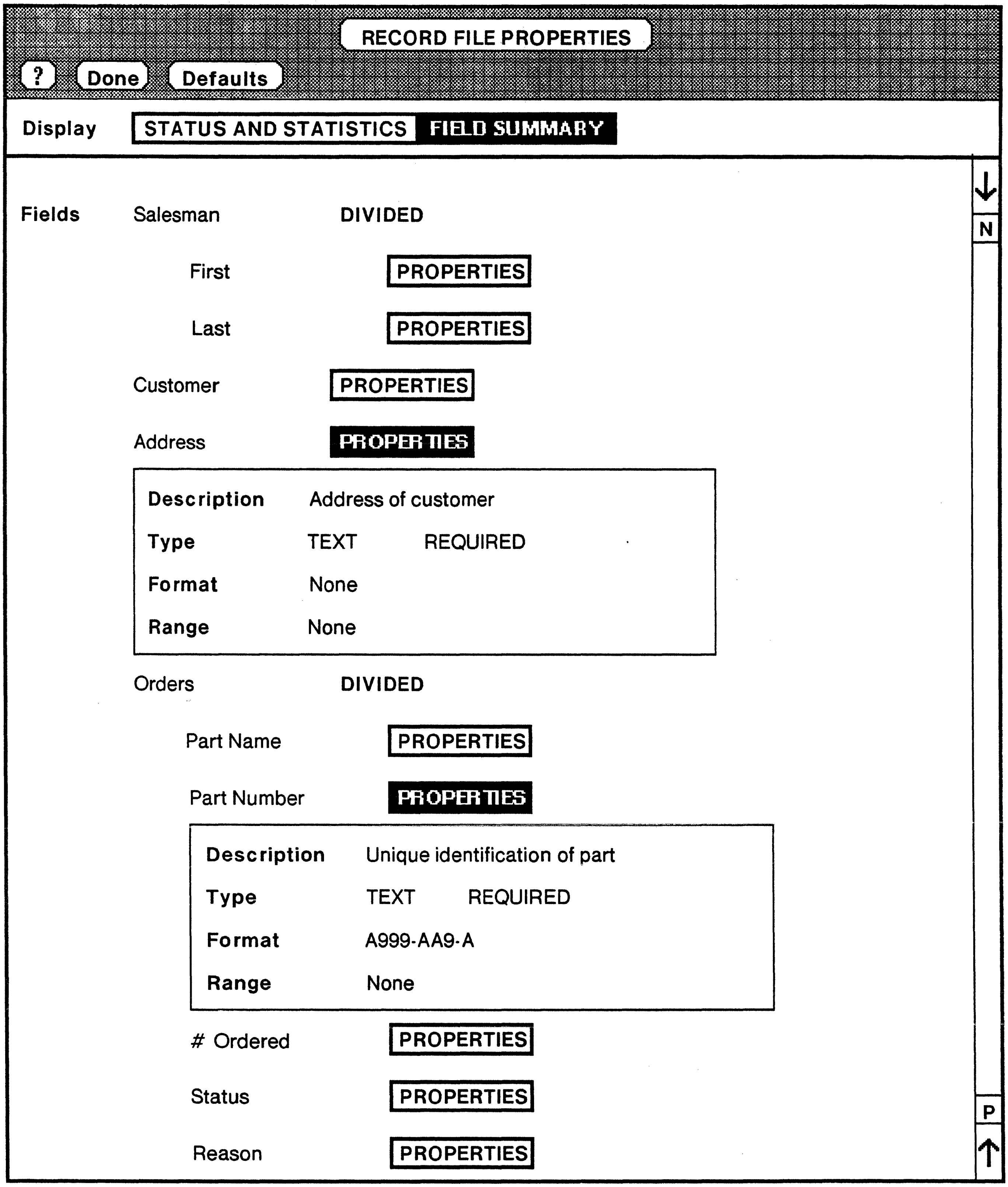


Display STATUS AND STATISTICS **FIELD SUMMARY**

Fields
Salesman DIVIDED
First PROPERTIES
Last PROPERTIES
Customer PROPERTIES
Address PROPERTIES
Orders DIVIDED
Part Name PROPERTIES
Part Number PROPERTIES
Ordered PROPERTIES
Status PROPERTIES
Reason PROPERTIES

Figure RP-11 -- Field Summary Property Sheet

(This property sheet is scrollable.)



Display STATUS AND STATISTICS **FIELD SUMMARY**

Fields Salesman DIVIDED

First PROPERTIES

Last PROPERTIES

Customer PROPERTIES

Address **PROPERTIES**

Description	Address of customer	
Type	TEXT	REQUIRED
Format	None	
Range	None	

Orders DIVIDED

Part Name PROPERTIES

Part Number **PROPERTIES**

Description	Unique identification of part	
Type	TEXT	REQUIRED
Format	A999-AA9-A	
Range	None	

Ordered PROPERTIES

Status PROPERTIES

Reason PROPERTIES

Figure RP-12 -- Field Summary Sheet With Several Field Properties Displayed

26. CUSTOMER PROGRAMMING

This section will describe CUSP, the customer programming language. The section is in preparation, and will be included with the next revision to this specification, along with companion changes to the *Field Definition* and *Calculator* sections.

[*Stored Command Sequences*, which previously occupied this section, has been superseded by *Customer Programming*.]

(PAGES 355 THROUGH 382 ARE INTENTIONALLY BLANK)

B. THE INTRODUCTION

The Introduction is a program that interprets and executes a set of commands that guide a new Star user's first interaction with the system. This section describes these commands and the related facilities that enable the Xerox writer (the person who constructs training materials) to control both how information and instruction are presented to the user, and how the system monitors the user's responses.

USER OVERVIEW

The set of commands to be incorporated into The Introduction is called the *writer's file*. The Xerox writer creates his writer's file (a Star document) on his Star system. After he has tested and revised it, that file becomes the source of the commands that will be executed on the customer's system, presenting instruction to new users.

It is assumed that every new Star user will complete the Introduction, after which he will have gained an understanding of the most basic aspects of operating Star, some practice in the mechanics of the system such as pointing with the mouse, and finally, a degree of confidence in his own ability to make the machine work. Unlike the main body of Star on-line training, the Introduction is capable of evaluating the user's actions at an atomic level, and of slightly modifying the "normal" Star system behavior based on that evaluation, so that potential confusion is minimized. These two features make it possible for the writer to create a friendly, protected, and most of all, *informative* environment in which the new user can practice operating Star with very little risk. The facilities available to the writer also make it possible to create certain conditions on the user's Desktop under program control so that the results of actions can be visually demonstrated to the user.

Subject to Marketing plans and/or customer convenience, we expect that most new users will have received a human-directed introduction to Star before beginning the Introduction. However, since some users will receive no Xerox training before setting hands on the machine, the Introduction will present a sufficient amount of information and practice to allow all users both to begin useful work on the system and to continue constructively with the bulk of on-line training. While the program provides the facilities necessary to monitor the user's actions in all the "basic" domains of Star, it is the responsibility of the writer to include the appropriate instructional content, sequencing, etc.

DETAILED SPECIFICATION

OBJECTS

"Introduction" Desktop

When a new user logs on to Star for the first time, the Desktop with which he interacts is different from the "normal" Star Desktop described in the *Desktop* chapter. The difference is that this "Introduction" Desktop is the environment in which his first session of training is carried out. As described in detail below, only a specified set of actions can be performed in this Introduction Desktop. In general, the user is required by the Introduction to perform specified actions, on instruction from the writer (via commands in the writer's file). When the user has completed the training specified by the writer's file, the "normal" Star Desktop replaces the Introduction Desktop and the user has access to all the objects and actions described elsewhere in the Functional Specification.

Introduction Window

The *Introduction Window* is displayed on the right side of the LF screen, with its upper left corner at (536, 32). Its width is 488 pixels, and its height is 672 pixels, leaving enough space on the Desktop below the Introduction Window for icons to appear.

The Introduction Window's menu commands are Continue, Restart Practice, Restart Unit, and Restart Previous Unit. The menu commands are described in "Progressing Through the Introduction", under "ACTIONS". The contents of the Introduction Window are read-only, and the window cannot be scrolled. Its size cannot be changed.

The text that the writer expects the user to see and respond to falls into two categories:

> **Information and Instructions** appear in the body of the Introduction Window. These are all the words that the writer uses to tell the user about the system, and to describe the actions that the user is expected to carry out. The evaluation of his actions is described under each of the commands below.
>
> **Feedback** appears in the top three lines of the Introduction Window, just below its header. It is the writer's response to the user's action, telling him whether his action was correct or wrong, perhaps including a brief description of the nature of the error.

There is no icon corresponding to the Introduction Window. The contents of the Introduction Window are created by means of the NewInfo command in the writer's file; thus the text in the Introduction Window changes as execution proceeds through the writer's file, and the user progresses through the Introduction.

Interaction Windows

The space on the left side of the screen is used for *Interaction Windows* that are opened either by a writer's command or by the user, on instruction from the writer. These windows are a maximum of 496 pixels wide, so that they cannot overlap the Introduction Window. Their x-coordinate is 0; their y-coordinate and height depend on how many Interaction

Windows are open. A maximum of three windows may be opened, so that they may all appear on the left side of the screen.

Demonstrations of the results of Star actions (e.g., selection) will appear in an Interaction Window. (Demonstrations of icon-related actions appear in the Icon Area.) An Interaction Window may also be used as a workspace in which the user practices type-in, selection, etc. Interaction Windows display the contents of normally-created Star documents and folders, so the writer may use them to illustrate aspects of Star with or without instructing the user to perform actions inside them. For example, the writer might use a document containing a graphic illustration to give the user an overview of how Star functions relate to each other, without intending to teach the user anything about graphics.

Icon Area

The space on the Desktop vertically below the Introduction Area is reserved for icons. They appear as the result of the PutUpIcon command, and are under the control of both the writer and the user as described below. A maximum of six icons may appear at any one time; icons that are no longer needed in the instructional situation may be deleted. The "slots" in the Icon Area are numbered 1 to 6, from left to right. The writer specifies the slot in which an icon appears.

Introduction File Drawer

The Introduction file drawer is stored on the Help Server (see *Help*). On a customer's system, its contents can be changed only by a Xerox writer or System Administrator. (The Xerox writer assembles the contents of the Introduction file drawer as described under "ACTIONS".) It contains the writer's file from which the Introduction will be executed, as well as all the supporting documents and folders that the writer's file refers to. For example, the writer may instruct the user to open a specified icon on his desktop, and perform editing actions on the text in the resulting window; the system retrieves the document containing this text from the Introduction file drawer both to display its icon and to show its contents in a window.

Writer's File

The writer's file is a Star document written by a Xerox writer, containing the commands that the Introduction will interpret. The commands are executed sequentially, except as described under "Control Commands". The writer's file is located in the Introduction file drawer.

Commands

The writer's instructional intent is carried out by the execution of the sequence of commands, each of which invokes a single Introduction function, such as presenting an explanation to the user, or causing the program to evaluate a user's action. The kinds of functions include display and demonstration, monitoring the user's actions, and control over the sequence of execution. The commands are described in the following sub-sections: *"Display Commands", "Monitoring Commands", and "Control Commands"*.

A Note on Syntax and Frequently-Used Argument Types

This explanatory section is included, and the format of the commands given as indicated, so that non-implementors can understand the purpose of the argument list.

The actions to be performed by the *user* during the course of the Introduction are normal Star actions, carried out by keystrokes (e.g., pressing the MOVE key) and button clicks (e.g., invoking menu commands). The objects of interest in this specification are those created by the *writer*, as communicated to the Introduction program by the commands in the writer's file. The commands described below constitute the writer's interface to the Introduction facilities, and the required syntax of the writer's file is illustrated and explained by the descriptions in this section.

Since Xerox writers do not need to have any programming experience, the syntax of the commands must be manageable by an intelligent non-programmer. At the same time, it is necessary to provide enough flexibility to the writer so that he can control instructional situations in a variety of ways.

Each command has two parts: its *name* and its *arguments*. (Some commands have no required arguments.) The name of a command describes its function, while the arguments communicate such specifics as the characters that will be displayed, the interaction window in which the user should do his work, etc. The command name must be spelled correctly (though upper-lower case mismatches are not significant), and the arguments must be given in the specified order.

The list of arguments must be contained within matching square brackets ([]). Any sequence of characters referred to as "text" must be surrounded by quotation marks ("); if the writer wishes to include quotation marks as part of the text, they must be shown as a double set of marks (""). Otherwise, arguments within the list are separated by commas. These rules are illustrated in all the examples shown.

> **text** is any sequence of characters, such as "Now move the pointer into the CLOSE menu item, then press the Left button." The word *text* is used as an argument in the specification of commands whose purpose is either to present something that will be displayed to the user, or to evaluate a sequence of characters typed or selected by the user.
> In order to specify a long sequence of characters as an argument to a monitoring command (e.g., a long selection that must exist before the user presses the DELETE key), the writer may use the **ellipsis** [described under the Find command in the *Text Editing* chapter (p. 88)]. For example, if the user were expected to have selected the entire paragraph preceding this one, the writer could either type all of the characters (and probably produce errors), or preferably specify "text is any...by the user" The first occurrence of any text matching that "ellipsis search" would be considered to be the target selection.
>
> A **name** is similar to *text*, in that it consists of a sequence of characters specified by the writer. However, since it may not contain commas, it is not enclosed in quotation marks. It is used here in commands such as PutUpIcon, where the sequence of characters is something other than text for the user.
>
> A **window#** is an integer between 0 and 3 inclusive, and identifies one of the three available interaction windows, listed from top to bottom. The writer is responsible for knowing which windows are open at all times, so that the window# corresponds to the window appropriate to his purposes. Zero (0) is used to indicate the Desktop. (See the PointInto, PointAt, and (Un)HighlightMenu commands.)
>
> **flavor** specifies the type of an icon. The set of acceptable flavors corresponds to the different icon types (e.g., document, folder, recordfile) available in Star.
>
> An **icon#** is an integer between 1 and 6 inclusive, and indicates one of the "slots" in the Icon Area, reading from left to right.

A **menuItem** identifies an item in a window menu or an auxiliary menu. It is given by the writer as its string (e.g., Close or End Session). Since menu strings do not include commas, *menuItems* do not require quotation marks.

Each command name may be abbreviated by its "initials." Each command's abbreviation is shown following its argument list. The writer may use any combination of upper and lower case in the command names: NEWinfo, newInfo, NI, ni, etc. are all acceptable.

Display Commands

The first command is the means by which the writer gives information and instruction to the user.

NewInfo [text] NI

The contents of the text replace whatever was in the Introduction Window.

The commands in this next group provide two facilities to the writer. First, they allow him to illustrate the effects of user actions before asking the user to perform those actions himself. For example, a human trainer might say "Watch the screen while I move the mouse around with the Left button held down," to familiarize the user with the resulting visual effects. Second, these commands allow the writer to establish a set of Star objects and/or conditions such that the desired instruction can proceed. For example, if the writer wants the user to learn about selecting icons, he can cause an icon to appear on the screen and thereby to become available for future instruction.

PointInto [window#, location] PI

The location of the cursor on the screen is changed, and mouse-tracking is disabled (so that the user cannot control the cursor) until "pointer-control mode" terminates, as described below. The purpose of this command, as distinguished from PointAt, is to show the user where the cursor needs to be in order to select or invoke the specified object or command. Thus, the cursor should be located such that, if the SELECT button were pressed, the highlighting for a selection or command invocation would result. The location of the cursor is determined by the format of the *location* argument, as follows:

If the *location* argument is a **quoted string**, the cursor points at the first character in the first occurrence of that string within the text contents of the specified (document) window.

If the *location* is (the unquoted string name of) a **menuItem**, the cursor points into the specified menu item. If it occurs in an auxiliary menu, that menu is displayed until "pointer-control mode" terminates.

If the *location* consists of a **flavor-name** pair such as document-memos, the location of the cursor is determined by the *window#* argument: If the *window#* is zero, the cursor points into the named icon of the given flavor on the Desktop. If the *window#* is non-zero, the cursor points at the specified icon within that window. This non-zero case applies to an open container window, where the cursor would point to the named icon in that window.

The cursor points into the various regions of the scroll bar, as described in *System Overview*, for the following values of the *location* argument reading from top to bottom: **scrollDown, scrollPrevious, scrollJump, scrollNext, scrollUp.**

PointAt [window#, location] PA

The purpose of this command is either to draw the user's attention to the general area of an object of interest, or to allow the writer to move the cursor away from an object that he has recently Point(ed)Into. Thus, in each case the cursor should be located near, but still outside, the specified object, preferably in gray or white space where a button click would have no effect. The *location* is interpreted as follows:

menuItem: The cursor is near, but not inside, the menu item.

flavor-name pair: The cursor is near, but not inside, the icon.

any of the scroll locations: The cursor is close to the right side of the window, but not actually in the scroll area. Its vertical location is determined as in PointInto. [This assumes that the scroll area is always visible. If it is visible only when the cursor is in the area, then no "scroll" locations are available with PointAt.]

Pointer-control mode refers to the system state in which the user's mouse actions do not control the cursor, and/or in which an auxiliary menu continues to be displayed, as though the user were holding the SELECT button down. This mode ends when the Introduction executes any command that either requires a change in the mouse or cursor state, or requires that control return to the user. For example, the writer may use a PointAt or PointInto command, then alter the contents of the Instruction space, without affecting the cursor, but the execution of any monitoring command will return control of the mouse to the user.

PutUpIcon [flavor, name, real, icon#] PUI

The Introduction displays the icon of the specified flavor and name, in the specified location in the Icon Area. The value of the *real* argument is either genuine or fake, indicating whether or not the icon (document or folder) exists in the Introduction file drawer. If it exists, then it may be opened later in the Introduction and its contents will appear; if not, then the icon has no contents and it cannot be opened. Any Star icon type can be displayed with this command, but the value of *real* will be genuine only for those documents and folders that will actually be opened. [The purpose of the *real* argument is to have the implementation search the file drawer only if the object really exists.]

If the space designated by *icon#* is already occupied, the icon is displayed in any other open space in the Icon Area.

To reduce complexity, the writer may specify only one icon of a given name at any time. For example, the Desktop (and any container on the Desktop) may not contain both a folder and a document named "Memos."

OpenIcon [name] OI

The named icon opens into its window, displaying its contents. Only documents and folders can be opened into windows, and among those, only icons representing objects that exist in the Introduction file drawer can be opened. If the named icon is in a folder, then the folder must be open at the time OpenIcon is executed.

CloseWindow [window#] CW

The window returns to its iconic form. If it was opened from within a container window, that container window remains displayed.

CloseAllWindows [window#] CAW

The window, and all that contain it, are closed. The container window returns to iconic form.

DeleteIcon [name] DI

The specified icon is deleted from the Desktop.

SelectIcon [name] SI

The cursor points "into" the specified icon on the Desktop [such that the user can still see it, if possible]. The icon is selected and highlighted.

SelectText [window#, text] ST

The cursor points to the end of the first occurrence of the specified (quoted) text in the appropriate (document) window, and the text is selected and highlighted. The caret is displayed following the selection.
If the *text* argument is an "@", no selection is made, but the caret is displayed at the top of the window, preceding the first visible character.

InsertText [text] IT

The specified sequence of characters appears in the window following the caret, as though it had been typed by a user. Typically, this command would be used after the SelectText command.

HighlightMenu [window#, menuItem] HM

The cursor is placed so that it points "into" the specified command in the given window. The menu item is highlighted, as though the user had moved the cursor into it and pressed the SELECT button. The menu command is not executed. The Introduction Window menu is specified by a *window#* argument of 4; the Desktop transient menu by a *window#* of zero (0).

UnHighlightMenu [window#, menuItem] UHM

The menu command is un-highlighted, and the cursor is displayed below (but outside) its box. Again, commands in the Introduction Window and the Desktop transient menu are accessible.

ForceScroll [window#, distance] FS

The contents of the window are scrolled by the given distance in number of lines (smoothly, if implementable). A positive distance scrolls the window toward its end; a negative distance scrolls it toward its beginning. The cursor is displayed in the appropriate region of the scroll bar (i.e., in the upward region if the *distance* is positive, in the downward region if it is negative).
If the *distance* is of the form p<number>, then the window is scrolled by pages. The cursor and the number appear in the jumping region of the scroll area, and the window is scrolled to the corresponding page.

PutUpPropertySheet [psheet] PUPS

The property sheet whose string name corresponds to the *psheet* argument is displayed. It must be one of the following: Character, Paragraph, Tab, or Document (Status and Statistics, not Formatting), since those four are the only property sheets that can be monitored in the Introduction. The settings on the property sheet reflect the properties of the currently selected object.

TakeDownPropertySheet [] TDPS

The currently displayed property sheet disappears. The selection is updated if necessary to reflect the property sheet settings.

HighlightParameter [parameter, value] HP

[Note: Some details of psheet manipulations, both here and in UserEdit/MarkParameter, may change.]

The cursor is displayed pointing "into" the *parameter* (state or choice) on the property sheet currently displayed. The parameter is inverted as though it had been marked by the user. No change is made to the selection until TakeDownPropertySheet is executed.

The *parameter* is the identifying name that appears nearest the parameter of interest. On the Character property sheet, the valid *parameters* are Font, Size, Face, and Position. For Paragraph properties, they are Alignment, Hyphenation, Between Lines, Before Para., After Para., and Keep On. The Document property sheet has no Highlight-able parameters, since it can only appear with the Display choice of Status and Stat's turned on.

The *value* is the string that appears in the invertable box, except in the case of the Tabs property sheet. Thus some combinations for the Character property sheet might be:

HighlightParameter [Font, CENTURY]
HighlightParameter [Face, ITALICS]

For Paragraph properties:

HighlightParameter [Alignment, FLUSH LEFT]
HighlightParameter [Before Para., 1 1/2]

On the Tabs property sheet, the tab types are indicated by integers, counting left to right, top to bottom, including the "Spaces" text parameters. Since the value of the tab types are not strings, and are unambiguously specified by the integer, no *value* argument appears. Thus, if the Tabs property sheet were currently displayed,

HighlightParameter [3]

would indicate the first centered tab symbol.

UnHighlightParameter [parameter, value] UHP

The parameter is un-inverted and the cursor is displayed below (but outside) its box. The screen feedback of UnHighlightParameter simulates the user's action of moving the cursor out of the box before releasing the button; all the parameters that were affected by the preceding HighlightParameter command are returned to their original settings.

Note that the preceding two commands can cause parameters to become either black or white, depending on their settings when the property sheet was displayed. State and choice parameters work differently: If the writer Highlights a state parameter that was black when the property sheet came up, it turns white. If he Highlights a choice parameter that was black, it stays black. If he Highlights a choice parameter that was white, it turns black and all the other members of its set become white.

Monitoring Commands

The following groups of commands allow the writer to specify that the user is expected to perform a discrete, program-detectable action. The detailed flow of control within each command is not under the writer's control; the conditions under which feedback messages are displayed, and the criteria by which the user's action is evaluated, are built-in aspects of the Introduction facility. The specification of each command below describes these features.

Features Common to All Monitoring Commands

For each command, the Introduction's response to the correct user behavior is to display the RightMessage (see below) in the Message Area, then to execute the next command in the writer's file. If any error condition occurs, the Introduction displays the appropriate message, then waits for the user to try again. The Message Area is cleared whenever execution of a monitoring command begins, and whenever the user presses a key or a mouse button. (Thus any negative feedback disappears when the user begins an action.)

Throughout the Introduction, mouse-ahead and type-ahead (including repeated keys, usually generated by the user's holding the key down) are disabled.

The "Fall-Through" Feature. There will be occasions on which users, for whatever reasons, cannot correctly complete the action required by the command. To allow the user to escape from such a situation, each monitoring command is capable of modifying the state of the user's Desktop so that the action of the command is carried out. For example, if the user is required to move a text sequence from one document to another, and he cannot do it correctly, then the Introduction carries out the move itself. This "fall-through" occurs after the user makes three incorrect responses, and is accompanied by a default message. (Both the number of incorrect responses and the content of the message can be changed by the writer, as described below.)

Default Feedback Messages. For each error condition in each of the monitoring commands, the Introduction provides a default Feedback message. However, since the writer knows the instructional context in which the monitoring command will be carried out, he may choose to replace the default with a more informative message. In order to keep the syntax of the monitoring commands simple, it is necessary for the writer to establish the messages before specifying the monitoring to be performed. If no message is established, the default is displayed; once a message has been established, it remains as the content (for its particular error condition) until another "Message" command is executed. For any "Message" command, the writer may specify the argument default, which causes the system-supplied default value for the message content to be restored.

Each monitoring command has a corresponding "Message" command, with one argument for each of the error conditions. Each argument is a quoted text string; in the specification of these commands, only the identifiers of the individual messages are listed. A complete example is given under ClickButtonMessage, following the UserClickButton command, below. If a Message command has only one argument, the identifier and the colon following it need not appear; only the value of the argument (i.e., the quoted string) is significant.

Mouse-only actions. Some monitoring commands expect the user to carry out an action by using only the mouse, and therefore include an implicit error condition in which the user presses a key, causing the MouseOnlyMessage to be displayed. For any action that specifically requires the SELECT button (e.g., selecting a single character), the implicit error condition of pressing the ADJUST button elicits the SelectOnlyMessage. The incorrect keystroke or button click is otherwise ignored. These implicit error conditions are indicated in the commands below by the notations "(mouse-only)" and "(SELECT-only)" following the command name.

Timing out. Each monitoring command requires the user to perform some number of discrete actions, such as pressing and releasing a button, and/or pressing a key, etc. The user is allowed as much time as he likes to begin any action, but once he has pressed any key or button, he has one minute in which to proceed. Whenever one minute has elapsed following any key press or button up/down, the message area displays the TooMuchTimeMessage. Timing out is not considered to be an incorrect response; it simply allows the user to ignore his workstation as his needs require.

Common Message Commands. The following commands allow the writer to change the content of the "general purpose" messages. The default is shown.

RightMessage ["Good. You did the right thing."] RM

The specified text is established as the message that will appear in the Message Area following the next "correct" action by the user.

If the writer wishes to incorporate positive feedback as part of the next piece of information, he may set the RightMessage to nothing, by the command RightMessage [""], causing the Message Area to remain blank after the user's correct action.

WrongMessage ["No. Re-read the instructions and try again."] WM

The specified text will appear in the Message Area if the user performs some action that is neither correct nor covered by a specific error condition.

FallThroughNumber [3] FTN

This command sets the number of incorrect responses triggering the Introduction's "fall-through" action. Any integer greater than zero is acceptable.

FallThroughMessage ["You're having some trouble with this. Let's go on anyway. The correct action will be done for you."] FTM

MouseOnlyMessage ["No, don't use the keyboard. You need the mouse for this."] MOM

SelectOnlyMessage ["No, use the Left button."] SOM

TooMuchTimeMessage ["A very long time has passed. Please re-read the instructions and continue your work."] TMTM

Monitoring the Mouse

A major purpose of the Introduction is to make the user familiar and comfortable with the mouse. The value of the *button* argument is either L or R, for SELECT or ADJUST respectively.

UserClickButton [button] (mouse-only) UCB

The user is expected to click (press and release) the specified button. The cursor may be anywhere on the screen. The only error condition is handled as follows:

> **Wrong button goes down**: The message "No, that's the wrong mouse button." is displayed. Note that the message is displayed as soon as the wrong button is pressed. If it stays down for more than one second, the message " Release the button, then try again." is appended to the Message Area.

ClickButtonMessage [wrong, release] CBM

The writer may replace the content of either feedback message that can be displayed by the UserClick command. The messages are identified by the names *wrong* and *release*, which allows the writer to replace only one message if he chooses. The writer must use the identifier, followed by a colon, before each of the messages whose content he wishes to replace. For example, the writer might use any one of these three commands:

> ClickButtonMessage [wrong: "No, you pressed the Right button instead of the Left."] (The *release* content would remain unchanged.)
>
> ClickButtonMessage [release: " Please release the Right button, then press and release the Left button."] (The *wrong* content would remain unchanged.)
>
> ClickButtonMessage [wrong: "No, you pressed the Right button instead of the Left.", release: " Please release the Right button, then press and release the Left button."] (Both feedback messages would be changed.)

Once either default message has been changed, the writer may return to the system-supplied default with any of these commands:

> ClickButtonMessage [wrong: default] (The *release* content would remain as it was.)
>
> ClickButtonMessage [release: default] (The *wrong* content would remain as it was.)
>
> ClickButtonMessage [wrong: default, release: default] (Both feedback messages would revert to the system-supplied default.)

Finally, the various identifiers and arguments may be used in combination, as in:

> ClickButtonMessage [wrong: "You pressed the button on the right. You should have pressed the button on the left side of the mouse.", release: default]

Monitoring Text Actions

The commands in this group allow the writer to give the user practice in making text selections and scrolling.

UserSelectCharacter [window#, character] (mouse-only) (SELECT-only) USC

The user is expected to select any instance of the character specified by the quoted *character* argument (required to be a single character). The Introduction evaluates the selection immediately after the user has pressed and released the SELECT button. The error conditions are handled as follows:

> **Mouse click did not produce any selection**: The message "No, you haven't selected anything. Try again." is displayed. If the mouse action would have invoked a command, it is not executed.
>
> **Selection is in another window**: The message "No, that's the wrong window. Try again." is displayed. The incorrect selection remains.
>
> **Some object other than a character was selected**: The message "No, you didn't select a character. Try again." is displayed. The incorrect selection remains.
>
> **A character was selected, but selected character is wrong**: The message "No, you haven't selected the correct character. Try again." is displayed. The incorrect selection remains.

SelectCharMessage [noSelection, wrongWindow, notChar, wrong] SCM

UserSelectText [window#, text] (mouse-only) UST

The Introduction waits for the user to invoke the Continue menu item, then evaluates the selection. The *text* may consist of one or more characters; this command is thus more general and more lenient than UserSelectCharacter. The Feedback messages are handled identically, except that the "notText" message (corresponding to "notChar") says "No, you didn't select any text. Try again.", and the "wrong" message says "No, you haven't selected the correct text.".

UserSelectText is used whenever the writer wants the user to make a selection with his own discretion, and/or a selection that requires the "select-adjust" mechanism.

SelectTextMessage [noSelection, wrongWindow, notText, wrong] STM

UserDoubleClick [window#, text] (mouse-only) (SELECT-only) UDC

This command evaluates the user's action twice. The first button-click must result in the selection of a character that is contained in the target word, and the second click must be made when the cursor is still within that character, resulting in the correct word selection. Error conditions are handled as follows:

First click does not result in any character selection: The message "No, begin by selecting a character that is in the correct word." is displayed. The incorrect selection, if any, remains. If the mouse action would have invoked a command, it is not executed.

First click selects a character not within the word: The message "No, that character is not in the correct word. Try again." is displayed. The incorrect button click produces a character selection, which remains highlighted. The user's next click is interpreted as a first click. If it points at the same incorrect character, the same message is displayed, but the click does not produce a word selection.

First click is correct, but second click is outside the character: The message "No, the pointer was away from the selected character when you released the button. Start all over again." is displayed. The incorrect selection remains, and the user begins again with his first click.

Note: The first selection must be a character that actually is in the target word, not simply an instance of a character that occurs in the word. For example, if the correct word is "amazing " and the user selects an "a" in the word "grace," the Introduction recognizes his action as an error on the first click.

DoubleClickMessage [notChar, firstWrong, secondWrong] DCM

UserTripleClick [window#, text] (mouse-only) (SELECT- only) UTC

This command is exactly like UserDoubleClick, except that the first click must be within the target sentence, the default "notChar" and "firstWrong" messages say "sentence" instead of "word," and another error condition exists:

Third click is outside the word: The message "No, the pointer was away from the selected word when you released the button. Start all over again." is displayed, and the incorrect selection remains. The user's next click is interpreted as a first click. If it is made at the same character, a character (not word) selection results; that selection may or may not be in the target sentence.

TripleClickMessage [notChar, firstWrong, secondWrong, thirdWrong] TCM

UserQuadrupleClick [window#, text] (mouse-only) (SELECT- only) UQC

This command is exactly like UserTripleClick, except that the first click must be within the target paragraph, the default "notChar" and "firstWrong" messages say "paragraph" instead of "sentence," and another error condition exists:

Fourth click is outside the sentence: The message "No, the pointer was away from the selected sentence when you released the button. Start all over again." is displayed, and the resulting incorrect selection remains. The user's next click is interpreted as a first click. If it is made at the same character, a character (not word) selection results; that selection may or may not be in the target paragraph.

QuadrupleClickMessage [notChar, firstWrong, secondWrong, thirdWrong, fourthWrong] QCM

UserScrollWindow [window#, direction] (mouse-only) USW

The user is expected to scroll the designated window in the correct direction, where the *direction* argument is one of the following: down, previous, next, or up. These identifiers correspond to the regions of the scroll area, from top to bottom (except for the jumping region). The user's action is correct if he activates the appropriate region of the scroll area for any period of time; it is not possible to evaluate scrolling (other than to the previous or next page) by specified distances.

Error conditions are handled as follows, with the window restored to its pre-scrolled state in each case:

User tried to scroll in wrong window (including the Introduction Window): The message "No, that's the wrong window. Try again." is displayed.

A button-up mouse event occurred, but no scrolling resulted: The message "You need to move the pointer close to the right edge of the window. Try again." is displayed. If the scroll area is always visible, this message says "into the scroll region at the right edge ..." instead of "close to the right edge ..."

User scrolled in wrong region: The message "No, you used the wrong scroll region. Start again." is displayed.

***Direction* was previous or next, but either button-down or button-up occurred elsewhere**: The message "No, when you turn pages, keep the pointer inside the labeled box and click the button. Try again." is displayed.

ScrollWindowMessage [wrongWindow, noScroll, wrongRegion, down/upOutside] SWM

UserTypeText [window#, caretLocation, text] UTT

The user is expected to type the specified text into the window. The system examines the location of the caret as soon as the first character is typed. The correct location of the caret is determined by the location of an @ character within the *caretLocation* argument, which is a quoted string. The user's initial action of positioning the caret is correct if the caret appears in the same relative position as the writer's @ within the quoted string. For example, a *caretLocation* of "@The user" would indicate that the user must position the caret before the first word of this paragraph; "of characters.@" would indicate a caret following this paragraph; "ex@pected" indicates that the caret should be between the x and the p. The user may position the caret by any means he chooses, including (but not limited to) selecting a sequence of characters.

Any mouse or keystroke actions within the span of typed-in characters following the correctly-placed caret are carried out; mouse actions outside the text contents of the window (in the menu, for example) cause the Introduction to complain.

When the user invokes the Continue menu item in the Introduction Window (the only allowable mouse action outside the specified interaction window), the Introduction examines the contents of the window to determine whether or not he inserted the correct text following the caret.

Error conditions (including those described above) are handled as follows:

> **Caret missing or in wrong place when first character is typed**: The message "Please place the caret correctly before you begin typing." is displayed, and the typed character is otherwise ignored.
>
> **Mouse action outside the text contents of the window**: The message "No, you don't need to do anything outside the window." is displayed, and the mouse action is ignored. Any text correctly typed so far remains in the window.
>
> **After user invokes Continue, first character of his type-in is incorrect**: The message "The first thing you typed was wrong. Invoke Continue when you're ready to start again." is displayed. The Introduction waits for the user to invoke Continue (ignoring any other mouse action or keypress), then his type-in is cleared, leaving the correct caret (and selection, if any).
>
> **After user invokes Continue, something after first character is incorrect**: The message "Something that you typed was wrong. Invoke Continue to try again, starting from the place where you first went wrong." is displayed. The Introduction waits for the user to invoke Continue (ignoring any other mouse action or keypress), then the characters beginning with the first incorrect character are cleared. The caret appears following the last character that was correctly typed.
>
> **User's type-in begins correctly, but he attempts to modify other parts of the window's contents**: The message "No, you don't need to change anything there. Check your typing, and invoke Continue when you're finished." is displayed. The attempted modification is not carried out. The selection (if any) is killed, the caret is displayed following the last character that was typed in the correct place, and the Introduction continues to wait for type-in or mouse action within the span of typed-in characters.

TypeTextMessage [wrongCaret, outsideWindow, wrongFirst, wrongLater, extraChanges] TTM

UserTypeAnything [window#, caretLocation] UTA

The user is expected to type a sequence of characters that will not be evaluated by the Introduction. The cursor and the caret are displayed at the *caretLocation* as determined in UserTypeText. (The location is always specified by the writer, not left to the user.) Until the user invokes Continue, he is free to type, and his characters are added to the document. Function keys (DELETE, FIND, etc.) are ignored. Text-related functions such as backspace and backword operate normally. No mouse actions other than Continue are allowed. The only error condition is:

Mouse action outside the Continue menu item: The message "No, you don't need the mouse. Just invoke Continue when you're ready to go on." is displayed, and the mouse action is otherwise ignored.

TypeAnythingMessage [mouseAction] TAM

UserDeleteText [window#, selection] UDT

The user is expected to delete the *selection* (a quoted text argument) from the window. The system examines the user's selection when the DELETE key is pressed, complaining if it does not match the *selection*; if it is correct, it is deleted.

Error conditions are handled as follows:

Mouse click outside contents of any window: The message "No, use the mouse to make the correct text selection, then press the <DELETE> key." is displayed. The mouse click is otherwise ignored; e.g., no menu command is carried out.

User presses a key other than DELETE: The message "No, use the <DELETE> key." is displayed, and the keystroke is otherwise ignored.

Non-existent or wrong selection when DELETE **is pressed**: The message "No, you don't have the correct selection. Select correctly, then press the <DELETE> key." is displayed.

DeleteTextMessage [outside, notDelete, wrongSelection] DTM

UserMoveText [window#, selection, destination] UMT

The user is expected to move the *selection* to follow the *destination* (which are both quoted text arguments). The system examines the user's selection when the MOVE key is pressed, complaining if it does not match the *selection* argument. After the user completes the move by clicking the button at the new location, the system checks the result. The user's action is correct if he moved the selected text to follow the last character of the *destination*.

Error conditions are handled as follows:

Mouse click outside contents of any window: The message "No, use the mouse to make the correct text selection, then press the <MOVE> key." is displayed. The mouse click is otherwise ignored.

User presses a key other than MOVE: The message "No, use the <MOVE> key." is displayed, and the keystroke is otherwise ignored.

Non-existent or wrong selection when MOVE is pressed: The message "No, you don't have the correct selection. Select correctly, then press the <MOVE> key." is displayed.

Any keystroke after move is in progress: The message "No, you don't need to type anything." is displayed, and the keystroke is otherwise ignored.

User attempts to terminate the move in an illegal location: The message "No, the system can't leave the text there. Try again, starting with the <MOVE> key." is displayed. The selected text returns to its original location and remains selected.

User terminates the move in an incorrect location: The message "No, you moved the text to the wrong place. Invoke Continue when you're ready to try again, then press the <MOVE> key." is displayed. The Introduction waits for the user to invoke Continue (ignoring any other mouse action or keypress), then replaces the text at its original location and leaves it selected. [Implementation may require that an incorrect move be aborted at button-up, instead of carried out and then undone.]

MoveTextMessage [outside, notMove, wrongSelection, keystroke, illegal, incorrect] MTM

UserCopyText [window#, selection, destination] UCT

This command is identical to UserMoveText except that the user is expected to copy rather than move the sequence of text. "<COPY>" replaces all occurrences of "<MOVE>", and in the case of an illegal or incorrect destination, the copied text simply disappears after the user invokes Continue, leaving the correct selection highlighted.

CopyTextMessage [outside, notCopy, wrongSelection, keystroke, illegal, incorrect] CTM

Monitoring Menu Actions

UserMarkMenu [window#, menuItem] (mouse-only) UMM

The user is expected to invoke the specified menu command, and it is then executed. If the user invokes a menu item other than the correct one, it does not take effect. A *window#* argument of 4 specifies the Introduction Window.

The window menu commands that can be monitored and executed in the Introduction are Close, Close All, Show Next, Show Previous, and all of the commands in the Introduction Window. The Set Window, Redisplay, and "?" menu commands are inoperable, and cannot be required of the user. Error conditions are handled as follows:

Mouse click outside any menu: The message "No, move the pointer into the <1> menu command, then press and release the Left button." is displayed. The <1> variable is replaced by the *menuItem* argument.

Wrong window's menu: The message "No, that menu is not in the right window. Try again." is displayed.

Wrong menu item: The message "No, you marked the wrong menu item. You need the <1> command. Try again." is displayed.

MarkMenuMessage [noMenu, wrongWindow, wrongItem] MMM

UserEndSession [] (mouse-only) UES
The user is expected to invoke the End Session command in the Desktop transient menu, and it is then executed. Error conditions are handled as follows:

Mouse click outside transient menu symbol: The message "No, move the pointer to the box in the upper right corner, then hold down the Left button to get the menu." is displayed. The mouse action is otherwise ignored.

Wrong menu item in the transient menu: The message "No, the pointer was away from the End Session command when you released the button. Try again." is displayed. The incorrect command is not executed. The user must begin again, holding down the button to display the menu.

EndSessionMessage [outsideMenu, wrongItem] ESM

UserMarkPsheetMenu [menuItem] (mouse-only) (SELECT-only) UMPM

This command is identical to UserMarkMenu except that it applies to the property or option sheet currently displayed. The error conditions are handled identically, except that the "wrongWindow" message is "No, the pointer should be in the property sheet menu. Try again." [The writer must change the content of the message when this command is used to monitor option sheet actions like UserMoveIcon to a printer.]

MarkPsheetMenuMessage [noMenu, wrongWindow, wrongItem] MPMM

Monitoring Icon Actions

UserSelectIcon [name] (mouse-only) (SELECT-only) USI

The user is expected to select the specified icon. His actions are evaluated immediately after he presses and releases the SELECT button.

Error conditions are handled as follows:

Mouse click outside any icon: The message "No, move the pointer into the icon named <1>, then press and release the Left button." is displayed. The variable <1> is replaced with the *name* argument. If the mouse click resulted in a selection, it remains highlighted; however, if the mouse click would have invoked a menu command, it is not executed.

Wrong icon selected: The message "No, you didn't select the icon named <1>. Try again." is displayed. The incorrect selection remains.

SelectIconMessage [outside, wrongIcon] SIM

UserOpenIcon [name] UOI

The user is expected to press the OPEN key when the correct icon is selected. Only documents and folders may be opened.

Error conditions are handled as follows:

Mouse click outside any icon: The message "No, use the mouse to select the icon named <1>, then press the <OPEN> key." is displayed. The mouse action is otherwise ignored.

Any key other than OPEN pressed: The message "No, you need the <OPEN> key. Try again." is displayed.

Non-existent or wrong selection when OPEN is pressed: The message "No, select the icon named <1>, then press the <OPEN> key." is displayed. The keystroke is otherwise ignored.

OpenIconMessage [outside, notOpen, wrongIcon] OIM

UserDeleteIcon [name] UDI

The user is expected to press the DELETE key when the correct icon is selected.

Error conditions are handled as follows:

Mouse click outside any icon: The message "No, use the mouse to select the icon named <1>, then press the <DELETE> key." is displayed. The mouse action is otherwise ignored.

Any key other than DELETE pressed: The message "No, you need the <DELETE> key. Try again." is displayed.

Non-existent or wrong selection when DELETE is pressed: The message "No, select the icon named <1>, then press the <DELETE> key." is displayed. The keystroke is otherwise ignored.

DeleteIconMessage [outside, notDelete, wrongIcon] DIM

UserMoveIcon [name, destination] UMI

The named icon may be either on the Desktop or in an open folder window. The *destination* may be either the word desktop or one of these icon flavors: printer, folder, or filedrawer.

If the *destination* is a folder, filedrawer, or printer, then the *name* argument must name a document or a folder, since these three destinations will behave as though they had received the moved icon. Upon successful completion of the move, the folder will contain the target document. The filedrawer will show the normal "receiving" highlighting, but no filing will actually take place.

When an icon is correctly moved to the printer, the printing option sheet appears, and the UserMoveIcon command has completed its execution. The writer must then call UserMarkPsheetMenu [Start] so that the user is required to make the option sheet disappear.

After three seconds, an icon that was correctly moved to a printer is returned to an empty location on the Desktop. [Placeholder: The actual printing of a document will be described here. It will appear to the user exactly as though "normal" Star printing were in operation, which may or may not actually be the case.]

If the *destination* is a folder, then it may be either a window or an icon, depending on the state of the user's Desktop at the time. The writer may have only one icon or window of the specified *destination* flavor on the Desktop when this command is executed, so that the Introduction can unambiguously determine the correct destination object.

If the *destination* is desktop, the user must move the icon to an empty "slot" within the Icon Area, different from the icon's original location. If the named icon's original location was inside a folder window, then a correct UserMoveIcon to the desktop removes the icon from the folder.
The user is successful if he selects the target icon, presses the MOVE key, and clicks either mouse button at the specified destination.

Error conditions are handled as follows:

Mouse click outside any icon: The message "No, use the mouse to select the icon named <1>, then press the <MOVE> key." is displayed. The mouse action is otherwise ignored.

User presses a key other than MOVE: The message "No, use the <MOVE> key." is displayed, and the keystroke is otherwise ignored.

Non-existent or wrong selection when MOVE is pressed: The message "No, select the icon named <1>, then press the <MOVE> key." is displayed. The incorrect selection remains selected.

Any keystroke after move is in progress: The message "No, you don't need to type anything. Click the button to finish the move." is displayed, and the keystroke is otherwise ignored.

User attempts to terminate the move in an illegal location (such as inside a non-container window, or over a non-container icon): The message "No, the system can't leave the icon there. Try again, starting with the <MOVE> key." is displayed. The selected icon is replaced at its original location, and remains selected.

User attempts to terminate the move in an incorrect location (i.e., over a "receiver" icon, or outside the Icon Area, when the *destination* is desktop; or in any legal location that is not the correct "receiver" icon or window when the *destination* is folder, filedrawer, or printer): The message "No, you should have moved the icon to the <1>. Try again, starting with the <MOVE> key." is displayed. The variable <1> is replaced with the word desktop, folder, filedrawer, or printer, as appropriate. The selected icon is replaced at its original location, and remains selected.

MoveIconMessage [outside, notMove, wrongIcon, keystroke, illegal, incorrect] MIM

UserCopyIcon [name, destination] UCI

This command is identical to UserMoveIcon, except as follows: If the user selects the correct icon and presses the COPY key, but then attempts to leave the copied icon in an illegal or incorrect location, the copy disappears and the original icon remains selected. The user must again press COPY and place the copied icon at the correct destination. The content of the default error messages is the same, except that the word "<COPY>" replaces the word "<MOVE>."

CopyIconMessage [outside, notCopy, wrongIcon, keystroke, illegal, incorrect] CIM

Monitoring Property Sheet Actions

The first two of these commands (UMP and UEP) can be executed only when a property sheet has already been opened. Both expect the user to change only one parameter. (A sequence of property sheet actions can be carried out by a sequence of separate commands; the property sheet is not removed until its "Done" has been invoked by means of the UserMarkPsheetMenu command.) The third command (UCP) requires the user to select an object, press the PROPS key, then modify the property sheet as instructed.

The *parameter* and *value* arguments are identical to those described under HighlightParameter command, with the addition of a Display parameter for all except the Document property sheet.

All of the parameters on the four available property sheets can be monitored by the Introduction, with the following exceptions: (1) on the Character, Paragraph, and Tab property sheets, the only operable "Display" options are CHAR, PARA, and TABS; (2) the OTHER choices are not operable in the Position or Space parameters; (3) the Name property of Characters is not operable; (4) on the Document property sheet, only the Name parameter can be monitored.

UserMarkParameter [parameter, value] (mouse-only) (SELECT-only) UMP

The user is expected to mark the appropriate *value* of the given *parameter* in the property sheet currently displayed. (For the Tabs property sheet, only a single argument appears, as explained under HighlightParameter.) This command applies to state and choice parameters only, and the user's action is evaluated as soon as he has pressed and released the SELECT button. The selection is changed to reflect his actions only if he is correct, and only if the Character property sheet or a Display choice was involved.

Error conditions are handled as follows:

User clicks outside property sheet: The message "No, the pointer should be inside the property sheet. Try again." is displayed. The button click is otherwise ignored.

User clicks in property sheet menu: The message "No, you don't need the menu until you've marked the <1> parameter. Try again." is displayed. The variable <1> is filled in from the *parameter* argument. The menu command is not executed.

User clicks in wrong parameter: The message "No, that's the wrong parameter. You need the <1> parameter." is displayed. The <1> is replaced by the *parameter-value* pair. The incorrect parameter is reset to its original value.

MarkParameterMessage [outside, menu, wrong] MPM

UserEditParameter [parameter, value] UEP

This command applies to text parameters. For the Paragraph property sheet, the valid *parameter*s are (for the margins) Left and Right. For Tabs, the integers 1, 6, 11, 16, etc. specify the "Spaces" parameters. For Document properties, only Name is valid.

The user is expected to edit the contents of the *parameter* so that they become identical with the (unquoted) *value*. This may involve making a selection in the text already there, or placing the caret at the left side of the empty parameter and typing in the specified text. [Although a Star document name may contain commas, the writer may not specify a name containing commas in the Introduction. Thus the *value* argument is unquoted, and contains either an integer (for the Paragraph Left-Right or Tabs "Spaces" values) or a sequence of characters (for the Document Name parameter.]

The contents of the parameter are evaluated when the user invokes the Continue menu item in the Introduction Window (but the user's actions are reflected in the selection only after he invokes Done, since this command does not apply to the Character property sheet).

The first three error conditions are the same as in UserMarkParameter, and one more condition is added:

Value of parameter does not match *value*: The message "No, your typing was not exactly right. Try again." is displayed. The selection and caret appear as the user placed them, so that only the typing must be redone.

EditParameterMessage [outside, menu, wrong, noMatch] EPM

UserChangeProps [selection, window#, parameter-value, parameter-value, . . .] UCP

The user is expected either to have already, or to make, the correct text or icon selection, then press PROPS to display the appropriate property sheet. If the *selection* argument is quoted, then it specifies a sequence of characters. Otherwise, it is interpreted as the name of a document icon. If the *selection* specifies text, the *window#* identifies the interaction window; if the *selection* is an icon, the *window#* argument is ignored, but may not be omitted. The other arguments specify any number of state and/or choice parameters, as a list of *parameter-value* pairs separated by commas. Some valid pairs are Face-BOLD, Alignment-JUSTIFIED, Display-PARA, 5 (i.e., the first decimal-aligned tab).

If no *parameter-value* pairs are specified, then the purpose of the command is simply to display the correct property sheet, and the user's action is evaluated when he presses the PROPS key. (For example, this is the mechanism by which the writer would have the user display the Document property sheet, which has no state or choice parameters that can be changed.)

If a list of *parameter-value* pairs is specified, they are checked when the user invokes the Continue menu item in the Introduction Window; his actions are correct if each parameter-value listed has changed from its original value, while the others remain unchanged. In the case of the Character properties and the Display parameters, the resulting changes to the selection (or property sheet displayed) are made only after the user's actions are all correct. Other changes are only made after Done is invoked, as in "normal" Star.

The order in which the user clicks at the parameters is ignored, and any mouse action within the property sheet (not including in the property sheet menu) is carried out, so that the highlighting of the parameters accurately reflects the user's actions; no changes are made to the selection until the user's actions are correct.

Error conditions are handled as follows:

Mouse click outside contents of any window: The message "No, use the mouse to make the correct text selection, then press the <PROPS> key." is displayed. The mouse click is otherwise ignored.

User presses a key other than PROPS: The message "No, use the <PROPS> key." is displayed, and the keystroke is otherwise ignored.

Non-existent or wrong selection when PROPS is pressed: The message "No, you don't have the correct selection. Select correctly, then press the <PROPS> key." is displayed. The incorrect selection remains.

Any keystroke after property sheet displayed: The message "No, you don't need to type anything." is displayed. The keystroke is otherwise ignored; thus it is impossible for the user to change a text parameter during the execution of this command.

User clicks outside property sheet (other than Continue in the Introduction Window): The message "No, you don't need to do anything outside the property sheet until you're ready to invoke the Continue command." is displayed. The button click is otherwise ignored.

State or choice parameter values incorrect: The message "No, the following parameter(s) are not right: <1>, <1>, . . . Change them, then invoke Continue." is displayed. The parameters whose values are wrong are listed by their string names so that the user can locate and change them. [On the Tabs property sheet, this information will be meaningless to the user. The writer should use the ChangePropsMessage to display something more useful.] All parameters, both correct and incorrect, remain as the user set them.

ChangePropsMessage [outsideWindow, notProps, wrongSelection, keystroke, outsidePsheet, wrong] CPM

Monitoring the Carriage

The first of these commands is the means by which the Carriage is displayed on the screen; the others control the user's settings of the various parts of the Carriage, and its menu. Except for UserGetCarriage, all of the commands require that the Carriage be displayed on the screen before the command is executed.

UserGetCarriage [selection, window#] UGC

When the MARGINS key is pressed, the Introduction checks the user's selection, which must correspond to the *selection* argument in the appropriate window. The Carriage is then displayed. Error conditions are handled as follows:

Mouse click outside contents of any window: The message "No, use the mouse to make the correct text selection, then press the <MARGINS> key." is displayed. The mouse click is otherwise ignored.

User presses a key other than MARGINS: The message "No, use the <MARGINS> key." is displayed, and the keystroke is otherwise ignored.

Non-existent or wrong selection when MARGINS is pressed: The message "No, you don't have the correct selection. Select correctly, then press the <MARGINS> key." is displayed. The incorrect selection remains.

GetCarriageMessage [outside, notMargins, wrongSelection] GCM

UserSetMargin [side, location] USM

The user is expected to select the *side* (left or right) margin marker in the Carriage, then press the MOVE key, then click the button at the *location* on the Carriage scale.

Error conditions are handled as follows:

Mouse click outside the Carriage before any key is pressed: The message "No, you must select the <1> margin marker inside the Carriage." is displayed. The variable <1> is replaced by the *side* argument. The mouse click has no other effect; i.e., the existing text selection is not changed, and no menu command is invoked.

User presses a key other than MOVE: The message "No, use the <MOVE> key." is displayed, and the keystroke is otherwise ignored.

Correct margin box not selected when MOVE is pressed: The message "No, select the small box on the <1> side of the Carriage, then press the <MOVE> key." is displayed (with *side* replacing the <1>).

Mouse click outside the Carriage after MOVE was (correctly) pressed: The message "No, you need to click the button when the pointer is above <1> on the Carriage scale." is displayed. The variable <1> is replaced by the *location* argument. The mouse action is otherwise ignored.

Any keystroke after move is in progress: The message "No, you don't need to type anything now." is displayed, and the keystroke is otherwise ignored.

User moves the marker to the wrong place: The message "No, the marker should be above <1> on the scale. Try again, starting with the <MOVE> key." is displayed (with *location* replacing the <1>). The margin marker remains selected and in its incorrect location.

SetMarginMessage [outside, notMove, wrongMargin, extraClick, keystroke, wrongPlace] SMM

UserEraseTab [location] UET

The user is expected to select the tab marker in the specified location on the Carriage, then hit the DELETE key. Errors are handled as follows:

Mouse click outside the Carriage before any key is pressed: The message "No, you must select the tab marker above <1> on the Carriage." is displayed (with <1> replaced by *location*). The mouse click has no other effect.

User presses a key other than DELETE: The message "No, use the <DELETE> key." is displayed, and the keystroke is otherwise ignored.

Correct tab not selected when DELETE is pressed: The message "No, select the tab marker above <1> on the Carriage, then press the <DELETE> key." is displayed (with <1> replaced by *location*).

EraseTabMessage [outside, notDelete, wrongTab] ETM

UserRelocateTab [from, to] URT

The user is expected to select the tab marker appearing at the *from* location, and move it to the *to* location.

Errors are handled as follows:

Mouse click outside the Carriage before any key is pressed: The message "No, you must select the tab marker above <1> on the Carriage." is displayed (with <1> replaced by *from*). The mouse click has no other effect.

User presses a key other than MOVE: The message "No, use the <MOVE> key." is displayed, and the keystroke is otherwise ignored.

Correct tab not selected when MOVE is pressed: The message "No, select the tab marker above <1> on the Carriage, then press the <MOVE> key." is displayed (with <1> replaced by *from*).

Mouse click outside the Carriage after MOVE was (correctly) pressed: The message "No, you need to click the button when the pointer is above <1> on the Carriage scale." is displayed (with <1> replaced by *to*), and the mouse action is otherwise ignored.

Any keystroke after move is in progress: The message "No, you don't need to type anything now." is displayed, and the keystroke is otherwise ignored.

User moves the marker to the wrong place: The message "No, the tab should be above <1> on the scale. Try again, starting with the <MOVE> key." is displayed (with <1> replaced by *to*), and the tab marker remains selected and in its incorrect location.

RelocateTabMessage [outside, notMove, wrongTab, extraClick, keystroke, wrongPlace] RTM

UserPlaceTab [type, location] UPT

The value of *type* is either left, centered, decimal, or right. The user is expected to select the marker of the specified type in the Carriage source, hit COPY, then click at the *location* position in the Carriage scale. [Note that a tab could also be set by copying an existing tab from above the units scale on the Carriage, rather than copying from the source. This command does not expect or allow the user to set a tab this way.] Errors are handled as follows:

Mouse click outside the Carriage before any key is pressed: The message "No, you must select the correct tab type from the top line of the Carriage." is displayed. The mouse click has no other effect.

User presses a key other than COPY: The message "No, use the <COPY> key." is displayed, and the keystroke is otherwise ignored.

Correct tab type not selected when COPY is pressed: The message "No, select the correct tab type from the top line of the Carriage, then press the <COPY> key." is displayed.

Mouse click outside the Carriage after COPY was (correctly) pressed: The message "No, you need to click the button when the pointer is above <1> on the Carriage scale." is displayed (with <1> replaced by *location*), and the mouse action is otherwise ignored.

Any keystroke after copy is in progress: The message "No, you don't need to type anything now." is displayed, and the keystroke is otherwise ignored.

User copies the marker to the wrong place: The message "No, the marker should be above <1> on the scale. Try again, starting with the <COPY> key." is displayed (with <1> replaced by *location*). The user's incorrectly placed copy disappears, and the tab marker of the correct type in the Carriage source remains selected.

PlaceTabMessage [outside, notCopy, wrongTab, extraClick, keystroke, wrongPlace] PTM

UserMarkCarriageMenu [menuItem] (mouse-only) (SELECT-only) UMCM

This command is identical to UserMarkMenu except that it applies to the Carriage, which must be displayed at the time the command is executed. The allowable menu items are Done, Reset, and Defaults. (The Units command cannot be monitored.) Error conditions are handled as in UserMarkMenu, except that the "wrongWindow" message is "No, the pointer should be in the Carriage menu. Try again."

MarkCarriageMenuMessage [noMenu, wrongWindow, wrongItem] MCMM

Miscellaneous Monitoring

UserPressKey [name] UPK

The user is expected to press the key whose keytop says *name*. [The figure on p. 13 of FS Version 5.2 is the authoritative source; the *name* comes from the bottom of the key, e.g., the slash rather than the question mark.] The writer may specify a combination of keys, such as keyboard-bold, separating the two names with a hyphen. The effect of the keystroke (e.g., inserting a special character) is not carried out by this command; its purpose is to provide early instruction that helps the user feel comfortable with the keyboard. If the writer wants the user to see the effect of the keystroke, he must use the appropriate monitoring command (e.g., UserOpenIcon), to insure that other conditions, notably the selection, are appropriate.

Another use of the command, however, is to control the execution of the Introduction. If the writer prefers to have the user signal his readiness to continue by means of the keyboard (e.g., the NEXT key) instead of the Introduction Window menu, he may use a command like UserPressKey [next] with appropriate instructions.

Error conditions are handled as follows:

User presses either mouse button: The message "No, you don't need the mouse for this. Press the <1> key on the keyboard." is displayed (with <1> replaced by *name*), and the button click is otherwise ignored.

User presses a different key: The message "No, that's the wrong key. Press the <1> key." is displayed. The keystroke is otherwise ignored.

PressKeyMessage [mouse, wrongKey] PKM

UserContinue [] (mouse-only) (SELECT-only) UC

This command allows the user to invoke any of the menu items in the Introduction Window, or the End Session command in the Desktop menu. Normally, the user will invoke Continue in order to see the next piece of information, but he may also choose to repeat a section of instruction or log out. (See "Progressing Through the Introduction", below.) The only error condition is:

User clicks outside Introduction Window menu or Desktop transient menu: The message "No, please invoke Continue when you're ready for more information." is displayed. The button click is otherwise ignored.

ContinueMessage [] CM

UserFreeToContinue [] UFTC

This command allows the user to perform any sequence of Star actions he chooses. The Introduction does not monitor his actions in any way. Execution from the writer's file continues after the user invokes the Continue command in the Introduction Window. (The other menu commands disappear until Continue is invoked.) The purpose of this loophole is to allow the writer to present information describing a Star function that cannot be monitored, then allow the user to practice with it as he chooses. It is only appropriate in the later stages of the Introduction.

When the user invokes Continue, the Desktop is cleared of all icons and open windows, to insure that nothing exists to conflict with the writer's subsequent commands. Any object overlapping the Introduction Window (e.g., a property sheet) is also cleared. The constraints associated with the Introduction Desktop (e.g., availability of only specified property sheets, operability of only specified window menu items, etc.) are again in force.

Control Commands

In order to allow for some variation in users' ability to acquire information and desire for reinforcement, two levels of structure are available. These two levels are called "Unit" and "Practice," and allow the user a degree of control over the sequence of instruction. In addition, if the user ends his session before completing the Introduction, the Unit marker serves as a restart point from which instruction will begin the next time he signs on.

Unit [label] U

This command is mainly a marker in the writer's file. If the user ends his session before completing the Introduction, the last Unit marker that he passed will be the point at which instruction begins when he next signs on. Because a user may begin a Unit from a "blank" Desktop (since he may just have signed on after ending a previous session), the beginning of each Unit should contain whatever commands are necessary to establish the environment in which the user is expected to work -- appropriate icons on the Desktop, etc.

The *label* argument is ignored by the Introduction, but may be used by the writer as a comment.

EndUnit [label] EU

This command marks the end of a Unit. When it is executed, all icons, open windows, and objects covering the Introduction Window are cleared, so that subsequent instruction proceeds from a clean slate.

Practice [label] P

A Practice block is used to allow the user more information and/or more work on a particular topic. Commands within a Practice block are not special in any system-significant way.

The writer may include as many Practice blocks inside a Unit as he chooses; the beginning of each is indicated by this command. The end of the current Practice block is indicated either by another Practice label or by an EndUnit label. Non-unique *label*s in Practice commands may produce unexpected results

ChoosePractice [label, userAnswer, label, userAnswer, . . .] CP

This command allows the user to enter a Practice block at his option. The argument is a series of pairs, each of which consists of a *label* and a *userAnswer.* Each *label* identifies the Practice block from which instruction continues, if the user's typed-in response matches the corresponding *userAnswer* argument. If the user types anything other than one of the *userAnswer*s, then instruction continues from the command following this current ChoosePractice command. The user's response appears on the next blank line in the Introduction Window, and is terminated by NEXT.

If nothing follows the first *label* argument, then the labeled Practice block is entered unconditionally.

Practice blocks are only entered directly from ChoosePractice commands, and must be located at the end of a Unit. When the Introduction encounters a Practice label in the course of execution, one of two things happens: (1) If the user is currently completing a preceding Practice block, then instruction continues from the command following the ChoosePractice that sent the user into the now-completed Practice block; (2) if the user is not in a Practice block, then everything in the writer's file up to the next EndUnit label is skipped, and instruction continues from the following Unit label.

BeginTiming [timer], EndTiming [timer] BT, ET

The writer has access to five timers, identified by the letters A through E, which maintain the elapsed time (in seconds) between the execution of the BeginTiming and EndTiming commands. He may include as many commands between the BeginTiming and EndTiming as he chooses, and the elapsed time will reflect the user's pace in completing the actions required by any monitoring commands (as well as system response time, which is not eliminated from the elapsed time).

SkipIfFasterThan [seconds, timer] SIFT

This command would normally follow an EndTiming command. The elapsed time maintened by the *timer* is compared to the *seconds* argument; if the elapsed time is less, then the command immediately following the SkipIfFasterThan is ignored, and instruction continues from the next command (i.e., the second command after SkipIfFasterThan). This allows the writer to have a slower user enter a Practice block unconditionally.

SkipIfSlowerThan [seconds, timer] SIST

This command is identical to SkipIfFasterThan, except that the next command is skipped if the elapsed time in the *timer* is greater than the *seconds* argument.

BeginErrorCount [], EndErrorCount [] BEC, EEC

These commands are similar to BeginTiming and EndTiming. The writer uses them to bracket a sequence of commands in which the user's performance is to be monitored on a larger scale. The Introduction counts the number of times that any monitoring commands delivered negative feedback; the result reflects the number of errors that the user made during that sequence of instruction.

SkipIfBetterThan [number] SIBT

This command is similar to SkipIfFasterThan, and would normally follow an EndErrorCount command. If the user made fewer than *number* errors, the command following the SkipIfBetterThan is skipped.

SkipIfWorseThan [number] SIWT

This command is similar to SkipIfSlowerThan. If the user made more than *number* errors, the command following the SkipIfWorseThan is skipped.

ReadFromDocument [name] RFD

This command allows the writer to assemble a long Introduction from a number of shorter documents. Its effect is to continue instruction from the first command in the named document, which must be in the Introduction filedrawer. This command automatically begins a new Unit, whether or not a Unit command appears.

Comment [anything] C

This command is ignored by the Introduction, and allows the writer to insert comments or reminders to himself.

ACTIONS

Assemble Introduction File Drawer Contents

The writer creates his file with the Star editor, as a normal document, and names it "Introduction Source". He may include any fonts and/or faces that he wishes the user to see. Similarly, he creates the contents of any documents or folders that the user will deal with. The writer moves or copies all of these objects into the Introduction file drawer, where they can be accessed by the Introduction facility.

Start Introduction (Writer)

The writer must execute the Introduction in order to debug its contents, which he does by selecting the document containing the first command, and pressing SHIFT-HELP. The Introduction Desktop appears and commands are executed sequentially from the selected document. The writer may terminate execution and return to his "normal" Desktop by pressing STOP at any time. (The "normal" Desktop will appear as though the user had logged off and logged on again; all windows will be closed.)

Start Introduction (User)

The Introduction begins as soon as the user logs on and the system recognizes him as "new", that is, as not having completed the Introduction already. (The Introduction does not provide instruction in logging in.) The Introduction Desktop, including the Message Area and the transient menu, is displayed. Then the Introduction Window opens and instruction begins, as commands from the writer's file are executed. The Directory, blank paper document, and any other objects provided by the System Administrator, appear after the user has completed the Introduction and is ready to begin real work.

If the user ends his session before execution of the writer's file is complete, the last-encountered Unit label is stored as part of the information in the User Icon. The next time he signs on, the Introduction continues from that Unit label.

Progressing Through the Introduction

The menu items in the Introduction Window allow the user to progress through the instruction at his own pace, repeating instruction if he chooses. The user may invoke any one of these menu items, or the End Session, whenever a UserContinue command is in effect.

> **Continue**: The next command in the writer's file is executed. This is the user's standard way of getting new information or instruction, and will certainly be the most-frequently executed menu item in the Introduction.
>
> **Restart Unit**: The Desktop is cleared of all icons and open windows, and instruction continues from the last-encountered Unit label.
>
> **Restart Previous Unit**: The Desktop is cleared of all icons and open windows, and instruction continues from the Unit label preceding the last Unit encountered.

Restart Practice: This command is visible only when the user is in a Practice block. Its effect is to clear the Desktop of all icons and open windows, then continue instruction from the command following the last-encountered Practice label. Thus the user will repeat his current Practice block of instruction.

User's Option to Terminate the Introduction

Whenever a UserContinue command is in effect, the user may exit the Introduction by pressing STOP. The Introduction Window displays the message:

> "If you want to take a break now, and continue this instruction when you return, you should use the End Session menu command. If you have completed the Introduction before, or if you already know how to operate the Professional Workstation, you may leave the Introduction and begin your normal work.
>
> **If you want to continue with the Introduction, press NEXT or use the Continue menu command. If you really want to stop the Introduction, press STOP again."**

If the user presses STOP a second time, the "normal" Star Desktop appears and the Message Area displays:

> "You have left the Introduction. The system is now at your command."

User's Option to Begin the Introduction

During "normal" Star operation, the user may invoke the Introduction by pressing HELP, causing the Help Window to appear, then pressing SHIFT-HELP. (Whenever the Help window is open, pressing SHIFT-HELP initiates this interaction.) The Message Area flashes once, then displays "The Introduction is a long lesson. Press NEXT to begin, STOP to remain in normal operation." If the user performs any mouse action or presses any other key, the Message Area flashes the message again.

If the user presses NEXT, the Introduction Desktop obscures his current Desktop, and commands are executed from the Introduction Source just as though he had logged on for the first time. If he presses STOP, he remains in normal operation and the Message Area displays "OK. Proceed with your normal work."

The user may begin the Introduction only at the beginning. Of course, if he logs off within the Introduction, he returns to the last-encountered Unit label when he logs on again. However, it is not possible to exit the Introduction via STOP, then return to the last-encountered Unit via HELP, SHIFT-HELP. [The Introduction is not a general overview of system features. It is a *special-purpose mode of interaction with Star* that is generally inappropriate for users with more than a few hours' experience on the system. Having the user think of "returning to the Introduction to review some fundamental operations" is probably a mistake.]

End Introduction (User or Writer)

When all the commands in the writer's file have been executed, the Introduction terminates. The Message Area displays "Congratulations! You have completed the Introduction. Good work!". The "normal" Star Desktop appears, with whatever objects are standard for the user's site.

Handling Runtime Errors

As well as interacting with the user via Star software, the Introduction facility operates from the writer's file. It is open to possible runtime errors, some of which cannot be prevented even by exhaustive software checks of the writer's file. If the writer plans his instructions scrupulously, errors of this kind can be minimized. At least during the development of the writer's file, however, and also during genuine customer-site operation, runtime errors must be dealt with in a uniform way.

Whenever the Introduction encounters an impossible situation (for example, an OpenIcon of an icon that is not displayed, or a UserEditParameter when no property sheet is displayed), the message "Something has gone wrong with the Introduction. Please wait while the next unit of instruction is found." is displayed. Then the Introduction Desktop is cleared of all icons and open windows, and the next Unit command is located in the writer's file. All other intervening commands except ReadFromDocument are ignored, and execution proceeds from the Unit command.

A notation is also added to the log containing all other software and hardware errors. Each runtime error notation includes the date and time, the user's name, the current command and its arguments, and a description of the impossible situation, such as "DeleteIcon: Window named *name* not in iconic form."

(THIS PAGE INTENTIONALLY BLANK)

C. HELP AND TRAINING

This section describes the features that allow a user to seek on-line assistance, either brief or detailed, while in the midst of normal Star operations. The Help and Training facility, hereafter referred to simply as Help, provides a simple set of actions; the complexity, organization, and depth of the set of materials made available to the user are determined by the Help writer, not the facility itself.

USER OVERVIEW

Help is a Star facility (encompassing both brief and detailed on-line assistance) available at all times to provide users with information about the system, in the form of a set of *Help documents*. As used here, the term "Help document" refers to a normal Star document, written by a Help writer, containing zero or more *Help References* that give the user access to other Help documents.

When the user wishes Help information, he can either press the HELP key, or invoke the question mark in any window menu, property sheet menu, or the Message Area.

When the HELP key is pressed, the Help Window displays a top-level outline of the on-line materials available. To the extent that these materials fall into different categories, the top-level "map" directs the user to the appropriate system of documents. When the user opens the Help window by invoking a question mark, a Help document containing an explanation of the appropriate menu, property sheet, or message is displayed. Once the Help window is open, the user can either open more detailed Help documents, or use the Search For Keyword menu command. The system indicates which Help documents relate to the user's typed-in keyword by displaying another document whose Help references point to more detailed information. The user can continue working on other documents while the Help window is open.

Help materials consist of a set of documents created by Help writers. The Help writer attribute allows the user to insert Help references into documents, and to examine the system of Help documents for completeness.

The Introduction can also be invoked from the Help window.

DETAILED SPECIFICATION

OBJECTS

Help Writer

The system administrator may designate a user as a Help writer at the time the user's name is registered in the Clearinghouse. Initially, only Xerox personnel who are developing Help materials on their own Star workstations will be so designated.

Help Document

The term "Help document" refers to a normal Star object written by a Help writer, containing zero or more *Help References* that give the user access to other Help documents. In general, the on-line assistance to the user consists of a large set of documents, most of which are accessed by a *reference* from another document; the rest are associated with menus, property sheets, options sheets, or system messages.

Help Key

The HELP key provides one way to open the Help window. The user may press the HELP key at any time while using Star, except during a multi-step process such as a MOVE. If the user presses the HELP key when the Help window is open, a Help document describing the Help system is displayed.

Question Mark

A question mark is located in the menu of every Star window, in every Property Sheet, and in the Message Area when a message is displayed. These question marks are invoked like menu commands, and result in the Help window opening and displaying a relevant Help document. The user can subsequently use the Help features in the standard way.

Help Window

The Help window is a Star window of standard document size usually located on the right side of the screen. As described in the *Desktop* chapter, the location of the Help window (right or left) depends on the "stimulus" by which the user invoked Help. That description is amended as follows: The Help window never covers a property sheet or option sheet, regardless of the location of the current selection. If a property or option sheet is displayed at the time that Help is invoked, the Help window opens on the other side of the screen. (The exact size and screen coordinates of the Help window are listed in the *Desktop* chapter.)

When open, the Help window overlays any other windows in that area; it does not participate in the normal screen division that takes place when a standard window is opened. The Help window displays Help documents and can be scrolled normally. The menu commands in the Help window are: ?, Close, Show Table of Contents, Show Previous View, Search For

Keyword, and Check References (Help writers only).

A document appearing in the Help window is effectively locked; only the contents of fields within the document can be modified. The user may copy material from the Help document at will.

Help Server, Help File Drawer

The complete set of Help documents, comprising the total of all on-line assistance, is expected to be large. At the discretion of the Help writer, some of the documents will reside in the Help folder (created by Xerox personnel, like the System Folder) on each workstation's rigid disk, and some will reside in the Help file drawer stored on a local file server, referred to here as the Help Server. The System Administrator associates the name "Help Server" with the local file server by means of the Clearinghouse. The distinction between locally- and remotely-stored documents will be evident to the user only by virtue of the time required to retrieve material from the server.

The Help file drawer will appear in the divider for its file server in the Directory, but its access will be limited, using the normal file server mechanisms, so that ordinary users cannot modify or overwrite the Help materials.

Although Help documents will be developed as normal non-special objects on the Help writer's Desktop, the system of Help materials available to the paying customer will all reside in a privileged location referred to as "the Help system." Throughout this specification, Help documents will be assumed to reside either on the rigid disk or on the Help Server; the distinction will be made explicit only when it is meaningful.

Table Of Contents

The Table of Contents is a particular Help document, named "Table of Contents," that appears immediately in the Help window when the user presses the HELP key. Its content is determined by the Help writer who develops and coordinates the set of on-line materials.

Help Reference

A Help Reference is a special character that has associated with it the unique name of a Help document. Help references exist within ordinary text like other special characters. The creation and use of Help references are described under *Actions*. A Help reference can only be invoked (with the SELECT button) when viewed through the Help window. When seen in a document window, it is selected the way other characters are selected.

The Help Reference Property sheet has two text parameters and one state parameter. The first text parameter is "Referenced Document," and contains the name of the Help document pointed to by the reference. The second, "Page Number," contains the page to which the document should be scrolled when it appears in the Help window. (Its default is 1). The state parameter is "Use Help Server." If it is set on, the system searches for the Help document on the Help Server; if it is off (the default), the system searches for the document in the local Help folder.

Practice Reference

For training purposes, it will be useful to provide the user with a document, folder, or record file that can be manipulated like any "normal" Star object. For example, a folder might be used to give the user practice in opening documents within the folder window, where the contents of the folder are designed by the Help writer to present a particular training example. A Practice Reference is a special character that points to such a practice object in the same way that a Help reference points to a Help document. The Practice Reference Property Sheet has one text parameter, called "Referenced Object," into which the Help writer types the name of the document, folder, or record file pointed to by the reference. When the Practice reference is invoked, the associated object is opened at its beginning; no page number can be specified. The practice object itself is stored on the Help Server.

The user's action of invoking a Practice reference results in a window of the appropriate type being opened, displaying the object named by the "Referenced Object" parameter. Invoking a Practice reference is a shorthand for opening the Help file drawer, selecting the associated document, record file, or folder, copying it to the Desktop, and opening it. See *Choose A Topic,* under *Actions.*

Keyword Document

A keyword document is a Help document associated by means of its name with a word (or phrase) that the user might be expected to ask about. For each keyword that the user might usefully type, there exists a document whose name is of the form *ExpectedWord.keyword* (e.g., Margins.keyword, or Rectangles.keyword). The keyword document contains Help references to other documents that define or explain the word. The matching algorithm by which the system retrieves the keyword document requires correct spelling but not capitalization. The Help writer must include keyword documents for plurals and synonyms (e.g., Margin.keyword, Rectangle.keyword, Box.keyword, etc.).

The Introduction

The Introduction is a separate Star facility that provides closely monitored instruction in the fundamental concepts and operation of the system. As described in the appendix section titled *The Introduction*, all new users are required to begin this sequence of instruction when they log on for the first time. The user may begin the Introduction at any time during his normal interaction with Star by pressing SHIFT-HELP when the Help window is already open. Information to that effect will appear in the Table of Contents document. The details of entering and leaving the Introduction at the user's option are listed in *The Introduction*.

ACTIONS

Open Help Window

The user gets help by pressing the HELP key, or by invoking one of the question marks visible in a menu or in the Message Area. In all cases, the Help window is opened and the user sees a Help document. If the HELP key is pressed, the Table of Contents is displayed. If one of the question marks is invoked, the displayed Help document explains the window menu, the property sheet, or the message with which the question mark was associated. The following menu commands appear in the Help window:

Show Table of Contents
The Table of Contents is displayed in the Help window.

Show Previous View
The document that is currently displayed in the Help window is replaced by the document that was previously displayed. Show Previous View may be invoked repeatedly up to ten times, during any single "Help session," after which the user will be shown the Table of Contents.

Search For Keyword
This command helps the user locate documents containing information related to a typed-in keyword. When the user invokes it, the Keyword Search Option Sheet is displayed. The user enters the desired text string in the Keyword parameter, and invokes Start. While Star is searching for the document whose name matches the user's type-in, the message "Search in progress" is displayed. The user may cancel the search by pressing the STOP key. When the search succeeds in finding the appropriate keyword document, the option sheet disappears, and the keyword document is displayed in the Help window, containing Help references to other documents related to the keyword. For example, if the user typed "margins," the keyword document (named "Margins.keyword") might show a list of five Help documents related to the topic of setting and changing margins.

If there is no document in the Help system whose name matches the word or phrase typed by the user, the message "Sorry, no Help document is linked to that keyword" appears in the Message Area. The contents of the Help window are not changed.

Check References
This menu item appears to Help writers only. It is used to check the Help system structure for completeness. All the Help and Practice references in all the Help documents on the rigid disk and on the Help Server are examined, to insure that they refer to existing objects, and to insure that all Help and Practice objects are referred to by some reference in the Help system. The Table of Contents and all documents with names ending in ".keyword" are ignored for purposes of identifying documents that are not referenced, but they are considered as pointers *to* other objects.

A document containing a two-column table is generated and displayed in the Help window, listing pairs of names. The first entry in each row is the name of the Help document containing the dangling reference; the second is the name of the non-existent object to which the reference points. In the case of an object that is not pointed to by any reference, the first entry is left blank, and the second entry is the name of the existing but un-referenced object.

Choose A Topic

The user chooses a topic in the Help window by invoking a Help reference with the SELECT button. When a Help reference is invoked, the contents of the document whose name is associated with it are displayed in the Help window, replacing the document already there. Both the Help file drawer and the local Help folder have their "Uniquely Named Contents" properties set on, insuring that only one document in either container is pointed to by any reference. If a given name appears in both containers, precedence is enforced by the "Use Help Server" parameter, specifying which container is to be searched, and by the fact that Practice objects are all stored on the Help Server. If there is no document corresponding to the Help reference at the time it is invoked, the message "Sorry, that Help document does not exist" is displayed.

If the user invokes a Practice reference, the system opens a window on the non-Help side of the screen, and displays the referenced object in that window. (The object itself is copied from the Help Server to the user's Desktop, and it then exists as a normal Star object which can be manipulated, stored, deleted, etc. in the normal ways.) If there is no room on the non-Help side of the screen, a message is displayed advising the user to close at least one window to provide space. Since the Practice object provides a workspace for the user, the Help writer will presumably not lock Practice documents.

When a Practice window is closed, it returns to its Desktop location (one that was not obscured by any window at the time the Practice reference was invoked, if possible).

Close Help Window

The user closes the Help window whenever he is satisfied with the on-line assistance he has received. All Help documents that were copied from the Help Server are deleted automatically and invisibly, except for objects that were accessed by means of a Practice reference, which remain in their current state on the user's Desktop as completely standard objects.

Create New Summary Sheet

The Help writer creates a new Help document by typing into a blank document. The new Help document is added to the Help system when it is moved or copied into the Help file drawer or added to the Help folder.

Create Help or Practice Reference

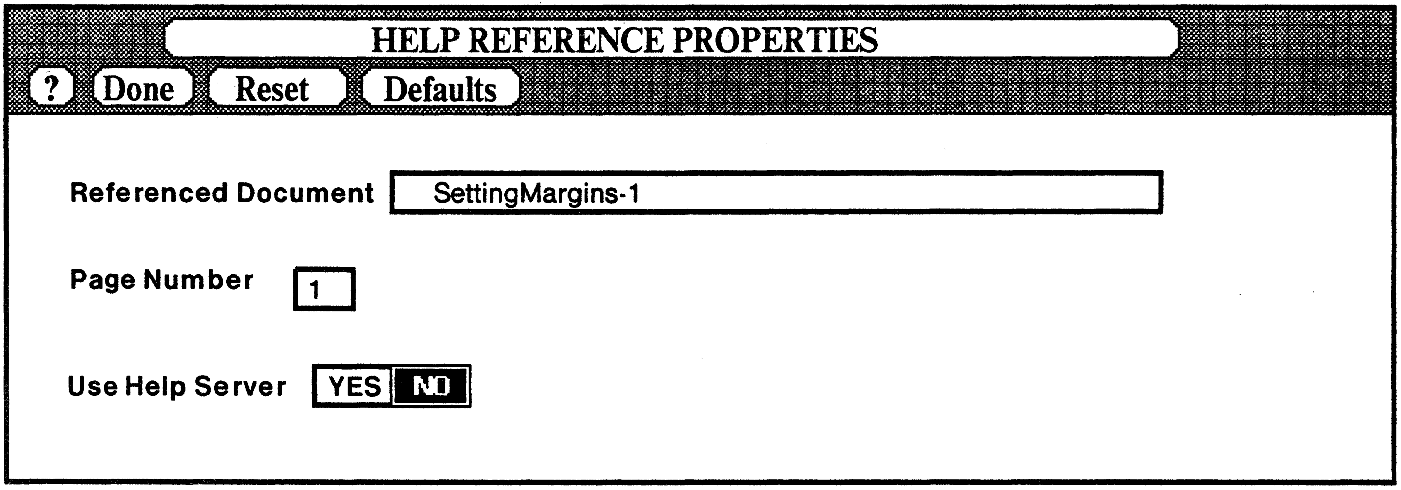
To create a Help or Practice reference, the Help writer inserts the appropriate reference character into a document. This is done using the KEYBOARD key as described in *Text Editing*. (This option is available only to Help writers.) The writer can select the reference and move it around in the document. The writer brings up its Property Sheet and types the unique name of the pointed-to object into the Referenced Document or Referenced Object parameter. For Help references, he types the page number at which the document is to be scrolled, and sets the Use Help Server parameter appropriately.

CHANGES

Pressing HELP key when Help window is open displays Help document describing Help system.

Maximum width and height, screen coordinates for Help window copied from *Desktop* section.

To get The Introduction from Help window, press SHIFT-HELP.

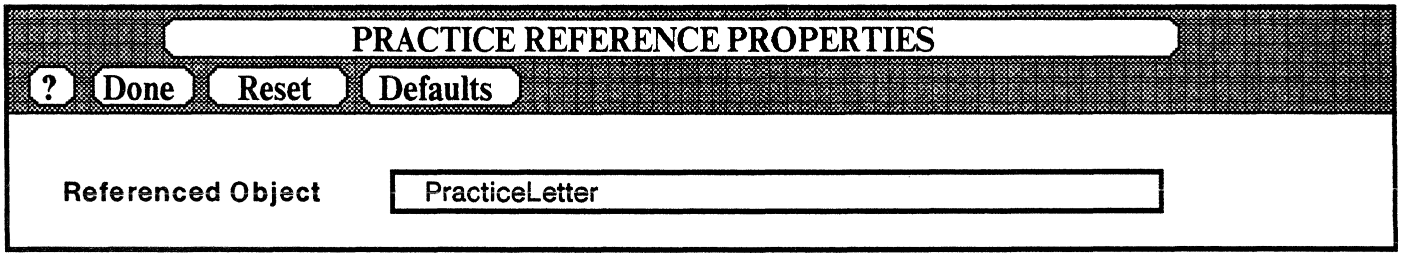


Referenced Document: SettingMargins-1

Page Number: 1

Use Help Server: YES [✓ NO]

Figure HLP-1 - Help Reference Property Sheet



Referenced Object: PracticeLetter

Figure HLP-2 - Practice Reference Property Sheet

(PAGES 439 THROUGH 444 INTENTIONALLY BLANK)

(PAGES 439 THROUGH 444 INTENTIONALLY BLANK)

(PAGES 445 THROUGH 448 INTENTIONALLY BLANK)

(PAGES 445 THROUGH 448 INTENTIONALLY BLANK)