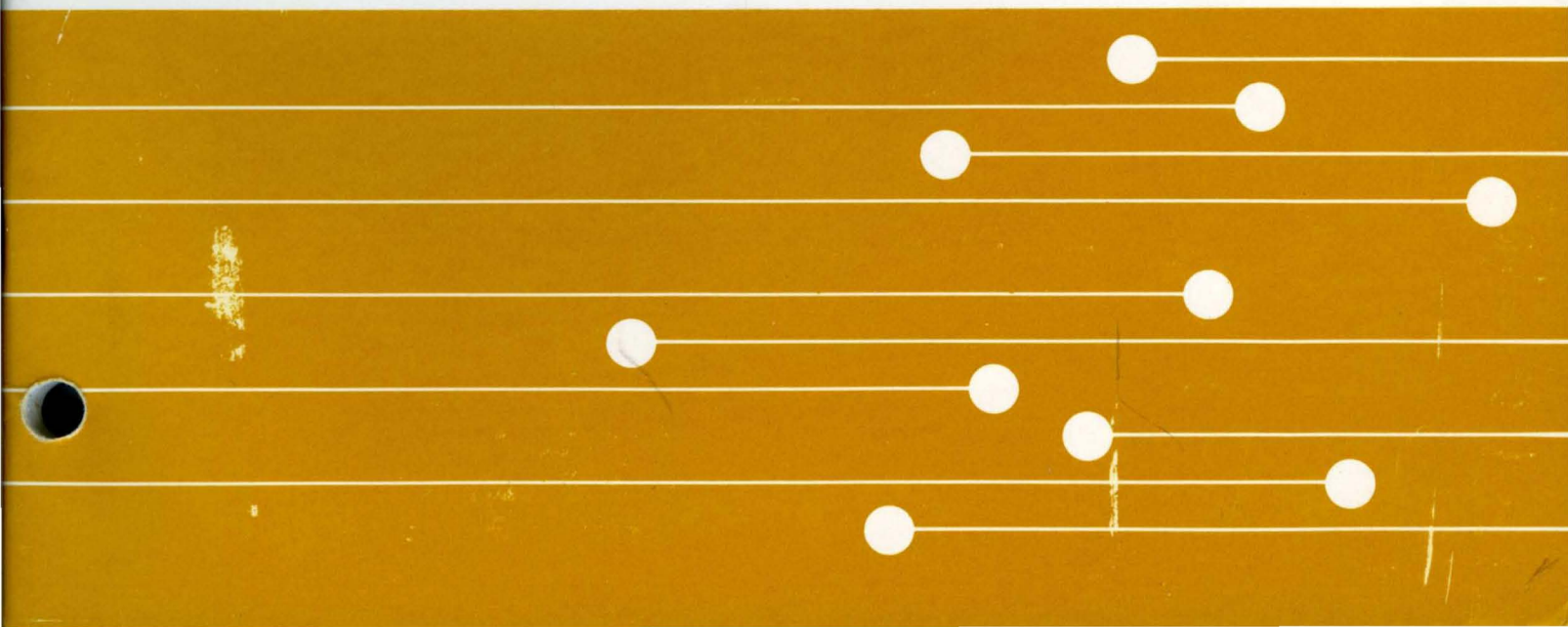
Publication Number
SC28-1306-0
File Number
S370-39

Program Number
5665-285

TSO Extensions Session Manager Program Reference

IBM



Publication Number
SC28-1306-0
File Number
S370-39

Program Number
5665-285

TSO Extensions Session Manager Program Reference

First Edition (June 1984)

This edition applies to TSO Extensions (TSO/E) Licensed Program, Program Number 5665-285 and all subsequent releases until otherwise indicated in new editions or Technical Newsletters. Changes are made periodically to the information herein: before using this publication in connection with the operation of IBM systems, consult the latest *IBM System/370 Bibliography*, GC20-0001, for the editions that are applicable and current.

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any references to an IBM program product in this publication is not intended to state or imply that only IBM's program product may be used. Any functionally equivalent program may be used instead.

Publications are not stocked at the address given below. Requests for IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form for readers' comments is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Corporation, Information Development, Department D58, Building 920, PO Box 390, Poughkeepsie, New York 12602. IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Preface

This publication describes the Session Manager default environment, how you can change it to suit your particular needs, and the syntax and function of the Session Manager commands. It is intended for TSO users who are already familiar with how the Session Manager works and want to use the commands to change the screen layout, program function (PF) keys, or other attributes related to the Session Manager environment. If you are not familiar with the Session Manager, first read *TSO Extensions Session Manager Terminal User's Guide.*

The major sections of this book are:

- "Introduction"
- "Controlling the Session Manager Environment"
- "Session Manager Commands"
- "Using Command Procedures (CLISTs)"

The **"Introduction"** describes what the Session Manager is and how it makes TSO easier to use.

"Controlling the Session Manager Environment" describes how you can change the IBM-supplied default environment. It includes the commands used to change the environment and examples of when and why you might want to use these commands.

"**Session Manager Commands"** describes the syntax and function of each command and provides examples of using the commands. The commands are presented in alphabetical order.

"Using Command Procedures (CLISTs)" describes several ways of using CLISTs to redefine program function (PF) keys and to split your display screen. For a complete description of how to create, edit, invoke, and execute a CLIST, see *CLISTs: Implementation and Reference.*

Related Publications

- *System Programming Library: TSO Extensions Planning and Installation Volume 1*, SC28-1379
- *System Programming Library: TSO Extensions User Exits and Modifications Volume 2*, SC28-1380
- *System Programming Library: TSO Extensions Command and Macro Reference Volume 3*, SC28-1381
- *TSO Extensions CLISTs: Implementation and Reference*, SC28-1304
- *TSO Extensions User's Guide*, GC28-1333
- *TSO Extensions Session Manager Logic Manual*, LY28-1312
- *TSO Extensions Session Manager Terminal User's Guide*, SC28-1305
- *TSO Extensions Command Language Reference*, SC28-1307
- *TSO Extensions Terminal Messages*, GC28-1310

Contents

Figures

Introduction

The TSO Session Manager is a part of MVS TSO Extensions (TSO/E) program product. It makes TSO easier to use and takes advantage of many of the features on a display terminal.

The Session Manager:

- Maintains complete journals of all work generated during a terminal session. These journals are called **streams**. You can use the streams to review prior input and output, construct new input from existing information, save portions of the records in data sets, or copy the records to a system printer.

- Supports program function (PF) keys. In the default environment, most of the PF keys are defined as Session Manager commands that move a window over a stream. You can change the PF keys to issue any commands you like.

- Allows you to tailor the layout of the display screen and the terminal environment to suit your individual needs. You can change the screen layout and other characteristics about the default environment by using the Session Manager commands. You can also save these definitions before and after you change them and later restore them to their previous meaning.

- Provides a split screen capability allowing you to view different information at the same time.

- Supports all TSO commands and line-oriented functions.

The Session Manager operates as an interface to TSO and is invoked by simply pressing the ENTER key after logging on. It supports all display terminals equipped with either 12 or 24 PF keys.

Controlling The Session Manager Environment

One of the features of the Session Manager is that you can change the default environment (the program function (PF) keys, streams, and screen layout) to suit your specific needs. This section explains how the Session Manager commands are used to set up this environment, how you can use the commands to change the environment, and provides examples of when these changes might be appropriate. For complete descriptions of all commands referred to in this section, see the command reference section of this book.

Streams

The Session Manager keeps several records of the different things that happen during your terminal session - all the commands, instructions, and input you enter and all the output from commands and messages that the system issues. These records are **streams**.

A stream is similar to a long sheet of paper that fills up with information as your terminal session progresses. The Session Manager places each line of information entering a stream in the next available line starting at the top.

The three types of Session Manager streams are **input**, **output**, and **extra**.

Input Stream

An input stream contains information that you want interpreted as commands or input to programs.

Output Stream

An output stream contains the output from commands and any messages from other TSO users, the operator, or from background jobs. You can also copy information from an input stream into an output stream, thereby creating a complete record of all commands entered and all of the output from those commands.

Extra Stream

You can copy information, such as the output from certain commands, into an extra stream. In the default environment, an extra stream is used to contain the lines, windows, and arrows that visually divide the display screen. You cannot specify an extra stream on any of the Session Manager commands when an input or output stream is required.

Changing the Streams

In the IBM-supplied default environment, nine streams have been defined. You cannot change the names of any of the streams, but you can use the CHANGE.STREAM command to:

- Erase the information in a stream
- Change whether or not the terminal's audible alarm sounds when information enters the stream

Figure 1 summarizes the attributes of all of the streams in the default environment. It identifies the stream name, type, size (in lines and bytes) and indicates whether or not the audible alarm sounds when information enters the stream.

Stream Name	Type	Size		Alarm
		Lines	Bytes	
TSOIN	Input	305	8192	No
TSOOUT	Output	4005	147456	No
SMIN	Input	305	8192	No
SMOUT	Output	155	4096	No
MESSAGE	Output	55	1024	Yes
EXTRA1	Output	405	32768	No
EXTRA2	Extra	105	1024	No
EXTRA3	Extra	105	1024	No
HEADER	Extra	55	1024	No

Figure 1. Streams

Session Functions

While working at the terminal, you are communicating with TSO and the Session Manager. In addition, you can receive messages from other TSO users, the operator, and from jobs you're executing. These three means of communication (TSO, Session Manager, and messages) are called **session functions**. The session functions receive the input you enter at the keyboard and generate output to you. The Session Manager places the input and output in streams, either the default streams or the streams that you specify on the CHANGE.FUNCTION command.

TSO Function

TSO has an input and an output stream.

TSOIN is the input stream for the TSO function. Information placed in this stream is interpreted as TSO commands or input to programs. While a command is executing in this stream, the Session Manager highlights it. Otherwise, when the command has completed execution, it is displayed at normal intensity.

TSOOUT is the output stream for the TSO function. All output generated by the TSO commands and programs is placed in this stream. The TSOOUT stream also contains a copy of the TSO commands from the TSOIN stream, thus creating a complete record of your TSO session. (The commands from the TSOIN stream appear highlighted in this stream.) In addition, the MESSAGE (MSG) function places in the TSOOUT stream any messages you receive from other TSO users, the operator, or from background jobs. As a result, the messages are interleaved with the TSO session input and output.

Session Manager (SM) Function

The Session Manager also has an input and output stream.

SMIN is the input stream for the SM function. Information placed in this stream is interpreted as Session Manager commands.

SMOUT is the output stream for the SM function. Any error messages from the Session Manager commands are placed in this stream. The SMOUT stream also contains a copy of the Session Manager commands from the SMIN stream.

However, you cannot see the commands at your terminal because they are copied at an intensity of 0. This is done so that the PASSWD window on your screen only displays the error messages.

Message (MSG) Function

You must place the messages you receive from other TSO users, the operator, or from jobs you're executing in an output stream. In the default environment, the TSOOUT stream contains any messages you receive interleaved with your TSO input and output. The message (MSG) function cannot have an input or a copy stream.

Changing the Session Function Streams

Figure 2 summarizes the session functions and their attributes. You cannot specify an input stream for the message (MSG) function. Therefore, this area has been left blank in the figure below.

Intensity (INT) refers to the brightness at which the information is displayed in a stream. The meanings of the numbers are:

0 The information in the stream is not displayed. You can see the line that the information occupies, but the information itself is invisible.

1 The information in the stream is displayed at normal intensity.

2 The information in the stream is highlighted.

ALARM indicates whether or not the terminal's audible alarm sounds when information enters the stream.

Session	Input Stream				Output Stream		
Function	Name	Alarm	Where Copied	INT	Name	INT	Alarm
TSO	TSOIN	NO	TSOOUT	2	TSOOUT	1	NO
SM	SMIN	NO	SMOUT	0	SMOUT	2	YES
MSG	-	-	-	-	TSOOUT	2	YES

Figure 2. Session Functions

You can change the input or output streams for any of the session functions, any attributes relating to those streams, and whether the alarm is to sound. For example, you might want your messages to go to the MESSAGE stream instead of the TSOOUT stream or you might not want the TSOIN stream copied to the TSOOUT stream. With the CHANGE.FUNCTION command, you can specify:

- The output stream for a session function and the intensity at which the information is to be displayed
- The input stream for a session function
- Whether or not information from an input stream is to be copied to an output stream and the intensity at which the information is to be displayed
- Whether or not the audible alarm is to sound when information enters an input or output stream

The commands that define the streams and their characteristics for each of the session functions in the default environment are:

```
CHANGE.FUNCTION TSO INPUT(TSOIN) COPY(TSOOUT 2)
   OUTPUT(TSOOUT 1) ALARM(NO)

CHANGE.FUNCTION MSG OUTPUT(TSOOUT 2) ALARM(OUTPUT)

CHANGE.FUNCTION SM INPUT(SMIN) COPY(SMOUT 0)
 OUTPUT(SMOUT 2) ALARM(OUTPUT)
```

For more information, see the syntax description for the CHANGE.FUNCTION command.

Changing the Screen Layout

The default screen layout consists of the definitions for the windows and the location of the cursor on the display screen. This section describes:

- The definition and characteristics of a window
- How to change and delete windows on the screen
- How to scroll a window
- How to lock and unlock a window
- How to display streams
- How to change the location of the cursor on the screen

Defining A New Window

A **window** is a physical area of the display screen that you can use to type into and/or display a particular stream.

You can define a window by giving it all of the desired attributes at once using the DEFINE.WINDOW command, or you can issue that command followed by one or more CHANGE.WINDOW commands to build the attributes for the window. The attributes for the window remain in effect until modified via the CHANGE.WINDOW command.

Following are the attributes you need to know to define a window. The operand for specifying each attribute is enclosed in parentheses.

- The name of the window (window-name).

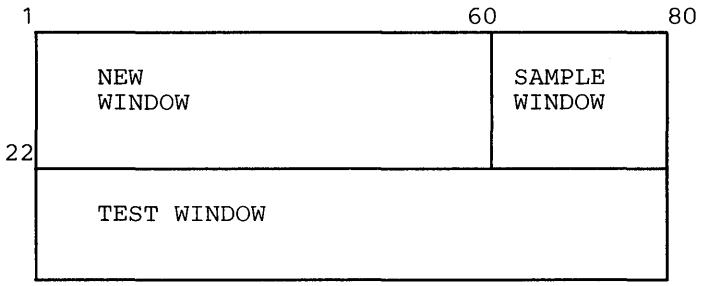
- The location of the window on the display screen (row, column, lines, and width).

  "Row" refers to the number of the top line in the window. "Column" specifies which column of the screen the left side of the window is to occupy. "Lines" specify the number of lines in the window. "Width" specifies the number of character positions in each line. If you specify a negative value for either "row" or "column," the Session Manager interprets the value as relative to the bottom or right side of the screen, respectively.

  You can use the word MAX for either the "line" or "width" operand instead of specifying a number. If you specify MAX for the line operand, the window contains the remaining lines on the screen or until a line is encountered that has been defined for another window. If you specify MAX for the width operand, the window's width is determined by the number of character positions in the first line of the window.

In the example below, the TEST window was defined first beginning in row 22, column 1 and the SAMPLE window begins in row 1, column -20. If you define a new window beginning in row 1, width MAX, and lines MAX, the window covers rows 1 through 21 and columns 1-59.

Note: The screen in the example has a column width of 80.



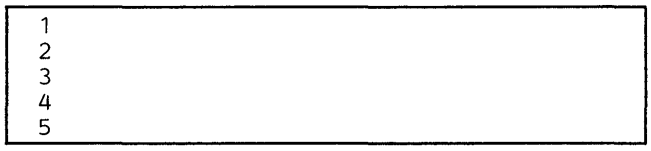
You can also specify WRAP for the width value instead of MAX or a number. In this case, the width of the window starts from the column value specified on the command and continues to either the beginning of the next window or to the last row and column on the screen. The WRAP operand can only be used when the value for "lines" is 1.

Note: The first character position in a window is used as a terminal attribute byte and is protected. Therefore, a window defined with a width of 1 is useless.

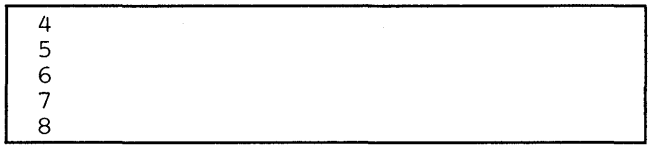
- Whether or not you want the terminal's audible alarm to sound when the window scrolls to display new information in the stream (ALARM).

- How long you want a window to be held in place before the Session Manager scrolls it (HOLD). The window can remain at one position a specified number of seconds or until you press the ENTER or any program function (PF) key. While the window is held in place, the keyboard is locked.

- How many lines are to be repeated in the window when it scrolls over the stream (OVERLAP). For example, if you specify OVERLAP(2) in defining a window, the bottom or top 2 lines in the window are repeated when you move the window forward or backward, respectively.



If you move the above window forward a complete page, lines 4 and 5 are repeated.



- Whether or not you can enter data in the window (PROTECT).

- The name of the stream that is to receive the information typed in the window and the intensity at which the information is to be displayed (TARGET).

- How much information is to enter the specified stream before the window scrolls to display it (UPDATE). When a window is displaying the bottom of a stream, it waits for new information to enter the stream before it moves. With the UPDATE operand, you can specify whether or not the window is to display every line of information as it enters the stream, every page of information, or just the newest information. In the latter case, some lines of information might be skipped over because the window scrolls directly to the bottom of the stream.

- The name of the stream the window is to display (VIEW).

For more information on the above operands, see the syntax description for the DEFINE.WINDOW command in the reference section of this book.

When you define a window on the display screen, the physical location of the window cannot exceed the physical limits of the display screen unless you define the window as one line that wraps the screen. In addition, a new window cannot overlap the physical location of an existing window.

Changing a Window

Instead of defining a new window on the display screen, you might want to simply change the attributes of an existing window. The following attributes of a window can be changed with the CHANGE.WINDOW command:

- Whether or not you want the terminal's audible alarm to sound when the window scrolls to display new information in the stream

- How long the window is to be held in place before the Session Manager scrolls it

- How many lines are to be repeated in the window when it scrolls over the stream

- Whether or not you can enter data in the window

- The name of the stream that is to receive the information typed in the window and the intensity at which the information is to be displayed

- How much information is to enter the specified stream before the window scrolls to display it

- The name of the stream the window is to display

You cannot change the window name, row, column, lines, or width values for an existing window unless you delete the entire window and redefine it.

Note: The previous topic "Defining a New Window" describes each of these attributes.

Deleting a Window

You can use the DELETE.WINDOW command to delete one or all of the windows on the display screen. The space previously occupied by the window(s) can then be defined as another window or windows.

If you delete all of the windows, you cannot enter input on the screen unless you press the CLEAR key. See "Entering Session Manager Commands" in the reference section of this book. To reestablish the default screen layout after deleting any of the windows, issue the RESET command.

Displaying Streams

Each window in the default environment displays a particular stream. The name of the stream is set in the DEFINE.WINDOW command with operand VIEW(stream-name). You can change the stream by using the same operand on the CHANGE.WINDOW command. The following paragraph describes some instances in which you might want to change the stream that the MAIN window displays.

In the default environment, the MAIN window displays the TSOOUT stream that contains a copy of all TSO commands entered, the output from those commands, and any messages received from other TSO users, the operator, or background jobs. Occasionally, you might want to have the MAIN window display only the TSO commands entered up to a particular point in your terminal session. In this case, you can change the MAIN window to display the TSOIN stream and use the scroll PF keys to review the entire stream.

Having the MAIN window display the TSOIN stream is also useful when executing and monitoring many commands at one time. For example, if you wanted to compile several programs, you can enter all of the commands to do the compilations at one time and then monitor the execution of those commands (the Session Manager highlights each command as it is being executed in the TSOIN stream).

To display a list of the windows defined in the default environment and their attributes, use the QUERY.WINDOW or QUERY.TERMINAL command.

Scrolling a Window

Several Session Manager commands allow you to move a window over the stream it is displaying. This process of moving a window over a stream is called **scrolling** and is controlled with the FIND and SCROLL commands.

With the FIND command, the Session Manager scrolls the window so that the specified text-string is the top line in the window. With the SCROLL commands, you can control the direction and amount the Session Manager scrolls the window. Both of these commands automatically lock the window in place once it has moved. The window remains at the locked position until you issue another SCROLL or FIND command or an UNLOCK command.

In the default environment, the program function (PF) keys are defined to issue the SCROLL and FIND commands. See the section "Changing Program Function (PF) Keys" for a description of how these keys are defined.

How to Lock or Unlock a Window

Whenever you or the system enters data in a stream, you can easily scroll towards the bottom of the stream by pressing the ENTER key each time the window fills up with information. Sometimes, however, you don't want your data to move, for example, when you are scrolling or using the FIND command to locate a specific text-string. Therefore, the Session Manager automatically locks the window when you issue the FIND or SCROLL commands.

When a window is locked, it means that the window is frozen at its current position and will not move to display new information even if you press the ENTER key. In the default environment where several of the program function (PF) keys issue the FIND and SCROLL commands, pressing the key causes the default window (MAIN) to lock. This window locks so that when you use a PF key to scroll over the information in a stream, the window remains at that position until you are ready to move it. A locked window, however, does not mean that new information cannot enter the bottom of the stream.

An unlocked window in the default environment means that each time you press the ENTER key, the MAIN window automatically moves to the next half-page of information.

You can unlock a window at its present position, at the bottom of the stream, or at the information it was displaying when it was last unlocked by using the UNLOCK command. After issuing this command, you can automatically move the window forward over the stream by pressing the ENTER key.

The Session Manager Display Screen

Figure 3 illustrates the windows on the Session Manager default display screen. Figure 4 lists the attributes of each window (the operands used to define it). The windows were defined in the order listed at the top of the figure to take advantage of the MAX and WRAP operands on the DEFINE.WINDOW command.

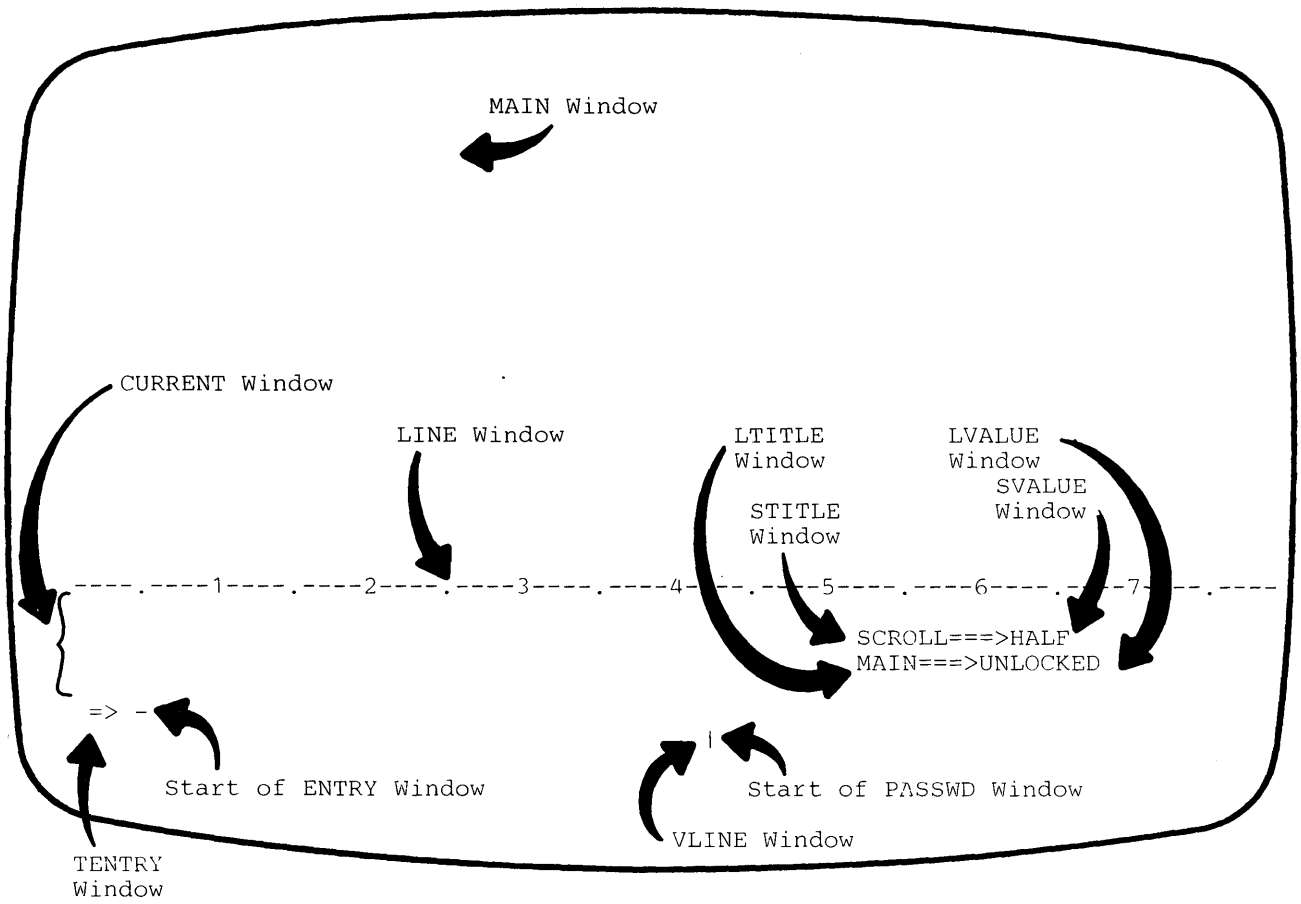


Figure 3. Session Manager Display Screen

	WINDOW NAME										
OPERANDS	**LINE**	**STITLE**	**SVALUE**	**LTITLE**	**LVALUE**	**VLINE**	**PASSWD**	**CURRENT**	**TENTRY**	**ENTRY**	**MAIN**
row	-5	-4	-4	-3	-3	-1	-1	-4	-2	-2	1
column	1	-18	-6	-18	-9	-40	-38	1	1	6	1
lines	1	1	1	1	1	1	1	2	1	1	MAX
width	MAX	12	6	9	9	2	38	MAX	5	WRAP	MAX
ALARM	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
HOLD	0	0	0	0	0	0	0	0	0	0	INPUT
OVERLAP	0	0	0	0	0	0	0	0	0	0	0
PROTECT	YES	YES	YES	YES	YES	YES	NO	NO	YES	NO	NO
TARGET STREAM			EXTRA3				TSOIN	TSOIN		TSOIN	TSOIN
UPDATE	NEWEST	NEWEST	NEWEST	NEWEST	NEWEST	NEWEST	NEWEST	NEWEST	NEWEST	NEWEST	LINE
VIEW STREAM	HEADER line 2	HEADER line 6	EXTRA3	HEADER line 9	HEADER line 7 line 8	HEADER line 4	SMOUT	TSOOUT line 3	HEADER line 1	HEADER line 10	TSOOUT

Figure 4. Session Manager Default Display Screen Window Definitions

Window	Description
MAIN	The MAIN window is the large portion of the display screen above the numbered line. This window displays the TSOOUT stream which contains a copy of all the TSO commands you enter, all output from those commands, and any messages you receive from other TSO users, the operator, or background jobs. If you enter information in the MAIN window, the Session Manager sends it to the TSOIN stream to be executed as a TSO command.
LINE	This window displays the numbered line located on line 2 in the HEADER stream. If you try to type in the LINE window, the keyboard locks. Press the RESET key to unlock the keyboard.
CURRENT	Just below the numbered line is the CURRENT window. Initially, this window displays the last two lines in the TSOOUT stream. If you press PF6/18, the CURRENT window changes to display the last two lines in the TSOIN stream (the last two commands you entered). See "Changing Program Function (PF) Key Definitions" for a complete description of how PF6/18 changes the CURRENT window between displaying the TSOOUT and TSOIN streams. If you enter information from the CURRENT window, the Session Manager sends it to the TSOIN stream to be executed as a TSO command.
TENTRY	The TENTRY window displays an arrow located on line 1 in the HEADER stream. If you try to type in this window, the keyboard locks . Press the RESET key to unlock the keyboard.
ENTRY	The ENTRY window begins after the TENTRY window and continues to the vertical bar on the last line of the screen. This window 'wraps the screen' which means that it is defined as one line that physically covers two or more lines on the screen. The ENTRY window is where you normally enter TSO commands to go to the TSOIN stream although you can enter them from the MAIN, CURRENT, and PASSWD windows. This window displays a blank line located on line 10 in the HEADER stream.
STITLE	The STITLE window displays the word "SCROLL" located on line 6 in the HEADER stream. This window, in combination with the SVALUE window, lets you know what the scroll value is for the scroll PF keys. If you try to type in the STITLE window, your keyboard locks. Press the RESET key to unlock the keyboard.
SVALUE	This window displays the last line in the EXTRA3 stream which contains the scroll amount value for the scroll PF keys. Initially, this value is set at HALF which means that if you press one of the scroll PF keys, the MAIN window scrolls one half-page. You can change the scroll amount value by typing a new value in the MAIN, CURRENT, ENTRY, or PASSWD windows and pressing PF2/14. If you try to type directly in the SVALUE window, the keyboard locks. Press the RESET key to unlock the keyboard. The valid scroll amounts are: • Page: Scrolls one page, with page being the number of lines in your MAIN window. You can abbreviate page by typing in 'P'. • Half: Scrolls one-half page. You can abbreviate half by typing in 'H'. • Max: Scrolls the maximum length or width of the stream depending on which scroll PF key you press. You can abbreviate max by typing in 'M'. • A number from 1 to 999999: Scrolls that number of lines forward or backward when you press PF7/19 or PF8/20. • A number from 1 to 32767: Scrolls that number of columns left or right when you press PF10/22 or PF11/23. See "Changing Program Function (PF) Key Definitions" for a complete description of how PF2/14 sets the scroll amount value for the scroll PF keys.
LTITLE	The LTITLE window displays the word "MAIN" on line 9 of the HEADER stream. This window, in combination with the LVALUE window, lets you know if the MAIN window is locked or unlocked. If you try to type in the LTITLE window, the keyboard locks. Press the RESET key to unlock the keyboard.

LVALUE — The LVALUE window displays either "UNLOCKED" (on line 7) or "LOCKED" (on line 8) in the HEADER stream. These words refer to the status of the MAIN window. When you issue a SCROLL or FIND command, the Session Manager automatically locks the MAIN window. The PF keys that issue these commands also issue a command to move the LVALUE window over the word "LOCKED." To unlock the MAIN window, issue an UNLOCK command or press PF12/24 which issues that command followed by a command that moves the window over the word "UNLOCKED."

See "Changing Program Function (PF) Key Definitions" and "How to Lock and Unlock a Window" for more information.

VLINE — The VLINE window displays a vertical bar on line 4 in the HEADER stream. If you try to type in this window, your keyboard locks. Press the RESET key to unlock the keyboard.

PASSWD — Located after the VLINE window is the PASSWD window. This window can be used to type in any information you do not want displayed on the screen (such as the passwords for data sets). Any information you enter from this window goes to the TSOIN stream. The PASSWD window displays the SMOUT stream. If you receive an error message from a Session Manager command, you will see the message in this window.

Changing the Location of the Cursor

The CHANGE.CURSOR command lets you change the location to which the cursor returns after you press a program function (PF) key, the ENTER key, the CLEAR key, the attention (PA1) key, or the cancel (PA2) key. You can establish a **permanent** or **temporary** location.

If you define a permanent location, the cursor returns to that location each time you press one of the mentioned keys. If you define a temporary location, the cursor moves to and remains at that location until the next keyboard entry. After the keyboard entry, the cursor moves to the permanent location.

The cursor is defined as being on a particular row and column within a window. In the default screen layout, the permanent location of the cursor is defined to be row 1 column 1 of the ENTRY window. A temporary cursor location is not defined.

Changing Program Function (PF) Key Definitions

Program function (PF) keys can be defined as TSO or Session Manager commands, input to programs, or text-strings that comment your streams. Use the CHANGE.PFK command to modify the meaning of any of the PF keys. This command requires that you enter the number of the PF key to be changed, the definition-text-string for the key, and the name of the stream where the definition-text-string is to be placed when the PF key is pressed.

You can make a PF key a TSO command (or commands) by defining the definition-text-string as a valid TSO command or subcommand and specifying TSOIN as the target stream-name. This is useful when working with programs that prompt for input or commands that have subcommands. For example, you can define a PF key as often-entered subcommands of the TSO TEST command, such as SAVE or GO.

You can make a PF key a Session Manager command (or commands) by defining the definition-text-string as a valid Session Manager command and specifying SMIN as the target stream-name. This is done in the default environment to allow you to use the PF keys for scrolling.

If you have the definition-text-string sent to another stream (such as the TSOOUT stream), the PF keys can be used to comment your streams.

When you press a PF key, the Session Manager:

- Places each line of information typed on the screen since the last time you pressed the ENTER, CLEAR, PA1, PA2, or any PF key in the appropriate stream, just as if you had pressed the ENTER key.

- Places the definition-text-string of the PF key definition in the target stream.

You can also define PF keys to perform symbolic substitution when they are pressed. When you define the PF key, you can specify **symbolic arguments** within the definition-text-string and specify the SUBSTITUTE keyword operand to indicate that symbolic substitution is to be performed. Each symbolic argument is denoted by an ampersand (&) followed by an integer (or an asterisk). For example,

```
&1., &2., &3., ...&n., and &*
```

If the ampersand appears elsewhere in the definition-text-string when SUBSTITUTE is specified, it must be doubled.

You must provide parameters (tokens) to this symbolic substitution process by entering them on a line of the screen (separated by one or more blanks), and pressing the PF key.

The Session Manager performs the symbolic substitution for each new line of information found on the screen when you press a PF key. It:

- Scans the line of information once from left to right and breaks the line into as many words (tokens) as there are numbered symbolic arguments in the PF key definition-text-string. This process is called tokenization. The remainder of the text in the line becomes the '*' token. If there are no numeric symbolic arguments in the text, the entire line of information becomes the '*' token.

- Substitutes the words (tokens) into the PF key definition-text-string, replacing the symbolic arguments. If there are more symbolic arguments than words, null characters are substituted for those arguments. If there are more words than symbolic arguments, the extra words are ignored. If there are no modified fields on the screen, null characters are substituted for each argument in the text-string.

- Places the resulting definition-text-string in the next available line of the target stream.

The periods immediately following the numeric symbolic arguments are used to distinguish the arguments from the characters immediately following them in the definition-text-string. The periods are not needed if the character after the argument is not a digit.

The example section of the CHANGE.PFK command includes several examples of how the PF keys can be defined.

Most PF keys in the default environment are defined as a string of Session Manager commands that are placed in the SMIN stream.

Figure 5 summarizes the default definitions of the PF keys. PF keys 13-24 have the same definition as PF keys 1-12.

PF1/13 Not Defined	PF2/14 Set Scroll Amount	PF3/15 Not Defined
PF4/16 Not Defined	PF5/17 Find/Repeat Find ↑	PF6/18 Change CURRENT Window
PF7/19 Scroll Backward ↑	PF8/20 Scroll Forward ↓	PF9/21 Scroll Top ↑
PF10/22 Scroll Left ←	PF11/23 Scroll Right →	PF12/24 Scroll Bottom ↓ Unlock

Figure 5. Program Function (PF) Key Definitions

The following section describes how the PF keys are defined in the default environment. Most of the keys are defined as several Session Manager commands that work together to create the desired effect. Unless you are using the split screen capability or have changed the definition for the default window, all references to the default window mean the MAIN window. For a complete description of each command, refer to the command reference section of this book.

PF1/13 PF1/13 is not defined in the default environment. If you press this key, the Session Manager does the following:

- Places the message "PF1 NOT DEFINED" or "PF13 NOT DEFINED" in the SMOUT stream
- Moves the PASSWD window so that the window displays the message

The commands that issue the message and move the PASSWD window over the SMOUT stream are:

```
CHANGE.PFK 1 'PFK 1 NOT DEFINED' SMOUT;
             CHANGE.WINDOW PASSWD
             VIEW(SMOUT)
```

PF2/14 PF2/14 accepts the value typed on the screen for the scroll amount field and places this value highlighted in the EXTRA3 stream. It then changes the definitions of the scroll PF keys (PF7/19, PF8/20, PF10/22, and PF11/23) to do the following:

- Set the scroll value to zero. This is done so that the default window will not move if an invalid value is entered.
- Scroll the amount entered. The Session Manager substitutes the value entered in the place of the symbolic argument in the AMOUNT operand.
- Scroll the LVALUE window over line 8 in the HEADER stream so the word "LOCKED" is displayed.
- Unlock the terminal keyboard so additional scrolling PF keys can be pressed.
- Scroll the LINE window the entered amount (for PF keys 10/22 and 11/23 only).

The commands that define PF2/14 are:

```
CHANGE.PFK 2 'PUT ''&1'' EXTRA3 INTENSITY(2);
              CHANGE.PFK 7 ''SCROLL.BACK 0;
               SCROLL.BACK AMOUNT(&1);
               SCROLL.ABSOLUTE 8 LVALUE;
               CHANGE.TERMINAL CONTROL(0)''
               SMIN;
              CHANGE.PFK 8 ''SCROLL.FORWARD 0;
               SCROLL.FORWARD AMOUNT(&1);
               SCROLL.ABSOLUTE 8 LVALUE;
               CHANGE.TERMINAL CONTROL(0)''
               SMIN;
              CHANGE.PFK 10 ''SCROLL.LEFT 0;
               SCROLL.LEFT AMOUNT(&1);
               SCROLL.LEFT LINE AMOUNT(&1);
               SCROLL.ABSOLUTE 8 LVALUE;
               CHANGE.TERMINAL CONTROL(0)''
               SMIN;
              CHANGE.PFK 11 ''SCROLL.RIGHT 0;
               SCROLL.RIGHT AMOUNT(&1);
               SCROLL.RIGHT LINE AMOUNT(&1);
               SCROLL.ABSOLUTE 8 LVALUE;
               CHANGE.TERMINAL CONTROL(0)''
               SMIN' SMIN SUBSTITUTE
```

PF3/15 PF3/15 is not defined in the default environment. If you press this key, the Session Manager does the following:

- Places the message "PF3 NOT DEFINED" or "PF15 NOT DEFINED" in the SMOUT stream
- Moves the PASSWD window so that it displays the message

The commands that issue the message and move the PASSWD window over the SMOUT stream are:

```
CHANGE.PFK 3 'PFK 3 NOT DEFINED' SMOUT;
              CHANGE.WINDOW PASSWD VIEW(SMOUT)
```

PF4/16 PF4/16 is not defined in the default environment. If you press this key, the Session Manager does the following:

- Places the message "PF4 NOT DEFINED" or "PF16 NOT DEFINED" in the SMOUT stream
- Moves the PASSWD window so that it displays the message

The commands that issue the message and move the PASSWD window over the SMOUT stream are:

```
CHANGE.PFK 4 'PFK 4 NOT DEFINED' SMOUT;
              CHANGE.WINDOW PASSWD VIEW(SMOUT)
```

PF5/17 PF5/17 is the find/repeat find key. It does the following:

- Locks the default window by issuing the SCROLL command specifying a value of 0.
- Issues the FIND.BACKWARD command substituting the information typed on the screen as the character string to be found. If you press PF5/17 without typing anything on the screen, the previous find command is repeated.
- Sets the terminal keyboard control to unlock after 5 seconds.
- Scrolls the LVALUE window to display the word "LOCKED" on line 8 in the HEADER stream.

The commands that define PF5/17 are:

```
CHANGE.PFK 5 'SCROLL.BACKWARD 0;
             FIND.BACKWARD ''&*'';
             CHANGE TERMINAL CONTROL(5);
             SCROLL.ABSOLUTE 8 LVALUE' SMIN
             SUBSTITUTE(&)
```

PF6/18

PF6/18 changes the CURRENT window to display either the last two lines in the TSOOUT stream or the last two lines in the TSOIN stream. Initially, the window displays the last two lines in the TSOOUT stream. When you press PF6/18, it does the following:

- Saves all of the current PF key definitions
- Changes the CURRENT window to display the TSOIN stream
- Redefines itself so that the next time you press the key, the CURRENT window again displays the TSOOUT stream and the PF key definitions are restored

The commands that define PF6/18 are:

```
CHANGE.PFK 6 'SAVE.PFK;
             CHANGE.WINDOW CURRENT
             TARGET(TSOIN) VIEW(TSOIN);
             CHANGE.PFK 6 ''CHANGE.WINDOW
             CURRENT TARGET(TSOIN)
             VIEW(TSOOUT);
             RESTORE.PFK '' SMIN ' SMIN
```

PF7/19

PF7/19 does the following:

- Scrolls the default window backward a half-page over the stream it is displaying. (As part of the SCROLL command, the Session Manager locks the window.)
- Unlocks the keyboard
- Scrolls the LVALUE window to display the word "LOCKED" on line 8 in the HEADER stream

The commands that define PF7/19 are:

```
CHANGE.PFK 7 'SCROLL.BACKWARD AMOUNT(HALF);
               CHANGE.TERMINAL CNTL(0);
               SCROLL.ABSOLUTE 8 LVALUE' SMIN
```

PF8/20

PF8/20 does the following:

- Scrolls the default window forward a half-page over the stream it is displaying. (As part of the SCROLL command, the Session Manager locks the window.)
- Unlocks the keyboard
- Scrolls the LVALUE window to display the word "LOCKED" on line 8 in the HEADER stream

The commands that define PF8/20 are:

```
CHANGE.PFK 8 'SCROLL.FORWARD AMOUNT(HALF);
               CHANGE.TERMINAL CNTL(0);
               SCROLL.ABSOLUTE 8 LVALUE' SMIN
```

PF9/21

PF9/21 does the following:

- Scrolls the default window to the oldest information in the stream it is displaying. (As part of the SCROLL command, the Session Manager locks the window.)
- Unlocks the terminal keyboard so that additional PF keys can be used

- Scrolls the LVALUE window to display the word "LOCKED" on line 8 in the HEADER stream

The commands that define PF9/21 are:

```
CHANGE.PFK 9 'SCROLL.OLDEST;
              CHANGE.TERMINAL CNTL(0);
              SCROLL.ABSOLUTE 8 LVALUE'
              SMIN
```

PF10/22 PF10/22 does the following:

- Scrolls the default window one half-page to the left side of the information in the stream it is displaying. (As part of the SCROLL command, the Session Manager locks the window.)
- Scrolls the LINE window one half-page to the left
- Unlocks the terminal keyboard
- Moves the LVALUE window over the word "LOCKED" on line 8 in the HEADER stream

The commands that define PF10/22 are:

```
CHANGE.PFK 10 'SCROLL.LEFT AMOUNT(HALF);
               SCROLL.LEFT LINE AMOUNT(HALF);
               CHANGE.TERMINAL CNTL(0);
               SCROLL.ABSOLUTE 8 LVALUE' SMIN
```

PF11/23 PF11/23 does the following:

- Scrolls the default window one half-page to the right side of the information in the stream it is displaying. (As part of the SCROLL command, the Session Manager locks the window.)
- Scrolls the LINE window one half-page to the right
- Unlocks the terminal keyboard
- Scrolls the LVALUE window over the word "LOCKED" on line 8 in the HEADER stream

The commands that define PF11/23 are:

```
CHANGE.PFK 11 'SCROLL.RIGHT AMOUNT(HALF);
               SCROLL.RIGHT LINE AMOUNT(HALF);
               CHANGE.TERMINAL CNTL(0);
               SCROLL.ABSOLUTE 8 LVALUE' SMIN
```

PF12/24 PF12/24 does the following:

- Scrolls the LINE window the maximum amount to the left (to column 1)
- Scrolls the default window the maximum amount to the left (to column 1)
- Moves the default window to the newest information in the stream and then unlocks the window
- Sets the maximum time that the keyboard can be locked to the last non-zero value entered
- Scrolls the LVALUE window over the word "UNLOCKED" on line 7 in the HEADER stream

The commands that define PF12/24 are:

```
CHANGE.PFK 12 'SCROLL.LEFT LINE AMOUNT(MAX);
               SCROLL.LEFT AMOUNT(MAX);
               UNLOCK.NEWEST;
               CHANGE.TERMINAL CNTL(LAST);
               SCROLL.ABSOLUTE 7 LVALUE'
               SMIN
```

Controlling the Terminal Keyboard

In order for the Session Manager to update your display screen, TSO requires that the terminal keyboard remain locked. When the keyboard is locked, you cannot enter other commands. The CONTROL operand of the CHANGE.TERMINAL command lets you specify the number of seconds the keyboard is to be locked while a command is executing or while the Session Manager is updating the screen. CONTROL can be any integer from 0 to 999.

If you want to wait for each TSO command or program to execute and have its output displayed before entering the next command, set CONTROL to a very high value.

If you want to enter TSO or Session Manager commands before the currently executing command has completed, set CONTROL to 0. The keyboard unlocks immediately after the Session Manager rewrites the screen and you can then enter new commands. However, the Session Manager only updates the screen after you have pressed the ENTER key or any PF key. You must press the ENTER key or a PF key every time you want new output from a command or program displayed.

Specifying CONTROL as a small number (10 to 20) is a compromise between the two extremes. A number in that range allows most TSO commands time to execute and have their output displayed, yet does not cause the terminal keyboard to be locked for long periods of time.

In the IBM-supplied default environment, the keyboard locks for a maximum of 15 seconds. The scrolling PF keys reset the CONTROL timer to 0 so that the keyboard immediately unlocks when you press one of them. PF key 12 (or 24) resets the CONTROL operand to the last non-zero value it had. This is done to allow you to scroll back and review previous information in a stream while a TSO command is executing without waiting for the command or program to complete.

Making a Copy of Your Display Screen

You can get a copy of the information on your display screen by using the SNAPSHOT command. This command places a copy of the information into a particular stream. You can then print the information by using the SMCOPY command. (See "Using TSO Commands" for a description of SMCOPY.) An easy way to accomplish these steps is to define a PF key to issue the commands for you.

The following example illustrates how to define PF1 to issue the SNAPSHOT command placing a copy of the display screen in the EXTRA1 stream. The second command in the example defines PF4 to print the stream, and then erases the information in it so you can use the stream again. Notice that PF4 uses the PUT command to place the SMCOPY and SMPUT commands into the TSOIN stream.

```
CHANGE.PFK 1 'SNAPSHOT EXTRA1 FORMAT' SMIN
CHANGE.PFK 4 'PUT ''SMCOPY FROMSTREAM(EXTRA1) PRINT(A)
              PREFORMAT CAPS'' TSOIN; PUT ''SMPUT
              /CHANGE.STREAM EXTRA1 CLEAR/'' TSOIN' SMIN
```

Displaying Information About the Environment

Before you change the default environment, you might want to review how it is set up. You can display the characteristics of the default environment at your terminal by using the QUERY command. With this command, you can display the following information:

QUERY.FUNCTION
Displays the following information for each currently defined session function:

- The name of the function
- The input, output, and copy streams for the function
- Whether or not the audible alarm is to sound when information enters the input and output streams for the function
- The intensity at which the information in the output and copy streams is to be displayed

QUERY.PFK
Displays the following information for each currently defined program function (PF) key:

- The number of the PF key
- The name of the stream where the text-string is to be placed
- The identifier and delimiter characters used in the command
- The text-string used to define the key

QUERY.STREAMS
Displays the following information for each currently defined stream:

- The name of the stream
- The numbers of the top and bottom lines in the stream
- The maximum size of the stream (in lines and bytes)
- The number of lines and bytes currently being used by the stream
- The type of stream it is (input, output, or extra)
- Whether or not the audible alarm is to sound when information enters the stream

QUERY.TERMINAL
Displays the following information about the terminal environment:

- The control setting for the keyboard indicating the maximum time the keyboard is to remain locked
- Whether or not the audible alarm is to sound when the keyboard unlocks
- The current number of windows defined on the display screen

- The maximum number of windows that can be defined
- The name of the default window
- The location of the cursor
- The following information for each currently defined window:
  - The name of the window
  - The name of the stream that the window displays
  - Whether or not the window is locked
  - Whether or not you can enter data in the window
  - The name of the stream that is to receive the information entered in the window
  - The intensity at which the information in the stream is to be displayed
  - Whether or not the terminal's audible alarm is to sound when the Session Manager scrolls the window to display new information in the stream
  - How long the window (when unlocked) is to be held in place before it is scrolled towards the bottom of the stream
  - How many lines of the window's old position are to be repeated when the window scrolls to the new position
  - How much new information must enter the stream before the Session Manager updates the window

QUERY.WINDOWS

Displays the following information for each currently defined window:

- The name of the window.
- The starting location of the window on the display screen (in rows and columns).
- The size of the window (in lines and width).
- The name of the stream the window is displaying.
- The numbers of the top and bottom lines of the stream that the window is presently displaying.
- The numbers of the top and bottom lines of the stream that the window was displaying when it was last unlocked. These numbers are used when the UNLOCK.RESUME command is issued.

- The numbers of the top and bottom lines of the newest information in the stream.

Saving The Environment

To help you easily manipulate your environment, the Session Manager lets you save definitions on a stack, restore definitions from a stack, and reset the default environment via the RESET command.

Saving and Restoring Definitions

The Session Manager maintains three push-down LIFO (last in, first out) stacks. The three kinds of stacks are **PF key stack**, **screen stack**, and **window stack**. You can use these stacks to save the current program function (PF) key definitions, screen definitions, and window definitions. For example, you might want to save the default PF key definitions, redefine the PF keys as TSO TEST subcommands for use with TSO TEST, and then restore the previous definitions when you leave TEST.

To place definitions in a stack, use the SAVE command. To remove or restore the definitions from the stack, use the RESTORE command. At the beginning of your terminal session, the stacks are empty. Issuing the RESTORE command then has no effect. Once the first item has been saved on a stack, it remains as the first (bottom) item. It cannot be removed from the stack by issuing the RESTORE command although the definitions are restored to what they were. In effect, all items are removed except the first one.

PF Key Stack

When you issue the SAVE.PFK command, the Session Manager saves all current PF key definitions as the top element on the PF key stack.

Screen Stack

When you issue the SAVE.SCREEN command, the Session Manager saves the following information on the screen stack:

- A description of the display screen layout
- The location of the cursor
- The value indicating how long the keyboard is to remain locked while a command is executing (as set in the CHANGE.TERMINAL command) including the last non-zero control value entered
- The name of the default window
- The name and attributes of each currently defined window

Window Stack

When you issue the SAVE.WINDOW command, the Session Manager saves the following information for the default window or for the window whose name is specified on the command:

- The audible alarm setting for the window
- The amount of time the window (when unlocked) is held in place before it is scrolled toward the bottom of the stream
- The number of lines from the window's old position that will be repeated when the window moves
- Whether data can be entered in the window
- The name of the stream that is to receive the information typed in the window and the intensity at which the information is to be displayed
- How often the window is to move over the new information in the stream
- The name of the stream the window is displaying
- The top and bottom line numbers in the stream that the window is currently displaying
- Whether the window is locked or unlocked

Resetting the Default Environment

If you accidentally deleted any of the windows on your display screen, you can use the RESET command to get the default display screen back. The RESET command lets you restart your Session Manager display environment. This command removes the entries from all of the stacks and re-executes the commands that created the default environment. The information in the streams is not altered.

The RESET command should not be followed by any other Session Manager command on the same line. A command entered on the same line will be executed before the RESET command can reestablish the default screen layout.

Ending Session Manager Support

To end Session Manager support of your TSO session, use the END command. END erases all information in your streams. Your TSO session then continues with standard display support. If you want Session Manager support again, you must reissue the TSO LOGON command.

Session Manager Processing

Listed below are some informational notes on the execution of the Session Manager.

- Multiple TPUTs to a single line result in a single line for each TPUT. In addition, TSO does not guarantee the order of execution of multiple single line TPUTs generated from another address space (for example, the operator, another TSO user, or from a job you have submitted). Therefore, these multiple single line TPUTs might be out of order.

- All line mode output produced when executing a full screen program (in full screen mode) is logged in the TSO function's output stream.
- All cross memory messages received when executing a full screen program (in full screen mode) are logged in the MSG function output stream.

Using TSO Commands

You can use any of the TSO commands or line-oriented functions with the Session Manager. These commands can be found in *TSO Command Language Reference*. In addition, three TSO commands are provided for use with the Session Manager: SMCOPY, SMFIND, and SMPUT. You can enter these commands from the keyboard, using PF keys, or via CLISTs. The functions of these commands are:

Command	Function
SMCOPY	Copies all or part of a stream or data set into another stream or data set (that is, stream to stream, stream to data set, data set to stream, or data set to data set). The receiving data set can be a SYSOUT data set.
SMFIND	Locates a string of characters in a Session Manager stream.
SMPUT	Places a string of characters in a Session Manager stream.

Session Manager Commands

This section describes the functions and syntax of each Session Manager command. It includes:

- How to enter the commands.
- The general format and syntax rules for the commands.
- A description of the function and syntax for each command. The commands are described in alphabetical order.

Entering Session Manager Commands

You can enter Session Manager commands by:

- Pressing the CLEAR key and entering a command anywhere on the screen
- Pressing a program function (PF) key set up to issue a command
- Executing a TSO CLIST, which contains commands
- Defining the command as the text-string of the TSO SMPUT command

Regardless of how you enter the commands, the following rules apply:

- You can enter multiple Session Manager commands on one line provided you separate them with a semicolon (;). The number of characters on any one line cannot exceed 512. When multiple commands are entered on a line, an error in one command does not prevent the remaining commands from executing.
- In order for a Session Manager command to execute, it must be placed in the SMIN stream.
- Any change you make to the definitions of the windows, cursor, PF keys, session functions, streams, or the terminal, remain in effect for your terminal session. You can place these definitions in a CLIST to be executed each time you log on.

Command Format

A Session Manager command consists of a **command name**, usually followed by a **command modifier** and one or more **operands.**

A command name is typically a familiar English word that describes the function of the command. Some command names are followed by command modifiers, which qualify the action of the command name.

Operands provide specific information about how an operation is to be performed. The two types of operands used with Session Manager commands are **positional** and **keyword.**

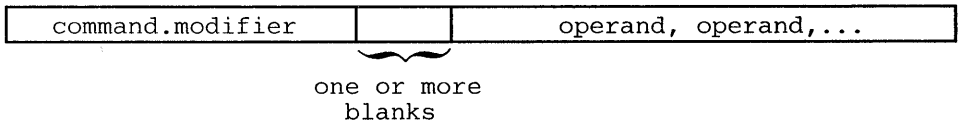


Figure 6. Session Manager Command Format

You must separate the command name and command modifier with a period or one or more blanks. Also separate the command or command modifier from the first operand by one or more blanks.

The command descriptions include some operands in lowercase letters and some in uppercase letters. If the operand is in lowercase letters, you must substitute a specific value for the letters. For example, in the command

```
CHANGE.PFK pfk-number ...
```

you must replace 'pfk-number' with the number of the program function (PF) key to be changed.

```
CHANGE.PFK 1 ....
```

Lowercase operands are **positional** operands because they follow the command names or modifiers in a prescribed order.

The operands in uppercase letters are **keyword** operands. You must type those operands as shown. For example:

```
        {YES}
ALARM( {NO } )

CONTROL(seconds)
```

Both of the preceding keyword operands have subfield values enclosed in parentheses. You must type the keyword (or its abbreviation), a left parenthesis, and then the subfield value. You can omit the closing parenthesis if it is the last character of the command.

The ALARM keyword shows the two possible subfield values you can enter within the braces. Braces indicate that the two values are mutually exclusive; enter one but not both.

The CONTROL operand shows the subfield value in lowercase letters. Therefore, you are to substitute a value for the lowercase name.

Defaults

To make the commands easier to enter, certain operands default to a specific value. If the default value is the value you want to use, you do not have to enter the operand. The default values are underlined in the syntax description for each command.

Many Session Manager commands refer to a **default window**. The default window is the MAIN window or the window you have assigned as the default window via the CHANGE.TERMINAL command.

Abbreviations

You can abbreviate nearly all Session Manager command names, modifiers, and keyword operands. These abbreviations can be as short as possible, while still providing uniqueness among them. For example, the minimum abbreviation for the DELETE command is:

```
DEL
```

DEL distinguishes DELETE from the DEFINE command.

The exceptions to this rule are the following frequently used commands:

```
DEFINE     D
SCROLL     S
RESTORE    R
```

You can also abbreviate any keyword operand. For example, the keyword operands of the CHANGE.WINDOW command and their minimum abbreviations are:

```
ALARM      A
HOLD       H
OVERLAP    O
PROTECT    P
TARGET     T
UPDATE     U
VIEW       V
```

The Session Manager also accepts the following commonly used abbreviations:

```
CONTROL    CNTL
FORMAT     FMT
```

Session Manager Command Syntax

The following figure summarizes the notation used to describe the Session Manager commands.

Notation	Meaning	Example	
		Book Syntax	Sample Entry
Asterisk *	Means all	DELETE.WINDOW { window-name / * }	delete.window *
Blank	Used to separate parameters of a command. You can use a comma instead.	CHANGE.STREAM stream-name [CLEAR]	change.stream tsoin clear
Brackets	Indicates that the operands within the brackets are optional and can be omitted. Do not type the brackets when entering the command.	CHANGE.TERMINAL [DEFAULT(window-name)]	change.terminal default(main)
Braces	Indicates that you must enter one of the items. You cannot enter more than one. Do not type the braces when entering the command.	CHANGE.FUNCTION { MSG / SM / TSO }	change.function tso
Comma	Used to separate parameters of a command. You can use blanks instead.	PUT text-string stream-name	put time,tsoin
Hyphen	Joins lowercase words to form a single variable. Do not type the hyphen when entering the command.	SNAPSHOT stream-name	snapshot tsoin
lowercase	Represents variables for which you are to substitute specific information. You can enter the information in upper or lowercase letters.	FIND.BACKWARD text-string	find.backward listds
Parenthesis	Must be entered as shown. You do not have to type the closing parenthesis if it is the last character in the command.	CHANGE.FUNCTION SM [INPUT(stream-name)]	change.function sm input(smin)
Period	Used to separate the command and command modifier. You can use blanks instead.	CHANGE.CURSOR	change.cursor
Quotes	Separates text-strings. A single quote in the character string must be entered as two adjacent quotes. If the text-string contains no imbedded blanks, commas, or parenthesis, enclosing quotes are not needed and single quotes in the text do not need to be repeated. If the text-string is entered without quotes, it is translated to uppercase.	PUT text-string stream-name	put 'the job''s done' tsoout

Figure 7 (Part 1 of 2). Session Manager Command Syntax

Notation	Meaning	Example	
		Book Syntax	**Sample Entry**
Under-scoring	Indicates that the keyword or value is the default for the operand.	SMCOPY [ASIS / CAPS / NOTRAN]	SMCOPY
Uppercase	The parameter must be spelled as shown (or its abbreviation). You can enter the parameter in either upper or lower case.	RESET	reset

Figure 7 (Part 2 of 2). Session Manager Command Syntax

Session Manager Command Summary

Figure 8 summarizes the Session Manager commands and their functions.

Command	Use This Command To:
CHANGE.CURSOR	Change the permanent or temporary location of the cursor.
CHANGE.FUNCTION	Change whether or not the terminal's audible alarm is to sound when information enters an input or output stream. Specify whether or not information from an input stream is to be copied to an output stream and the intensity at which the information is to be displayed. Specify the input stream for a session function. Specify the output stream for a session function and the intensity at which the information is to be displayed.
CHANGE.MODE	Indicate whether you want to run under VS/APL or under the Session Manager.
CHANGE.PFK	Change the definition of a program function (PF) key.
CHANGE.STREAM	Specify whether or not the terminal's audible alarm is to sound when information enters a given stream. Erase all of the information in a given stream.
CHANGE.TERMINAL	Specify whether or not the terminal's audible alarm is to sound when the keyboard unlocks. Specify the maximum time the keyboard is to be locked while a command is executing. Change the default window.
CHANGE.WINDOW	Specify whether or not the terminal's audible alarm is to sound when the Session Manager scrolls a window to display new information. Specify how long a window is to be held in place before the Session Manager scrolls it. Specify how many lines of a window's old position are to be repeated when the window scrolls to a new position. Specify whether or not information can be entered in a window. Indicate the name of the stream that is to receive the information in a window and the intensity at which the information is to be displayed. Indicate how much new information must enter a stream before the window scrolls to display it. Specify the name of the stream a window is to display.
DEFINE.WINDOW	Define a new window on the display screen.
DELETE.WINDOW	Delete a window from the display screen.
END	End Session Manager support of your TSO session.
FIND	Search for a text-string in a stream that is currently displayed by a window. Find the number of the top line being displayed by a window.
PUT	Place a text-string in a stream and indicate the intensity at which the text-string is to be displayed.

Figure 8 (Part 1 of 2). Summary of the Session Manager Commands

Command	Use This Command To:
QUERY	Display information about: - TSO, SM, and MSG functions - Program function (PF) keys - Streams - Terminal - Windows
RESET	Restart the Session Manager display environment.
RESTORE	Restore the definitions of the following, which were saved via the SAVE command. - Program function (PF) keys - Screen layout - Windows
SAVE	Save the definitions of the following: - Program function (PF) keys - Screen layout - Windows
SCROLL	Move a window over a stream.
SNAPSHOT	Copy a display screen of information into a stream.
UNLOCK	Unlock a window.

Figure 8 (Part 2 of 2). Summary of the Session Manager Commands

CHANGE.CURSOR

Purpose:

Use the CHANGE.CURSOR command to change the location of the cursor on the display screen. You can establish a **permanent** or **temporary** location for the cursor.

If you define a permanent location, the cursor returns to that location each time you press a program function (PF) key, the ENTER key, the attention (PA1) key, the CLEAR key or the cancel (PA2) key. If you define a temporary location, the cursor moves to and remains at that location until the next keyboard entry. After the keyboard entry, the cursor moves to the permanent location.

Syntax:

```
{CHANGE.CURSOR}  [row column]  [window-name     ]  [TEMPORARY]
{C       C    }  [ 1    1   ]  [default window ]
```

Description:

row column
The location in the specified window where the cursor is to go. The 'row' and 'column' numbers are relative to those in the window, not the entire display screen. If you specify a number for 'row' that is greater than the number of lines in the window, the Session Manager uses the last row in the window. If you specify a number for 'column' that is greater than the number of columns in the window, the Session Manager adjusts the column value to the number of columns in the window minus one. If you specify 0 or a negative number for 'row' or 'column', the Session Manager places the cursor in the first row and column in the window.

window-name
The name of the window where the cursor is to be placed.

TEMPORARY
Specifies that this change to the cursor location is to be temporary. If both the row and column and window-name operands are omitted, and a temporary location for the cursor was previously set, the Session Manager moves the cursor to that location. If a temporary location was not previously set, the Session Manager moves the cursor to the upper left hand corner of the display screen.

Unless you specify TEMPORARY, the change is permanent.

Example 1:

Set the permanent location of the cursor to row 5, column 3 of the TEST window.

```
change.cursor 5 3 test
```

Example 2:

Set the temporary location of the cursor to row 2, column 1 of the ENTRY window.

```
change.cursor 2 1 entry temporary
```

Example 3:

Set the cursor to the temporary location that was set in a previous CHANGE.CURSOR command.

```
change.cursor temporary
```

Example 4:

Change the permanent location of the cursor to row 1, column 1 of the default window.

```
change.cursor
```

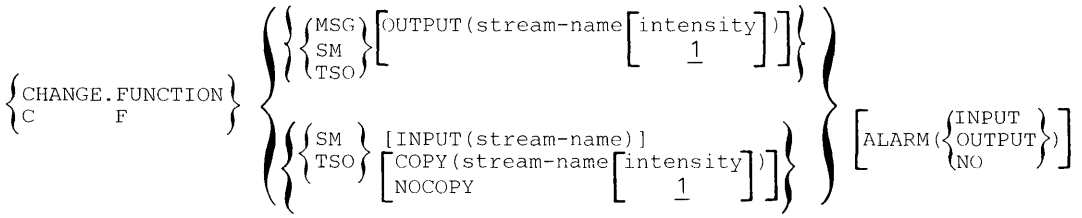
CHANGE.FUNCTION

Purpose:

Use the CHANGE.FUNCTION command to:

- Change whether or not the terminal's audible alarm is to sound when information enters an input or output stream
- Specify whether or not information from an input stream is to be copied to an output stream and the intensity at which the information is to be displayed
- Specify the input stream for the TSO, Session Manager (SM), or message (MSG) functions
- Specify the output stream for a session function and the intensity at which the information is to be displayed

Syntax:



Description:

MSG

Requests that the change apply to the message (MSG) function. The MSG function represents the messages from other TSO users, the operator, and background jobs.

SM

Requests that the change apply to the Session Manager (SM) function. The SM function represents work related to the Session Manager.

TSO

Requests that the change apply to the TSO function. The TSO function represents work related to TSO.

OUTPUT

stream-name

The name of the output stream for the specified function.

intensity

Specifies the brightness at which the information is to be displayed in the output stream. The valid values are:

1 The information is to be displayed at normal intensity.
2 The information is to be highlighted.

INPUT(stream-name)
The name of the input stream for the specified function.

COPY
Requests that the Session Manager copy the input stream for this function into an output stream.

stream-name
The name of the output stream that is to contain a copy of the information from the input stream.

intensity
Specifies the brightness at which the copied information is be displayed. The valid values are:

0 The copied information is not to be displayed. You can see the line that the information occupies, but the information itself is invisible.

1 The copied information is to be displayed at normal intensity.

2 The copied information is to be highlighted.

NOCOPY
Specifies that the Session Manager is not to copy the information from the input stream into an output stream.

ALARM
Specifies whether or not the terminal's audible alarm is to sound when information enters a stream. The stream does not have to be currently displayed for the alarm to sound. You can set ALARM to sound for either the input stream or the output stream, but not both.

Note: If your terminal does not have an audible alarm, the Session Manager still accepts this operand. It has no way of knowing whether or not your terminal has an audible alarm.

INPUT
Specifies that the audible alarm is to sound when a line of information enters the input stream for the specified function.

OUTPUT
Specifies that the audible alarm is to sound when the a line of information enters the output stream for the specified function.

NO
Specifies that the audible alarm is not to sound when information is added to any of the function streams.

Example 1:

Define the MSG function so that all messages are highlighted in the TSOOUT stream and the terminal's audible alarm sounds when you receive a message.

```
change.function msg output(tsoout 2) alarm(output)
```

Example 2:

Set the Session Manager function to get commands from the SMIN stream and place the output from these commands in the SMOUT stream highlighted. The information in the SMIN stream is not to be copied to the SMOUT stream.

```
change.function sm input(smin) output(smout 2) nocopy
```

Example 3:

Set the SM function to copy the SMIN stream into the SMOUT stream.

```
change.function sm copy(smout)
```

Example 4:

Set the TSO function to get its input from the TSOIN stream and highlight its output in the TSOOUT stream. The information in the TSOIN stream is to be copied into the TSOOUT stream.

```
change.function tso input(tsoin) output(tsoout 2)
  copy(tsoout)
```

CHANGE.MODE

Purpose

Use the CHANGE.MODE command to indicate whether you want to run under VS/APL or the Session Manager.

Syntax:

```
{CHANGE.MODE}  {APL}
{C       M  }  {SM }
```

Description:

APL

Indicates that you are running VS/APL and you want the Session Manager to provide any additional functions that were designed specifically to enhance the interface between the Session Manager and VS/APL.

Note: In order to use a VS/APL program function (PF) key definition, first make the corresponding Session Manager PF key definition null. To find out how to make a PF key null, refer to the CHANGE.PFK command.

SM

Indicates that you are running under the Session Manager.

Example:

Change the mode to run under VS/APL.

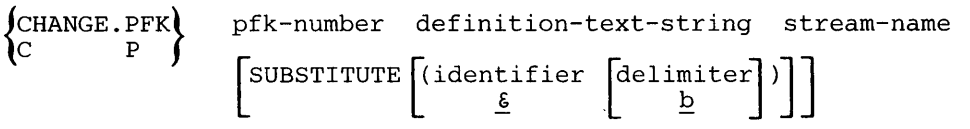
```
change.mode apl
```

CHANGE.PFK

Purpose:

Use the CHANGE.PFK key to change the definition of a program function (PF) key. You can define a PF key to issue one or more Session Manager commands, TSO commands, input to an application program, or any other string of characters.

Syntax:



Description:

pfk-number
The number of the PF key to be changed.

Note: If you specify a number that does not exist on your terminal, the Session Manager still accepts this operand. It has no way of knowing how many PF keys you have.

definition-text-string
The string of characters that are to be placed in the specified stream. If the text-string contains lowercase letters, blanks, commas, or parentheses, enclose it in single quotes. A single quote mark in the text-string must be represented as two adjacent quotes.

If there are no blanks, commas, or parentheses in the text-string, you can omit the enclosing quotes. If you omit the quotes, however, the Session Manager translates the text-string to uppercase letters. When using the CHANGE.PFK command in a CLIST, the Session Manager always stores the text-string in uppercase letters, even if it is enclosed in quotes.

If you enter more than one command for the text-string, separate them with a semicolon (;).

If you want to use a PF key defined under some other 3270 application (for example, VS/APL), first define the definition-text-string for the PF key as null to the Session Manager. The PF key can then be passed back to the application. To specify a null PF key, define the definition-text-string as two adjacent quotes ('').

stream-name
The name of the stream where the text-string is to be placed.

SUBSTITUTE

Specifies that the information read from the screen is to be substituted into the 'definition-text-string', replacing the symbolic arguments.

identifier

Identifies the symbolic argument that is to be replaced. Any character (except a blank or comma) can be used as the identifier. If the identifier character appears elsewhere in the definition-text-string, it must be doubled.

delimiter

Separates the information on the screen that is to be substituted into the text-string. One or more blanks are treated as a single delimiter. The delimiter can be any character except a comma.

Example 1:

Change PF1 to place the TSO 'time' command in the TSOIN stream where it will be executed.

```
change.pfk 1 'time' tsoin
```

Example 2:

Change PF12 to issue the QUERY.TERMINAL command. Direct the output to the EXTRA1 stream and cause the default window to display that stream.

```
change.pfk 12 'query.terminal extra1;change.window
  view(extra1)' smin
```

Example 3:

Change PF2 to issue the TSO LISTDS command. Each line typed just before the key is pressed is to be substituted as the data set name operand for the command.

```
change.pfk 2 'listds &1..* members' tsoin substitute
```

If you type the following on the screen:

```
test
sample
```

and pressed PF2, the following TSO commands are executed:

```
listds test.* members
listds sample.* members
```

CHANGE.STREAM

Purpose:

Use the CHANGE.STREAM command to:

- Change whether or not the terminal's audible alarm is to sound when information enters a stream
- Erase all of the information in a stream

(Use the QUERY.STREAMS command to display the names and attributes of all of the streams.)

Syntax:

```
{CHANGE.STREAM}   stream-name [ALARM({YES})] [CLEAR]
{C       S    }                 [      {NO }]
```

Description:

stream-name
The name of the stream to be changed.

ALARM({YES})
**   {NO }**
Specifies whether or not the terminal's audible alarm is to sound when information enters the stream. The stream does not have to be currently displayed for the alarm to sound.

Note: If your terminal does not have an audible alarm, the Session Manager still accepts this operand. It has no way of knowing whether or not your terminal has an alarm.

CLEAR
Erases all of the information in the stream. The stream itself is not erased.

Example 1:

Set the display terminal's audible alarm to sound when information goes into the EXTRA1 stream.

```
change.stream extra1 alarm(yes)
```

Example 2:

Erase all information in the TSOOUT stream.

```
change.stream tsoout clear
```

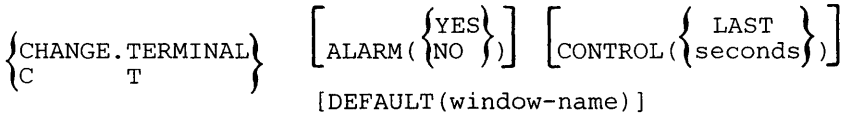
CHANGE.TERMINAL

Purpose:

Use the CHANGE.TERMINAL command to:

- Specify whether or not the audible alarm is to sound when the keyboard unlocks
- Indicate the maximum time the keyboard is to be locked while a command is executing
- Change the default window

Syntax:



Description:

ALARM({YES / NO})

Specifies whether or not the terminal's alarm is to sound when the keyboard unlocks. When the keyboard is unlocked, you can enter input.

Note: If your terminal does not have an audible alarm, the Session Manager still accepts this operand. It has no way of knowing whether or not your terminal has an alarm.

CONTROL

Specifies the maximum time, in seconds, that the keyboard is to remain locked. The Session Manager sets a timer to unlock the terminal keyboard when the time expires.

LAST

Specifies that the timer is to be set to the last non-zero value entered for the CONTROL keyword of this command.

seconds

Specifies that the keyboard is to be unlocked after the specified number of seconds has elapsed. "Seconds" must be an integer from 0 to 999.

DEFAULT(window-name)

The name of the window that you want to be the default window. This window serves as the default window for other Session Manager commands when a window name is entered with the command. The MAIN window is the IBM-supplied default window.

Example 1:

Set the terminal so that the keyboard will be locked for no more than 10 seconds.

```
change.terminal control(10)
```

Example 2:

Set the terminal so that each time the keyboard unlocks the audible alarm sounds.

```
change.terminal alarm(yes)
```

Example 3:

Set the terminal so that the keyboard will be locked for no more than 15 seconds and set the MAIN window as the default window.

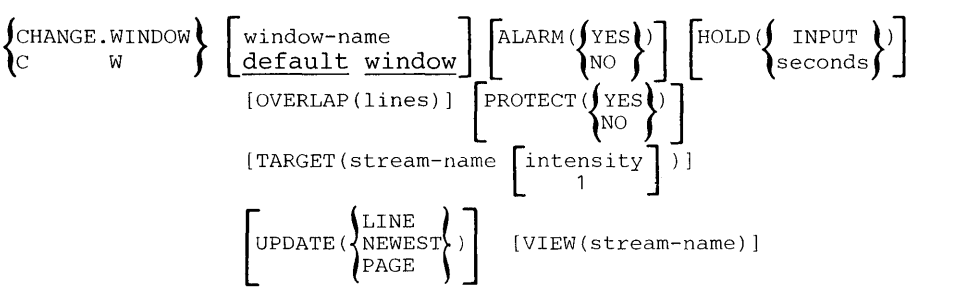
```
change.terminal control(15) default(main)
```

CHANGE.WINDOW

Purpose:

Use the CHANGE.WINDOW command to change the attributes of an existing window on the display screen. Use the QUERY.TERMINAL or QUERY.WINDOWS command to display the names and attributes of the currently defined windows.

Syntax:



Description:

window-name
The name of the window whose attributes are to be changed.

ALARM(YES NO)
Specifies whether or not the terminal's audible alarm is to sound when the Session Manager places new information in the stream.

Note: If your terminal does not have an audible alarm, the Session Manager still accepts this operand. It has no way of knowing whether or not your terminal has an alarm.

HOLD
Specifies how long the window (when unlocked) is to be held in place before it is scrolled towards the bottom of the stream.

INPUT
Specifies that the window (when unlocked) be held in place until you supply input by pressing the ENTER key or any program function (PF) key.

seconds
Specifies that the window (when unlocked) be held in place the specified number of seconds before it is scrolled toward the bottom of the stream. 'Seconds' must be an integer from 0 to 999.

During the time the window is held in place, the keyboard remains locked. The keyboard unlocks when the the time expires or when the window displays the bottom of the stream.

Note: The value specified on the CONTROL operand of the CHANGE.TERMINAL command overrides the value specified on this operand.

OVERLAP(lines)

Specifies how many lines of the window's old position are to be repeated when the window scrolls to a new position.

'Lines' must be an integer from 0 to 999. If you specify a value for 'lines' that is greater than or equal to the number of lines in the window, the Session Manager adjusts 'lines' to be the number of lines in the window minus one. Thus, at least the bottom line of the window's old position appears at the top of the window's new position.

PROTECT({YES})
{NO }

Specifies whether or not you can enter data in the window. You can enter data in an unprotected window only. If you try to enter data in a protected window, the keyboard locks.

TARGET

stream-name

The name of the stream that is to receive the information entered in the window.

intensity

Specifies the brightness at which the information in the stream is to be displayed. The valid values are:

0 The information in the stream is not to be displayed. You can see the line that the information occupies, but the information itself is invisible.

1 The information is to be displayed at normal intensity.

2 The information is to be highlighted.

UPDATE

Specifies how much new information must enter the stream before the Session Manager updates the window. The window only scrolls when it is unlocked.

LINE

Specifies that the window scroll sequentially towards the bottom of the stream. Thus, all of the new information is displayed as the window scrolls over the stream. When the window is full, it scrolls forward (repeating the number of lines specified by the OVERLAP operand) and the new information again starts to fill up the window.

NEWEST

Specifies that the window always display the newest information in the stream. When new information enters the stream, the window scrolls directly to the bottom of the stream. Some information in the stream might be skipped over. Thus, if a large amount of information is sent to the stream in a short period of time, only the last few lines (the number of lines in the window) are displayed.

PAGE

Specifies that the window scroll sequentially over the stream when there are enough new lines of information (minus the number of overlap lines) to fill the window. The window does not scroll to display the new information until enough additional information (a 'page' of information) enters the stream.

VIEW(stream-name)

The name of the stream the window is to display. Initially, the Session Manager places the window at the bottom of the stream and unlocks the window.

Example 1:

Change the TEST window to display the SMOUT stream.

```
change.window test view(smout)
```

Example 2:

Change the PASSWD window so that TSO passwords can be entered in non-display mode.

```
change.window passwd target(tsoin 0)
```

Example 3:

Change the default window so that input cannot be entered there.

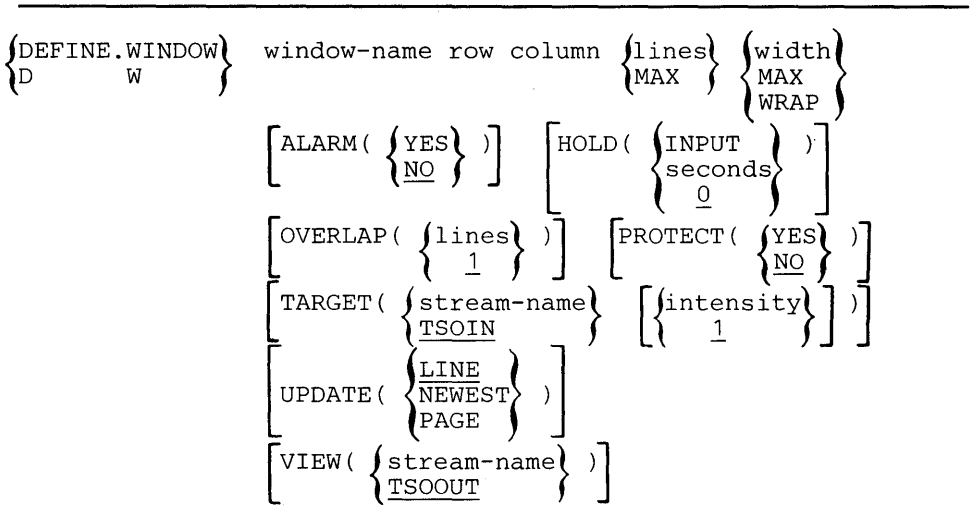
```
change.window protect(yes)
```

DEFINE.WINDOW

Purpose:

Use the DEFINE.WINDOW command to define a new window on the display screen.

Syntax:



Description:

window-name

The name of the window being defined. The name must be 1 to 8 alphanumeric characters, with the first character alphabetic.

row

Specifies which row of the display screen the top line of the window is to occupy. 'Row' must be an integer n or -n, where n can be any number from 1 to the number of rows on the display screen. An integer of -n is relative to the bottom of the screen. For example, a row value of -4 on a 24 line screen means that the top line of the window is to be row 21.

column

Specifies which column of the display screen the left side of the window is to occupy. 'Column' must be an integer n or -n, where n can be any number from 1 to the number of columns on the screen. An integer of -n is relative to the right side of the display screen. For example, a column value of -4 on an 80 column screen means that the left side of the window is to be in column 77.

lines

Specifies the number of lines in the window. 'Lines' must be an integer n or the character string 'MAX'. The value n can be any number from 1 to the number of lines on the display screen. 'MAX' indicates that the window is to consist of the remaining lines on the display screen or until a line is encountered that has already been defined as part of another window.

width

Specifies the number of character positions in each line of the window. 'Width' can be an integer n or the character string 'MAX' or 'WRAP'.

The value 'n' can be any number from 1 (or the number defined as the starting column) to the physical width of the display screen.

'MAX' indicates that the width of the window should be determined by the number of character positions available in the first line of the window (those not used by another window).

'WRAP' indicates that the width of the window is to start from the column value specified with this command and continue to either the beginning of the next window or to the last row and column of the screen. 'WRAP' can only be used when 'lines' is defined as 1.

Note: The first character position in a window is used as a terminal attribute byte and is protected. Therefore, a window defined with a width of 1 is useless.

ALARM({YES})
{NO }

Specifies whether or not the terminal's audible alarm is to sound when the Session Manager scrolls the window to display new information in the stream.

Note: If your terminal does not have an audible alarm, the Session Manager still accepts this operand. It has no way of knowing whether or not your terminal has an alarm.

HOLD

Specifies how long the window (when unlocked) is to be held in place before it is scrolled towards the bottom of the stream.

INPUT

Specifies that the window (when unlocked) be held in place until you supply input by pressing the ENTER key or any program function (PF) key.

seconds

Specifies that the window (when unlocked) be held in place the specified number of seconds before it is scrolled toward the bottom of the stream. 'Seconds' must be an integer from 0 to 999.

During the time the window is held in place, the keyboard remains locked. The keyboard unlocks when the time expires or when the window displays the bottom of the stream.

Note: The value specified on the CONTROL operand of the CHANGE.TERMINAL command overrides the value specified on this operand.

OVERLAP(lines)

Specifies how many lines of the window's old position are to be repeated when the window scrolls to the new position.

'Lines' must be an integer from 0 to 999. If you specify a value for 'lines' that is greater than or equal to the number of lines in the window, the Session Manager adjusts the value to be the number of lines in the window minus one. Thus, at least the bottom line of the window's old position always appears at the top of the window's new position.

PROTECT({YES})
{NO }

Specifies whether or not you can enter data in the window. You can enter data in an unprotected window only. If you try to enter data in a protected window, the keyboard locks.

TARGET

stream-name

The name of the stream that is to receive the information entered in the window.

intensity

Specifies the brightness at which the information in the stream is to be displayed. The valid values are:

0 The information in the stream is not to be displayed. You can see the line that the information occupies, but the information itself is invisible.

1 The information is to be displayed at normal intensity.

2 The information is to be highlighted.

UPDATE

Specifies how much new information must enter the stream before the Session Manager updates the window. The window only scrolls when it is unlocked.

LINE

Specifies that the window scroll sequentially towards the bottom of the stream. Thus, all of the new information is displayed as the window scrolls over the stream. When the window is full, it scrolls forward (repeating the number of lines specified by the OVERLAP operand) and the new information again starts to fill up the window.

NEWEST

Specifies that the window always display the newest information in the stream. When new information enters the stream, the window scrolls directly to the bottom of the stream. Some information in the stream might be skipped over. Thus, if a large amount of information is sent to the stream in a short period of time, only the last few lines (the number of lines in the window) are displayed.

PAGE

Specifies that the window scroll sequentially over the stream when there are enough new lines of information (minus the number of

overlap lines) to fill the window. The window does not scroll to display the new information until enough additional information (a 'page' of information) enters the stream.

VIEW(stream-name)
The name of the stream that the window is to display. Initially, the Session Manager places the window at the bottom of the stream and unlocks the window.

Example:

Note: The display screen for this example contains 24 lines and is 80 columns wide.

Create a screen layout having an output window occupying the top 22 lines of the screen with a character width of the entire screen and an input window that occupies the bottom two lines of the screen but is logically a single line. The output window is to display the TSOOUT stream and input window is to display the HEADER stream. All other attributes of the windows are to assume the default values.

```
define.window output 1 1 22 max
define.window input -2 1 1 wrap view(header)
```

DELETE.WINDOW

Purpose:

Use the DELETE.WINDOW command to delete a window from the display screen.

Syntax:

```
{DELETE.WINDOW}   {window-name}
{DEL     W    }   {*           }
```

Description:

window-name
The name of the window to be deleted.

*
Specifies that all of the windows on the display screen are to be deleted. When all windows are deleted, press the CLEAR key before entering commands from the keyboard.

Example 1:

Delete the TEST window.

```
delete.window test
```

Example 2:

Delete all of the windows on the display screen.

```
delete.window *
```

END

Purpose:

Use the END command to end Session Manager display support of your TSO session. If, after issuing this command, you want to have Session Manager support again, you must reissue the TSO LOGON command. The information in your streams is erased when you issue the END command.

Syntax:

```
END
```

FIND

Purpose:

Use the FIND command to search for a specific text-string in the stream that a window is displaying or to determine the number of the top line displayed in a window.

If the Session Manager finds the text-string, it scrolls the window so that the line containing the text-string is displayed on the top line of the window and the window is locked. If the text-string is not found, the Session Manager places a message in the Session Manager output stream. If you issue the FIND command with a null text-string (adjacent quote marks ''), the Session Manager searches for the previous text-string.

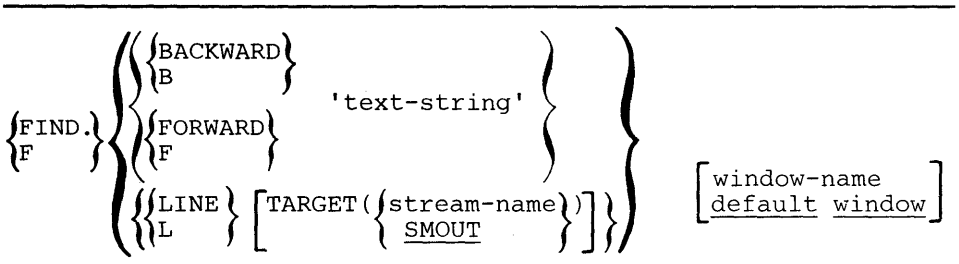
For the FIND.LINE command, the Session Manager writes the following message in the specified window:

```
ADF031I  window-name VIEWING LINE nnnnn
```

'Nnnnn' is the top line number in the window.

The FIND.LINE command is useful for finding a line number for the SCROLL.ABSOLUTE command, for copying a range of lines using the SMCOPY command, or for locating a text-string within a range of lines using the SMFIND command. The Session Manager does not lock the window when you use the FIND.LINE command.

Syntax:



Description:

BACKWARD
Causes the Session Manager to search for the specified text-string from the top line in the window on the display screen backward toward the top of the stream. The stream searched is the one displayed in either the default window or the window specified on the command.

FORWARD
Causes the Session Manager to search for the specified text-string from the current line on the display screen forward toward the bottom of the stream. The stream searched is the one displayed in either the default window or the window specified on the command.

LINE

Causes the Session Manager to find the number of the top line in the default window or the specified window and writes a message identifying the line number in the specified stream.

text-string

The string of characters to be searched for. If the text-string contains lowercase letters, blanks, commas, or parentheses, enclose it in single quotes. A single quote mark in the text-string must be represented as two adjacent quotes.

If there are no blanks, commas, or parentheses in the text-string, you can omit the enclosing quotes. If you omit the enclosing quotes, however, the Session Manager translates the text-string to uppercase letters before beginning the search. When the FIND command is used in a CLIST, the Session Manager always stores the text-string in uppercase letters, even if it is enclosed in quotes.

If you specify a null text-string, the Session Manager uses the last text-string you entered as the string of characters to search for. A null text-string is defined as two adjacent quotes (''').

TARGET(stream-name)

The name of the stream that is to contain the message produced by the FIND.LINE command.

window-name

For FIND.BACKWARD and FIND.FORWARD, window-name is the name of the window whose stream is to be searched.

For FIND.LINE, window-name is the name of the window whose top line number is to be found.

Example 1:

Assume that the default window is displaying the bottom of the TSOOUT stream. Find the last time the character string 'link' was issued.

```
find.backward 'link'
```

Example 2:

Assume that the TEST window is displaying the top of the TSOOUT stream. Find the first time you edited a data set named 'abc.asm'.

```
find.forward 'edit abc.asm' test
```

Now find the next occurrence of this same text-string.

```
find.forward '' test
```

PUT

Purpose:

Use the PUT command to place a text-string in a Session Manager stream. If you place the text-string in the TSOIN stream, it is sent to TSO to be executed as a TSO command. If you place the text-string in the SMIN stream, it is interpreted as a Session Manager command. The entire command cannot exceed 512 characters.

Syntax:

```
PUT     'text-string'  stream-name [INTENSITY( {intensity} )]
P                                  {    1    }
```

Description:

text-string
The string of characters to be placed in the stream. If the text-string contains lowercase letters, blanks, commas, or parentheses, it must be enclosed in single quote marks. A single quote mark in the text-string must be represented as two adjacent quotes.

If there are no blanks, commas, or parentheses in the text-string, you can omit the enclosing quotes. If you omit the enclosing the enclosing quotes, however, the Session Manager translates the text-string to uppercase letters before beginning the search. When the PUT command is used in a CLIST, the Session Manager always stores the text-string in uppercase letters, even if it is enclosed in quotes.

stream-name
The name of the stream where the text-string is to be placed. The Session Manager places the text-string at the bottom of this stream.

INTENSITY(intensity)
Specifies the brightness at which the text-string is to be displayed in the stream. The valid values are:

0 The text-string is not to be displayed. You can see the line that the text-string occupies, but the information itself is invisible.

1 The text-string is to be displayed at normal intensity.

2 The text-string is to be highlighted.

Example 1:

Place a comment in the TSOOUT stream and highlight it.

```
put 'this is a comment' tsoout intensity(2)
```

Example 2:

Use the PUT command in the definition-text-string of the CHANGE.PFK command. When pressed, PF3 is to issue the TSO TIME command and the Session Manager UNLOCK.NEWEST command. The commands are to be placed in the SMIN stream.

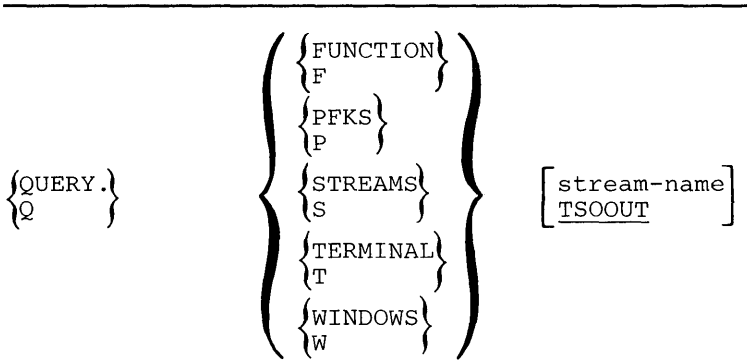
```
change.pfk 3 'put time tsoin;put ''unlock newest''
smin' smin
```

QUERY

Purpose:

Use the QUERY command to display information about the Session Manager functions, program function (PF) keys, streams, terminal, and windows.

Syntax:



Description:

FUNCTION

Displays the following information for each currently defined session function:

- The name of the function
- The input, output, and copy streams for the function
- Whether or not the audible alarm is to sound when information enters the input and output streams
- The intensity at which the information in the output and copy streams is displayed

PFKS

Displays the following information for each currently defined program function (PF) key:

- The number of the PF key
- The name of the stream where the text-string is to be placed
- The identifier and delimiter characters for the PF keys
- The text-string used to define the key

STREAMS

Displays the following information for each currently defined stream:

- The name of the stream
- The numbers of the top and bottom lines in the stream
- The maximum size of the stream (in lines and bytes)

- The number of lines and bytes currently used by the stream
- The type of stream it is (input, output, or extra)
- Whether or not the audible alarm is to sound when information enters the stream

TERMINAL

Displays the following information about the terminal environment:

- The control setting for the keyboard indicating the maximum time the keyboard is to remain locked
- Whether or not the audible alarm is to sound when the keyboard unlocks
- The current number of windows defined on the display screen
- The maximum number of windows that can be defined
- The name of the default window
- The permanent location of the cursor
- The following information for each currently defined window:
  - The name of the window
  - The name of the stream that the window displays
  - Whether or not the window is locked
  - Whether or not you can enter data in the window
  - The name of the stream that is to receive the information entered in the window
  - The intensity at which the information in the stream is to be displayed
  - Whether or not the terminal's audible alarm is to sound when the Session Manager scrolls the window to display new information in the stream
  - How long the window (when unlocked) is to be held in place before it is scrolled towards the bottom of the stream
  - How many lines of the window's old position are to be repeated when the window scrolls to the new position
  - How much new information must enter the stream before the Session Manager updates the window

WINDOWS

Displays the following information for each currently defined window:

- The name of the window.
- The starting location of the window on the display screen (in rows and columns).
- The size of the window (in lines and width).
- The name of the stream that the window displays.
- The numbers of the top and bottom lines of the stream that the window is currently displaying.
- The numbers of the top and bottom lines of the stream that the window was displaying when it was last unlocked. (These numbers are used when the UNLOCK.RESUME command is issued.)
- The numbers of the top and bottom lines of the newest information in the stream that the window is currently displaying.

stream-name

The name of the stream where the output from the command is to be placed. The output is in table format.

Example 1:

Display the information for all session functions.

```
query.function
```

The output from the command is as follows:

```
FUNCTION      INPUT              OUTPUT              COPY
  NAME      STREAM   ALARM   STREAM   INT ALARM   STREAM   INT
TSO         TSOIN      N     TSOOUT     1     N   TSOOUT     2
SM          SMIN       N     SMOUT      2     Y    SMOUT     0
MSG          *NONE*          TSOOUT     2     Y     *NONE*
```

Example 2:

Display the information for all streams defined.

```
query.streams
```

The output from the command is as follows:

```
 STREAM     LINE  RANGE    MAXIMUM  SIZE        USED
  NAME      LOW    HIGH    LINES    BYTES   LINES  BYTES  TYPE   ALARM
TSOIN         1       4      305     8192       4     82  INPUT    N
TSOOUT        1      47     4005   147456      47   4678  OUTPUT   N
EXTRA1        1       1      405    32768       1     38  OUTPUT   N
SMOUT         1      61      155     4096      61    424  OUTPUT   N
HEADER        1       9       55     1024       9    349  EXTRA    N
EXTRA3        1       2      105     1024       2     47  EXTRA    N
EXTRA2        1       1      105     1024       1     38  EXTRA    N
MESSAGE       1       1       55     1024       1     39  OUTPUT   Y
SMIN          1      59      305     8192      59   6192  INPUT    N
QUERY COMPLETE
```

Example 3:

Display all information related to the terminal.

```
query.terminal
```

The output from the command is as follows:

```
KEYBOARD    CNTL     ALARM
              15       N
WINDOWS     CURRENT # MAXIMUM #        DEFAULT WINDOW     CURSOR POSITION
               11         25              MAIN             ENTRY      1     1
  NAME     VIEW   LOCKED   PROT   TARGET   INTENSITY   ALARM   HOLD   OVERLAP   UPDATE
LINE     HEADER     Y        Y    TSOIN        1          N       0        0      N
STITLE   HEADER     Y        Y    TSOIN        1          N       0        0      N
SVALUE   EXTRA3     N        Y    EXTRA3       1          N       0        0      N
LTITLE   HEADER     Y        Y    TSOIN        1          N       0        0      N
LVALUE   HEADER     Y        Y    TSOIN        1          N       0        0      N
VLINE    HEADER     Y        Y    TSOIN        1          N       0        0      N
PASSWD   SMOUT      N        N    TSOIN        0          N       0        0      N
CURRENT  TSOOUT     N        N    TSOIN        1          N       0        0      N
TENTRY   HEADER     Y        Y    TSOIN        1          N       0        0      N
ENTRY    HEADER     Y        N    TSOIN        1          N       0        0      N
MAIN     TSOOUT     N        N    TSOIN        1          N       I        9      L
QUERY COMPLETE
```

Example 4:

Display the information for all windows defined:

```
query.windows
```

The output from the command is as follows:

```
                                                  PRESENT        RESUME        NEWEST
                                          VIEWING  TOP  BOTTOM  TOP  BOTTOM  TOP  BOTTOM
 WINDOW   ROW  COL  LINES  WIDTH  STREAM   LINE  LINE  LINE  LINE  LINE  LINE
LINE       20    1      1     80  HEADER     2     2     1     1    10    10
STITLE     21   63      1     12  HEADER     6     6     1     1    10    10
SVALUE     21   75      1      6  EXTRA3     2     2     1     1     2     2
LTITLE     22   63      1      9  HEADER     9     9     1     1    10    10
LVALUE     22   72      1      9  HEADER     7     7     1     1    10    10
VLINE      24   41      1      2  HEADER     4     4     1     1    10    10
PASSWD     24   43      1     38  SMOUT     57    57     1     1    10    10
CURRENT    21    1      2     62  TSOOUT    36    37    36    37    46    47
TENTRY     23    1      1      5  HEADER     3     3     1     1    10    10
ENTRY      23    6      1    114  HEADER    10    10     1     1    10    10
MAIN        1    1     19     80  TSOOUT     1    19     1    19    32    50
QUERY COMPLETE
```

RESET

Purpose:

Use the RESET command to restart your Session Manager display environment. For example, if you accidentally deleted any of the windows on your display screen, use RESET to get the default display screen back. This command removes the entries from all of the stacks and re-executes the commands that created the default environment. The RESET command causes the HEADER stream to be cleared and then redefined containing only those lines needed in the default environment. In addition, the default scroll amount is placed in the EXTRA3 stream, thereby resetting the default scroll amount on your display screen. None of the other streams are altered.

The RESET command should not be followed by any other Session Manager command on the same line. A command entered on the same line executes before the RESET command can reestablish the default screen layout.

Syntax:

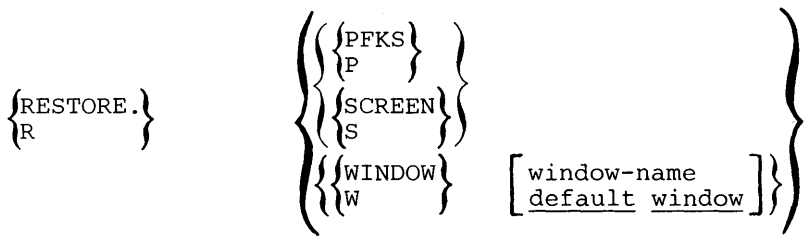
```
RESET
```

RESTORE

Purpose:

Use the RESTORE command to restore the definitions of the program function (PF) keys, screen layout, or windows previously saved via the SAVE command. If only one set of definitions exists on the stack, it is not removed. If more than one set of definitions has been saved, you must issue the RESTORE command as many times as you issued the SAVE command to get to the definitions you want.

Syntax:



Description:

PFKS
Specifies that the Session Manager is to restore the program function (PF) key definitions.

SCREEN
Specifies that the Session Manager is to restore the screen layout. The following items are included in each screen stack element:

- A description of the screen layout
- The location of the cursor
- The value indicating how long the keyboard is to remain locked while a command is executing (as set using the CHANGE.TERMINAL command)
- The name of the default window
- The name and attributes of each window

WINDOW
Specifies that the Session Manager is to restore the window definitions. Each window description element contains the following information:

- The audible alarm setting for the window (ALARM)
- The amount of time the window (when unlocked) is held in place before it is scrolled toward the bottom of the stream (HOLD)
- The number of lines from the window's old position that are to be repeated when the window scrolls (OVERLAP)
- Whether you can enter data in the window (PROTECT)

- The name of the stream that is to receive the information typed in the window and the intensity at which the information is to be displayed (TARGET)
- How often the window is to scroll over the new information in the stream (UPDATE)
- The name of the stream the window is displaying (VIEW)
- The numbers of the top and bottom lines in the stream that the window is currently displaying
- Whether the window is locked or unlocked

The location and size of the window are not restored.

window-name
The name of the window whose description is to be restored.

Example:

Restore the definition of the TEST window that was previously saved via the SAVE command.

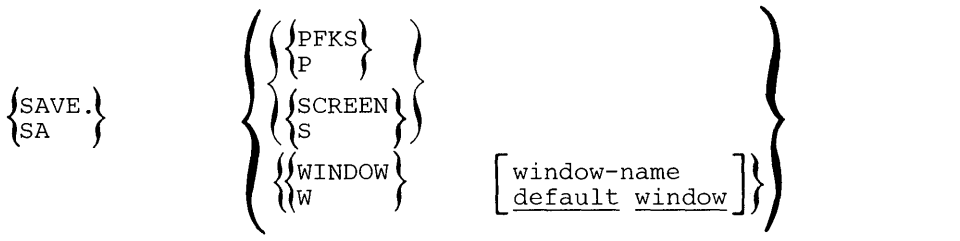
```
restore.window test
```

SAVE

Purpose:

Use the SAVE command to save the current definitions of program function (PF) keys, windows, and the screen layout. Later, you can restore these same definitions by using the RESTORE command.

Syntax:



Description:

PFKS
Specifies that all current PF key definitions are to be saved as the top element of the PF key stack.

SCREEN
Specifies that the current screen layout is to be saved as the top element on the screen stack. The following items are saved for each screen stack element:

- A description of the screen layout
- The location of the cursor
- The value indicating how long the keyboard is to remain locked while a command is executing (as set using the CHANGE.TERMINAL command)
- The name of the default window
- The name and attributes of each window

WINDOW
Requests that the definitions for the default window or the window specified on the command be saved as the top element of the window stack. Each window description element contains the following information:

- The audible alarm setting for the window (ALARM)
- The amount of time the window (when unlocked) is held in place before it is scrolled toward the bottom of the stream (HOLD)
- The number of lines from the window's old position that are to be repeated when the window scrolls (OVERLAP)
- Whether you can enter data in the window (PROTECT)

- The name of the stream that is to receive the information typed in the window and the intensity at which the information is to be displayed (TARGET)
- How often the window is to scroll over the new information in the stream (UPDATE)
- The name of the stream the window is displaying (VIEW)
- The numbers of the top and bottom lines in the stream that the window is currently displaying
- Whether the window is locked or unlocked

The location and size of the window are not saved.

window-name
The name of the window whose description is to be saved.

Example:

Save the definition of the TEST window on the window stack.

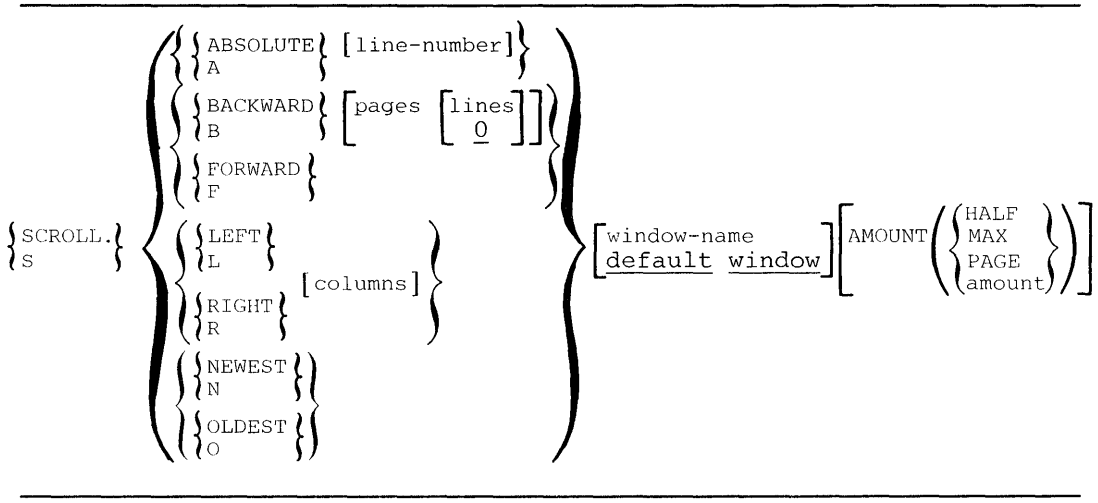
```
save.window test
```

SCROLL

Purpose:

Use the SCROLL command to move a window over a stream. After the Session Manager moves the window, it is locked in position. You can then move the window by using another scroll command or you can unlock the window by using the UNLOCK command.

Syntax:



Description:

ABSOLUTE
Specifies that the Session Manager is to scroll the window so that the identified line-number is the top line in the window. Use the QUERY, SMFIND, or FIND.LINE commands to find specific line numbers.

BACKWARD
Specifies that the Session Manager is to scroll the window backward toward the top of the stream.

FORWARD
Specifies that the Session Manager is to scroll the window forward toward the bottom of the stream.

LEFT
Specifies that the Session Manager is to scroll the window toward the left side of the stream. The limit for scrolling left is column 1 of the stream.

RIGHT
Specifies that the Session Manager is to scroll the window toward the right side of the stream. The limit for scrolling right is 32,768 column positions.

NEWEST
Specifies that the Session Manager is to scroll the window forward to the bottom of the stream.

OLDEST
Specifies that the Session Manager is to scroll the window backward to the top of the stream.

line-number
The number of the line you want to appear at the top of the window. If you enter a value for 'line-number' that is 0 or less, the Session Manager sets 'line-number' to 1. If you enter a value for 'line-number' that is greater than the highest line number in the stream, the Session Manager sets 'line-number' to the highest line number in the stream.

pages
default: 0 if the AMOUNT keyword is entered
1 if the AMOUNT keyword is not entered
Specifies how many pages to scroll the window. (A page is defined as the number of lines in the window.)

If you specify a value that would cause the window to scroll beyond the top or bottom of the stream, the Session Manager adjusts the value to place the window at the top or bottom (depending on the direction of the scrolling).

lines
Specifies how many lines to scroll the window.

If you specify a value that would cause the window to scroll beyond the top or bottom of the stream, the Session Manager adjusts the value to place the window at the top or bottom (depending on the direction of the scrolling).

columns
default: 0 if the AMOUNT keyword is entered
40 if the AMOUNT keyword is not entered
Specifies the number of columns to scroll the window.

If you specify a value that would cause the window to scroll beyond the left side of the stream, the Session Manager adjusts the value to place the window at column 1 of the stream. Values that would cause the window to scroll beyond column 32768 are adjusted to place the window at column 32768.

window-name
The name of the window to be scrolled.

AMOUNT
The amount the window is to be scrolled. AMOUNT can be specified instead of or in addition to, the operands 'columns', 'lines', or 'pages'. If you enter a value for one of the preceding operands and a value for AMOUNT, the Session Manager sums the two values and scrolls the window the resulting amount. The valid AMOUNT values are:

HALF
Specifies that the window is to be scrolled half a page. (For forward or backward scrolling, a page is defined as the number of lines in the window. For right or left scrolling, a page is defined as the number of columns in the window.)

MAX
Specifies that the window is to be scrolled the maximum amount. For forward scrolling, 'MAX' indicates to scroll to the bottom of the stream (equivalent to the SCROLL.NEWEST command). For backward scrolling, 'MAX' indicates to scroll to the top of the stream (equivalent to the SCROLL.OLDEST command).

PAGE
Specifies that the window is to be scrolled a full page.

amount
Specifies the number of lines or columns to scroll the window.

Example 1:

Scroll the default window to the oldest information.

```
scroll.oldest
```

or

```
scroll.backward amount(max)
```

Example 2:

Scroll the TEST window forward one page.

```
scroll.forward test
```

or

```
scroll.forward test amount(page)
```

Example 3:

Scroll the SAMPLE window backward 20 lines.

```
scroll.backward 0 20 sample
```

or

```
scroll.backward sample amount(20)
```

SNAPSHOT

Purpose:

Use the SNAPSHOT command to copy a display screen of information into a stream. You can then use the SMCOPY command to print the stream or copy it into a data set.

Syntax:

```
{SNAPSHOT}     stream-name     [FORMAT]
{SN      }
```

Description:

stream-name
The name of the stream where the information is to go.

FORMAT
Specifies that carriage control information is to be included in the copy of the information for printing on a system printer. Highlighted lines on the screen appear darker in the printed copy.

Note: If the copied information contains the carriage control characters, you must use the PREFORMAT operand of the SMCOPY command when printing it.

Example:

Place a copy of the display screen in the EXTRA1 stream.

```
snapshot extra1 format
```

Print the stream, using the SMCOPY command.

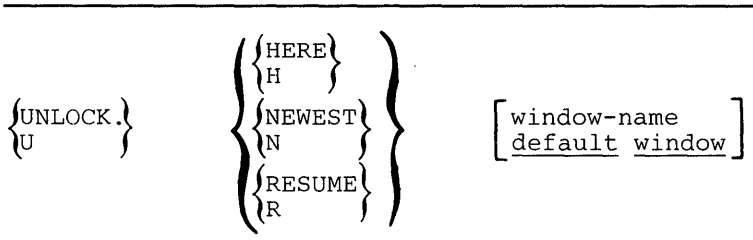
```
smcopy fromstream(extra1) preformat
```

UNLOCK

Purpose:

Use the UNLOCK command to unlock a window.

Syntax:



Description:

HERE
Causes the Session Manager to unlock the specified window at its current position.

NEWEST
Causes the Session Manager to display the newest information in the stream, then unlocks it.

RESUME
Causes the Session Manager to display the information the window was viewing before being locked, then unlocks the window.

window-name
The name of the window to be unlocked.

Example 1:

Move the default window to the bottom of the stream it is displaying and unlock it there.

```
unlock.newest
```

Example 2:

Unlock the SAMPLE window at its current position.

```
unlock.here sample
```

TSO Commands

Three TSO commands are provided for use with the Session Manager. You can enter them from the keyboard, using program function (PF) keys, or via CLISTs. These commands are TSO command processors, not Session Manager commands. For a complete description of the other TSO commands and of the syntax rules and conventions for TSO commands, refer to *TSO Command Language Reference*.

Command	Use This Command To:
SMCOPY	Copy all or part of a stream or data set to another stream or data set. The receiving data set can be a SYSOUT data set.
SMFIND	Locate a string of characters in a stream.
SMPUT	Place a string of characters in a stream.

Figure 9. Summary of the TSO Commands

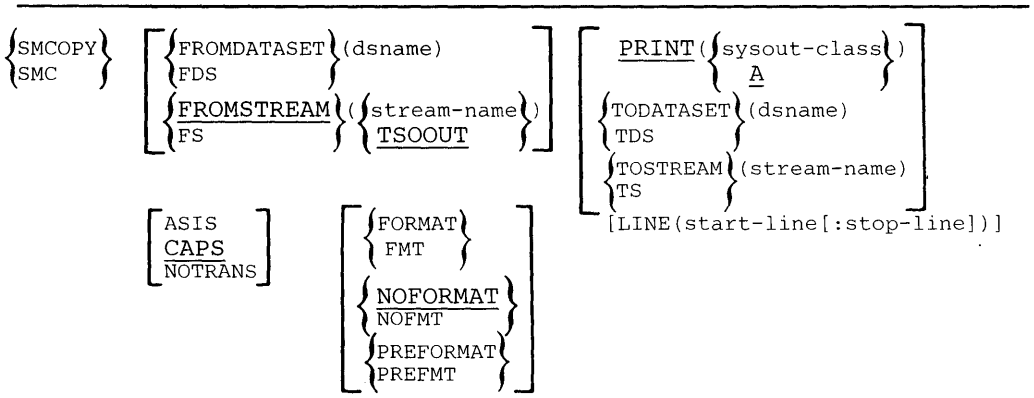
SMCOPY

Purpose:

Use the SMCOPY command to copy all or part of a stream or data set to another stream or data set (that is, stream to stream, stream to data set, data set to stream, or data set to data set).

Note: If the source and target of the copy request are both data sets, (SYSOUT or QSAM), you do not have to be logged on under the Session Manager to use the SMCOPY command.

Syntax:



Description:

FROMDATASET(dsname)
The name of the data set that contains the information to be copied. The data set must be a sequential data set or a member of a partitioned data set with either fixed or variable length records.

FROMSTREAM(stream-name)
The name of the input stream that contains the information to be copied.

PRINT(sysout-class)
Specifies that the information is to be copied to a SYSOUT data set of the specified SYSOUT class and printed on a system printer.

TODATASET(dsname)
The name of the data set into which the information is to be copied. The data set must be sequential or a member of a partitioned data set with either fixed or variable length records.

If the data set does not exist, the Session Manager allocates a new one. If the information is being copied from a data set (the FROMDATASET operand is specified), the attributes from this data set are used to create the new one. If the information is being copied from a stream (the FROMSTREAM operand is specified), the data set is allocated with the following attributes:

RECFM	VB or VBA if FORMAT or PREFORMAT is specified.
LRECL	255
BLKSIZE	3120

TOSTREAM(stream-name)
The name of the output stream for the copy operation.

ASIS
Specifies that the Session Manager is to leave lowercase letters as lowercase letters and translate the unprintable characters to blanks (hex '40').

Use the ASIS operand if the information is to be printed on a printer with a dual-case print train (TN or T11).

CAPS
Specifies that the Session Manager is to translate lowercase letters to uppercase and translate the unprintable characters to blanks (hex '40').

NOTRANS
Specifies that no translation is to occur.

FORMAT
Specifies that carriage control characters are to be placed in the copied information. If the information is being placed in a stream, the highlighted lines are highlighted in the stream.

If the information is being copied to a data set, the record format must be FBA VBA to indicate the presence of ASA control characters. If the data set is new, the Session Manager allocates it with a VBA record format.

FORMAT is ignored if FROMSTREAM is not specified.

NOFORMAT
Specifies that no control characters are to be placed in the copied information.

If the information is being copied from a data set, the data set must have a FB or VB record format. If the information is being copied from a stream to a data set, the data set must have a FB or VB record format. If the information is being copied from a data set to a data set, both data sets must have the same format (FB or VB). If the data set that the information is going into is new, the Session Manager allocates it with a VB record format (if it is being copied to a stream) or it is allocated with the same record format as the data set it is coming from (for a data set to data set copy operation).

PREFORMAT
Specifies that the source for the copy (stream or data set) already contains carriage control characters. Use this operand when the SNAPSHOT command was previously used to place information in a stream or data set.

If the information is being copied from a data set, the data set must have a FBA or VBA record format. If the information is being copied from a stream to a data set, the data set must have a FBA or VBA record format. If the information is being copied from a data set to a data set, both data sets must have the same format (FBA or VBA). If the data set that the information is going into is new, the Session Manager allocates it with a VBA record format (if it is being copied to a stream) or it is allocated with

the same record format as the data set it is coming from (for a data set to data set copy operation).

LINE(start-line:stop-line)
default: first line of the information
last line of the information
Specifies the range of lines to be copied.

If the information is being copied from a stream, you can find specific line numbers by using the QUERY, SMFIND, or FIND.LINE commands. If the information is being copied from a data set, 'LINE' represents records of the data set, not the line numbers within a numbered data set.

Example 1:

Copy the TSOOUT stream to the system printer, translating all lowercase letters to uppercase.

```
smcopy
```

Example 2:

Copy the member ZLOGON of the data set 'SYS1.CLIST' to the member ZLOGON of the data set TEST.CLIST.

```
smcopy fromdataset('sys1.clist(zlogon)')
       todataset(test.clist(zlogon))
```

Example 3:

Copy the data set containing TSO commands from the data set SAMPLE and place these commands in the TSOIN stream where they will be executed.

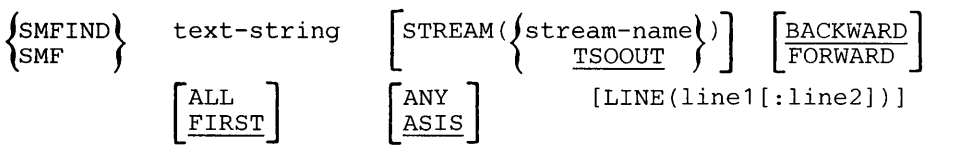
```
smcopy fromdataset('sample.commands.data')
       tostream(tsoin)
```

SMFIND

Purpose:

Use the SMFIND command to locate a string of characters in a stream. If the text string is found, the Session Manager displays the line number of the text string in the output stream for the TSO function (TSOOUT in the default environment) and puts the line number in register 15. If operating from a CLIST, you can access the line number from the CLIST variable '&LASTCC'.

Syntax:



Description:

text-string
The string of characters to be found. The text-string can be up to 256 characters in length and must be enclosed in delimiters that are not present in the text-string.

STREAM(stream-name)
The name of the stream to be searched.

BACKWARD
Specifies that the Session Manager is to search for the text-string from the current location backward toward the top of the stream.

FORWARD
Specifies that the Session Manager is to search for the text-string from the current location forward toward the bottom of the stream.

ALL
Specifies that the Session Manager is to find all occurrences of the text-string. The line number of each found text-string is displayed in the output stream for the TSO function. Register 15 (and the CLIST variable &LASTCC) contains the line number of the last occurrence of the text-string.

FIRST
Specifies that the Session Manager is to find only the first occurrence of the text-string. The Session Manager displays the line number of the found text-string in the output stream for the TSO function. It also places the number in register 15 and the CLIST variable &LASTCC.

ANY
Specifies that upper and lowercase differences are to be ignored when finding the text-string.

ASIS

Specifies that the Session Manager is to find an exact match of the entered text-string.

LINE(line1:line2)

Specifies the range of lines to be searched.

If only 'line1' is specified, the Session Manager searches from that line to the top or bottom of the stream depending on whether BACKWARD or FORWARD is specified.

If you specify a value for 'line1' or 'line2' that is not in the stream, the Session Manager uses the top or bottom line in the stream.

Upon completion, SMFIND returns the following in register 15:

line	A positive integer specifying the line number of the found text-string. The maximum value is 16,777,216, limited by &LASTCC, not by SMFIND.
0	Either: • The text-string was not found. • The specified stream does not exist. • The command was incorrectly specified and SMFIND was unable to prompt for correct information.

Example 1:

Find the next occurrence of 'time' in the TSOOUT stream.

```
smfind 'time' forward
```

Example 2:

Find the previous occurrence of 'time' in the TSOOUT stream.

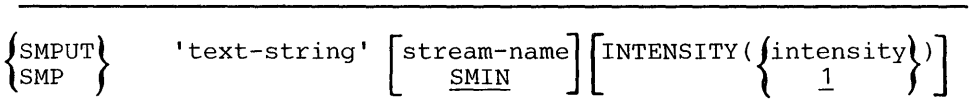
```
smfind 'time'
```

SMPUT

Purpose:

Use the SMPUT command to place a text-string in a stream. If you place the text-string in the TSOIN stream, it is interpreted as a TSO command. If you place the text-string in the SMIN stream, it is interpreted as a Session Manager command.

Syntax:



Description:

text-string
The string of characters to be placed in the stream. The text-string must be enclosed in delimiters that are not in the text-string. It can be up to 32,768 characters, excluding the delimiters. If the text-string is being sent to the SMIN stream, it can be up to 512 characters.

stream-name
The name of the stream where the text-string is to be placed.

Note: You must specify stream-name if you specify a value for INTENSITY.

INTENSITY
Specifies the brightness at which the information in the stream is to be displayed. The valid values are:

0 The information in the stream is not to be displayed. You can see the line that the information occupies, but the information itself is invisible.

1 The information is to be displayed at normal intensity.

2 The information is to be highlighted.

SMPUT returns the following return codes in register 15:

0 SMPUT executed successfully.

Note: If the text-string contained Session Manager or TSO commands, the zero return code does not indicate successful execution of those commands.

4 An error occurred while the Session Manager was placing the text-string in the stream.

8 The target stream was not found. The Session Manager might not be active.

12 An error occurred while the Session Manager was parsing the entered command.

Example

Place the TSO TIME command highlighted in the TSOIN stream.

```
smput 'time' tsoin intensity(2)
```

Using Command Procedures (CLISTs)

An easy way to execute a series of Session Manager or TSO commands is by using command procedures (CLISTs). A CLIST is an executable sequence of TSO and/or Session Manager commands, subcommands, or command procedure statements. You can use any TSO or Session Manager command in CLISTs. The same rules for entering Session Manager commands at your terminal apply when using a CLIST. (See the previous section "Entering Session Manager Commands.")

This section describes several examples of CLISTs and how they can be used to modify the default environment. The CLIST in the first example redefines a program function (PF) key using Session Manager commands. The remaining examples show CLISTs that split your display screen horizontally and vertically. The split screen CLISTs are located in SYS1.SAMPLIB so you do not have to type them in at your terminal. For a complete description of how to create, edit, invoke, and execute a CLIST, refer to *CLISTs: Application and Reference.*

Note: When you issue a WRITENR in a CLIST, the cursor is no longer positioned at the end of the text written to the screen. Instead, the cursor returns to the permanent cursor position to allow you to enter data.

Using CLISTs to Redefine Program Function (PF) Keys

Figure 10 is an example of a simple CLIST that redefines PF9 using Session Manager commands. After the CLIST is executed, pressing PF9 causes the MAIN window to display the MESSAGE stream so you can see the messages from other TSO users, the operator, or background jobs. Pressing PF9 again repositions the MAIN window at its previous location and restores the previous PF9 definition. The steps in the CLIST are as follows:

1. Uses the SMPUT command to place the commands in the SMIN stream.

2. Saves the definitions of the existing PF keys on the PF key stack.

3. Changes the MSG function so that any messages from other TSO users, the operator, or background jobs are displayed in the message stream at normal intensity. The terminal's audible alarm is to sound when a message enters the stream.

4. PF9 then does the following:

   - Saves the definitions of the existing PF keys on the PF key stack.

   - Saves the definition of the MAIN window on the window stack.

   - Changes the MAIN window to display the MESSAGE stream where the messages have been placed.

- Redefines itself to:
  - Restore the MAIN window to display the stream it was previously displaying.
  - Restore the PF key definitions. This resets PF9 to its previous definition.

```
/*********************************************************************/
/* THIS CLIST SETS UP A SESSION MANAGER PROGRAM FUNCTION KEY.        */
/* IT ASSUMES THE IBM-SUPPLIED DEFAULT SCREEN LAYOUT IS THE CURRENT  */
/* SCREEN LAYOUT WHEN IT IS INVOKED:                                 */
/* - THE "MSG" FUNCTION IS CHANGED TO SEND MESSAGES FROM OTHER TSO   */
/*   USERS TO THE "MESSAGE" STREAM, SOUNDING THE ALARM WHEN A        */
/*   MESSAGE IS RECEIVED.                                            */
/* - PF KEY 9 IS DEFINED TO MAKE THE "MAIN" WINDOW VIEW THE MESSAGE  */
/*   STREAM WHEN PRESSED. IF PRESSED AGAIN, PF 9 RETURNS THE "MAIN"  */
/*   WINDOW TO VIEWING THE STREAM IT WAS VIEWING WHEN PF 9 WAS FIRST */
/*   PRESSED.                                                        */
/* - THE EXISTING PF KEYS ARE SAVED BEFORE THE SCREEN IS MODIFIED.   */
/*********************************************************************/
                                                                  /* */
SMPUT /SAVE.PFK;+
       CHANGE.FUNCTION MSG OUTPUT(MESSAGE 1) ALARM(OUTPUT);+
       CHANGE.PFK 9 'SAVE.PFK;SAVE.WINDOW MAIN;+
                     CHANGE.WINDOW MAIN VIEW(MESSAGE);+
                     CHANGE.PFK 9 ''RESTORE.PFK;RESTORE.WINDOW MAIN'' +
                     SMIN' SMIN/ SMIN
```

Figure 10. A TSO CLIST that Redefines PF Key 9

Using CLISTs to Split the Display Screen

Often it is useful to view two streams at the same time. For example, you could view the listing of a program in the top window while using TSO TEST or some other debugging facility in the bottom window. You could also view HELP information or other data in the top window and enter new commands in the bottom window.

The examples on the following pages illustrate how you can use CLISTs to split the display screen horizontally and vertically. These CLISTs are located in SYS1.SAMPLIB so you do not have to create them yourself. The ADFHSPLT CLIST splits the screen horizontally as shown in Figure 13 . The ADFVSPLT CLIST splits the screen vertically as shown in Figure 15.

Both ADFHSPLT and ADFVSPLT depend on the ADFSETUP CLIST to place information in the HEADER and EXTRA2 streams. ADFSETUP requires a partitioned data set of CLISTs that you must allocate to the DDNAME of "SYSPROC" for implicit execution of the other CLISTS. If you use either ADFHSPLT or ADFVSPLT, execute ADFSETUP once at the beginning of your terminal session.

Note: If you issue the RESET command during your session, the Session Manager clears the HEADER stream and then redefines it to contain only that information needed for the default environment. Therefore, you will need to reexecute the ADFSETUP CLIST to place the CLIST information in the HEADER stream again.

```
/*****************************************************************/
/* THIS CLIST PLACES DATA IN THE "HEADER" STREAM AND IN THE "EXTRA2" */
/* STREAM FOR THE "ADFHSPLT" AND THE "ADFVSPLT" CLISTS.          */
/*   IT SHOULD BE EXECUTED ONCE AT THE BEGINNING OF              */
/* THE SESSION.  THIS CLIST REQUIRES A PDS CLIST DATA SET TO BE  */
/* ALLOCATED TO THE DDNAME OF "SYSPROC" FOR IMPLICIT EXECUTION OF */
/* OTHER CLISTS - AS DESCRIBED IN "CLISTS: IMPLEMENTATION AND    */
/* REFERENCE".                                                   */
/*****************************************************************/
SMPUT /PUT 'TOP ===>  BOTTOM=>  RIGHT =>LEFT ==>' HEADER I(2)/
SET I=1
DO WHILE &I < (&SYSLTERM-5)
SMPUT /PUT | EXTRA2 I(2);PUT | EXTRA2 I(2); PUT | EXTRA2 I(2)/
SET &I=&I+1
END
```

Figure 11. ADFSETUP CLIST

The comments in the ADFHSPLT CLIST (see Figure 12) outline how this CLIST redefines the screen. Notice also how the LTITLE window is scrolled to the right and left to display "TOP" or "BOTTOM" in the HEADER stream. This indicates which window is the current default window so you can easily control which window the Session Manager moves when you press one of the scroll PF keys. When you execute the ADFHSPLT CLIST, your display screen looks similar to Figure 13 .

```
PROC 0 LINE()
/*********************************************************************/
/* THIS CLIST SETS UP A "SPLIT - SCREEN" SESSION MANAGER SCREEN      */
/* LAYOUT. IT ASSUMES THE IBM - SUPPLIED DEFAULT SCREEN LAYOUT IS    */
/* THE CURRENT SCREEN LAYOUT WHEN IT IS INVOKED:                     */
/* - THE WINDOW "LINE" IS MOVED TO THE ROW SPECIFIED BY "LINE".      */
/* - THE WINDOW "CURRENT" IS EXPANDED TO FILL IN THE SPACE CREATED   */
/*       AND BECOMES THE "BOTTOM" WINDOW.                            */
/* - THE WINDOW "SPACE" IS DEFINED TO FILL IN THE AREA ABOVE THE     */
/*       SCROLL AMOUNT VALUE IF NEEDED.                              */
/* - THE SCREEN AND PF KEYS ARE SAVED BEFORE THE SCREEN IS MODIFIED. */
/*********************************************************************/
IF &LINE =     THEN SET &LINE = &EVAL((&SYSLTERM-5)/2)
IF &LINE < 2 | &LINE > &EVAL(&SYSLTERM-3) THEN +
 EXIT
SET &TOPS = &LINE -1
SET &BOT = &LINE +1
SET &BOTS = &EVAL(&SYSLTERM-1) - &BOT
SET &BOTSX = &EVAL(&SYSLTERM-3) - &BOT
IF &BOTSX > 0 THEN +
  SET &WIN = +
      DEF.W SPACE &BOT &EVAL(&SYSWTERM-17) &BOTSX 18 P(Y) V(EXTRA2);+
            S.A 2 SPACE;
ELSE +
  SET &WIN =
SMPUT /SAVE SCREEN;SAVE.PFK;+
       SAVE.WIN MAIN;SAVE.WIN LINE;SAVE.WIN CURRENT;+
       DEL.WIN MAIN;DEL.WIN LINE;DEL.WIN CURRENT;+
       DEFINE.WINDOW MAIN 1 1 &TOPS &EVAL(&SYSWTERM);+
       DEFINE.WINDOW LINE &LINE 1 1 &EVAL(&SYSWTERM);+
       DEFINE.WINDOW CURRENT &BOT 1 &BOTS &EVAL(&SYSWTERM-18);+
       &WIN.+
       RES.WIN CURRENT;RES.WIN LINE;RES.WIN MAIN;+
       CHANGE.WINDOW MAIN OVERLAP(1) HOLD(0)/
/*********************************************************************/
/* DEFINE PF 3/15 TO RE-INVOKE THIS CLIST, ACCEPTING AS INPUT THE    */
/* LINE (ROW) TO PLACE THE SPLIT AT. PRESSING PF 3/15 WITH NO INPUT  */
/* WILL RESTORE THE SCREEN AND PF KEYS TO THEIR ORIGINAL STATUS.     */
/*                                                                   */
/* PF 9/21 IS DEFINED TO SWITCH THE DEFAULT WINDOW BETWEEN THE "TOP" */
/* WINDOW AND THE "BOTTOM" WINDOW: THUS SETTING WHICH WINDOW IS      */
/* SCROLLED WHEN PF 5/17, 7/19, 8/20, 10/22, 11/23, 12/24 IS PRESSED.*/
/* THE "LTITLE" WINDOW IS MOVED TO DISPLAY THE NAME OF THE WINDOW    */
/* THAT IS CURRENTLY THE DEFAULT WINDOW. (THE NAMES WERE PREVIOUSLY  */
/* IN THE "HEADER" STREAM).                                          */
/*********************************************************************/
SMPUT /CHANGE.PF 3 'PUT ''RESTORE.SCREEN;RESTORE.PFKS;+
       RESTORE.PFKS;'' SMIN;+
       PUT ''%ADFHSPLT L(0¢1.)'' TSOIN' SMIN SUB(¢);+
       CHANGE.PF 9 'CHANGE TERMINAL DEFAULT(CURRENT);SAVE.PF;+
             SCROLL.LEFT LTITLE A(MAX);SCROLL.RIGHT LTITLE A(10);+
       CHANGE.PF 9 ''CHANGE.TERMINAL DEFAULT(MAIN);+
       RESTORE.PF;SCROLL.LEFT LTITLE A(MAX)'' SMIN' +
             SMIN;SCROLL.ABSOLUTE 11 LTITLE/
SMPUT /CHANGE.PF 15 'PUT ''RESTORE.SCREEN;RESTORE.PFKS;+
       RESTORE.PFKS;'' SMIN;+
       PUT ''%ADFHSPLT L(0¢1.)'' TSOIN' SMIN SUB(¢);+
       CHANGE.PF 21 'CHANGE TERMINAL DEFAULT(CURRENT);SAVE.PF;+
             SCROLL.LEFT LTITLE A(MAX);SCROLL.RIGHT LTITLE A(10);+
       CHANGE.PF 21 ''CHANGE.TERMINAL DEFAULT(MAIN);+
       RESTORE.PF;SCROLL.LEFT LTITLE A(MAX)'' SMIN' +
             SMIN;SCROLL.ABSOLUTE 11 LTITLE/
```

Figure 12. ADFHSPLT CLIST

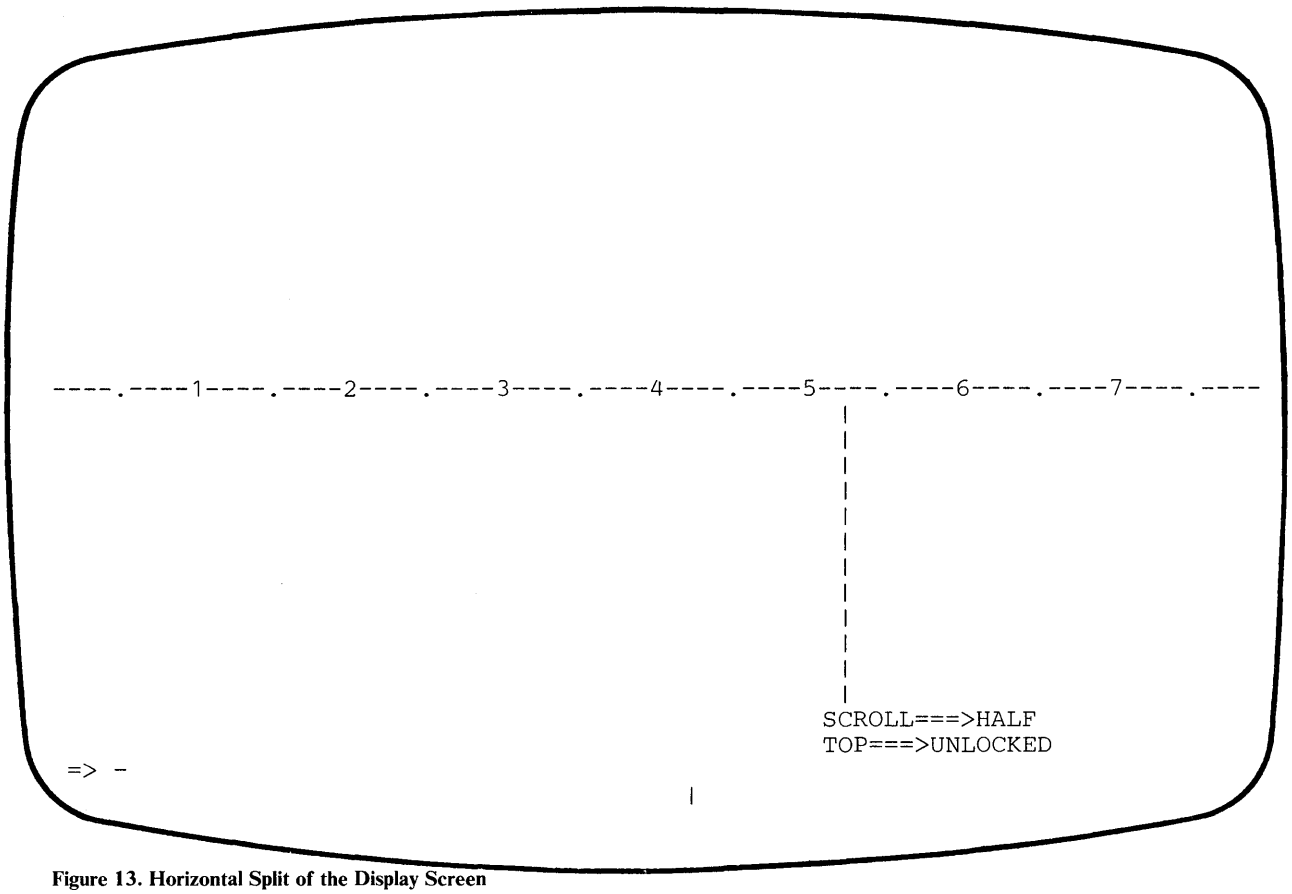


Figure 13. Horizontal Split of the Display Screen

The ADFVSPLT CLIST is very similar to the ADFHSPLT CLIST. You might find this CLIST useful when testing programs or comparing members of a partitioned data set. For example, you could use the CLIST to do the following:

- Use the LISTDS command to list the members of a partitioned data set
- Lock the MAIN window by pressing PF10 (scrolling to the left when the window is already at the left border merely locks the window)
- Execute the ADFVSPLT CLIST and press PF9 to make the RIGHT window the default window
- Press PF12 to unlock the RIGHT window and enter the LISTDS command for the second data set whose members you want to display
- By using PF9 to change the default window to the left side (the MAIN window), you can use the other PF keys to scroll the windows and compare the members of the data sets

Figure 14 is the ADFVSPLT CLIST and Figure 15 illustrates what the display screen looks like when this CLIST has been executed.

```
PROC 0 COL()
/*********************************************************************/
/* THIS CLIST SETS UP A "SPLIT - SCREEN" SESSION MANAGER SCREEN      */
/* LAYOUT. IT ASSUMES THE IBM - SUPPLIED DEFAULT SCREEN LAYOUT IS    */
/* THE CURRENT SCREEN LAYOUT WHEN IT IS INVOKED:                     */
/* - THE "MAIN" WINDOW IS SPLIT VERTICALLY:                          */
/* - THE "MAIN" WINDOW IS REDEFINED WITH A NEW WIDTH OF "COL" -1     */
/*       AND BECOMES THE "LEFT" WINDOW.                              */
/* - A NEW WINDOW "SPLIT" IS CREATED TO PROVIDE A VISUAL INDICATION  */
/*   OF THE SPLIT. IT VIEWS THE "EXTRA2" STREAM IN WHICH VERTICAL    */
/*   BARS HAVE BEEN PLACED PREVIOUSLY.                               */
/* - THE WINDOW "RIGHT" IS DEFINED TO FILL IN THE AREA TO THE RIGHT  */
/*   OF THE "SPLIT" WINDOW. ITS ATTRIBUTES ARE FILLED IN BY          */
/*   RESTORING A COPY OF THE ATTRIBUTES OF THE "MAIN" ("RIGHT")      */
/*   WINDOW FROM THE "WINDOW" STACK.                                 */
/* - THE SCREEN AND PF KEYS ARE SAVED BEFORE THE SCREEN IS MODIFIED. */
/*********************************************************************/
IF &COL =  THEN SET &COL = &EVAL((&SYSWTERM)/2)
IF &COL < 5 | &COL > &EVAL(&SYSWTERM-5) THEN +
 EXIT
SET &LEFTS = &COL - 1
SET &RIGHT = &COL + 2
SET &RIGHTS = &EVAL(&SYSWTERM+1) - &RIGHT
SMPUT /SAVE.SCREEN;SAVE.PFK;+
       SAVE.WIN MAIN;SAVE.WIN MAIN;DELETE.WIN MAIN;+
       DEFINE.WIN MAIN 1 1 &EVAL(&SYSLTERM-5) &LEFTS;+
       DEFINE.WIN SPLIT 1 &COL &EVAL(&SYSLTERM-5) 2 +
             HOLD(0) VIEW(EXTRA2) PROTECT(YES);+
             SCROLL.ABSOLUTE 2 SPLIT;+
       DEFINE.WIN RIGHT 1 &RIGHT &EVAL(&SYSLTERM-5) &RIGHTS;+
       RESTORE.WIN MAIN;RESTORE.WIN RIGHT/
/*********************************************************************/
/* PF 9/21 IS DEFINED TO SWITCH THE DEFAULT WINDOW BETWEEN THE "LEFT"*/
/* WINDOW AND THE "RIGHT" WINDOW: THUS SETTING WHICH WINDOW IS       */
/* SCROLLED WHEN PF 5/17, 7/19, 8/20, 10/22, 11/23, 12/24 IS PRESSED.*/
/* THE "LTITLE" WINDOW IS MOVED TO DISPLAY THE NAME OF THE WINDOW    */
/* THAT IS CURRENTLY THE DEFAULT WINDOW. (THE NAMES WERE PREVIOUSLY  */
/* IN THE "HEADER" STREAM.                                           */
/*                                                                   */
/* DEFINE PF 3/15 TO RE-INVOKE THIS CLIST, ACCEPTING AS INPUT THE    */
/* LINE (ROW) TO PLACE THE SPLIT AT. PRESSING PF 3/15 WITH NO INPUT  */
/* WILL RESTORE THE SCREEN AND PF KEYS TO THEIR ORIGINAL STATUS.     */
/*********************************************************************/
SMPUT /CHANGE.PF 9 'CHANGE.TERMINAL DEFAULT(RIGHT);SAVE.PF;+
             SCROLL.LEFT LTITLE A(MAX);SCROLL.RIGHT LTITLE A(20);+
       CHANGE.PF 9 ''CHANGE.TERMINAL DEFAULT(MAIN);RESTORE.PF;+
             SCROLL.LEFT LTITLE A(MAX);+
             SCROLL.RIGHT LTITLE A(28)'' SMIN' SMIN;+
             SCROLL.ABSOLUTE 11 LTITLE;+
             SCROLL.RIGHT LTITLE A(28);+
       CHANGE.PF 3 'PUT ''RESTORE.SCREEN;RESTORE.PF;+
             RESTORE.PF'' SMIN;+
             PUT ''%ADFVSPLT COL(0¢1.)'' TSOIN' SMIN SUB(¢)/
SMPUT /CHANGE.PF 21 'CHANGE.TERMINAL DEFAULT(RIGHT);SAVE.PF;+
             SCROLL.LEFT LTITLE A(MAX);SCROLL.RIGHT LTITLE A(20);+
       CHANGE.PF 21 ''CHANGE.TERMINAL DEFAULT(MAIN);RESTORE.PF;+
             SCROLL.LEFT LTITLE A(MAX);+
             SCROLL.RIGHT LTITLE A(28)'' SMIN' SMIN;+
             SCROLL.ABSOLUTE 11 LTITLE;+
       CHANGE.PF 15 'PUT ''RESTORE.SCREEN;RESTORE.PF;+
             RESTORE.PF'' SMIN;+
             PUT ''%ADFVSPLT COL(0¢1.)'' TSOIN' SMIN SUB(¢)/
```

Figure 14. ADFVSPLT CLIST

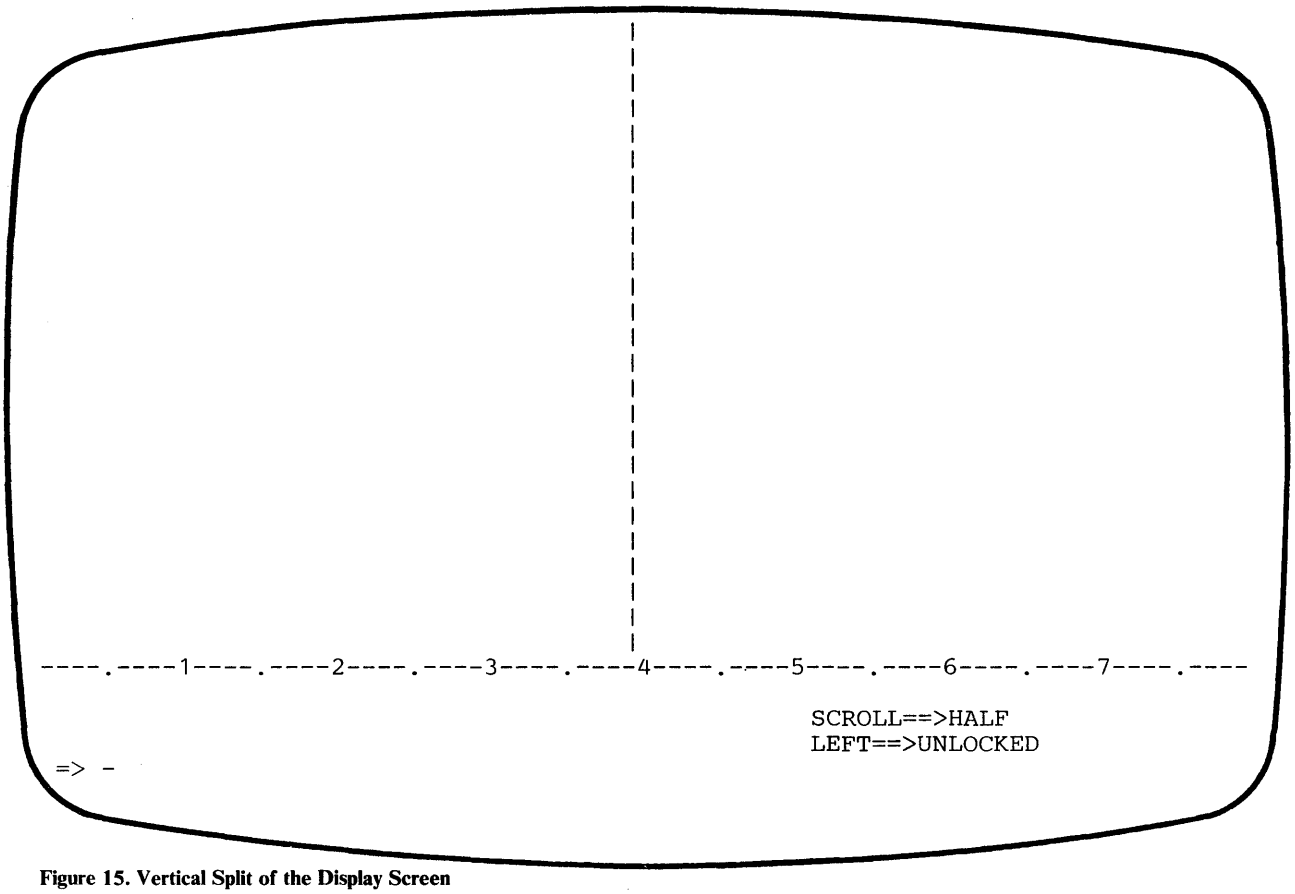


Figure 15. Vertical Split of the Display Screen

Both ADFHSPLT and ADFVSPLT define PF 3/15 as the END key. When you are ready to go back to your original screen, just press PF3/15. If at any time you want to switch back to the default screen layout and PF key definitions, clear the screen by pressing the CLEAR key and type in the RESET command.

Index

T

TSO Extensions (TSO/E)
Session Manager Program Reference
SC28-1306-0

READER'S COMMENT FORM

This manual is part of a library that serves as a reference source for system analysts, programmers, and operators of IBM systems. You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you. Your comments will be sent to the author's department for whatever review and action, if any, are deemed appropriate.

Note: ***Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.***

Possible topics for comment are:

Clarity Accuracy Completeness Organization Coding Retrieval Legibility

If you wish a reply, give your name, company, mailing address, and date:

_____

_____

_____

_____

Note: Staples can cause problems with automated mail sorting equipment.
Please use pressure sensitive or other gummed tape to seal this form.

Cut or Fold Along Line

What is your occupation? _____

How do you use this publication? _____

Number of latest Newsletter associated with this publication: _____

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail directly to the address in the Edition Notice on the back of the title page.)

SC28-1306-0

Reader's Comment Form

Cut or Fold Along Line

Fold and tape | Please Do Not Staple | Fold and tape

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE:

International Business Machines Corporation
Department D58, Building 920-2
PO Box 390
Poughkeepsie, New York 12602

Fold and tape | Please Do Not Staple | Fold and tape

TSO Extensions(TSO/E) Session Manager Program Reference (File No. S370-39) Printed in U.S.A. SC28-1306-0

Publication Number
SC28-1306-0

File Number
S370-39

Program Number
5665-285

Printed in
U.S.A.

IBM®

SC28-1306-0