IBM Session Manager for z/OS

Technical Reference

Version 3 Release 1

SC34-7147-00

IBM Session Manager for z/OS

Technical Reference

Version 3 Release 1

SC34-7147-00

Note

Before using this information and the product it supports, be sure to read the general information under "Notices" on page 311.

This edition applies to Version 3 Release 1 of IBM Session Manager for z/OS, program number 5655-U98, and to all subsequent versions, releases, and modifications until otherwise indicated in new editions.

Contents

About this manual

This is the *Technical Reference* manual for IBM® Session Manager for z/OS®. It is a detailed manual designed for use by Technical Programmers whose task it is to install, configure, and subsequently maintain Session Manager.

Note Any references in this manual to "Session Manager version 1.3.15" and to "1.3 Functional Enhancement PTF 3" are synonymous.

Configuring Session Manager

Session Manager can be tailored to Installation requirements using:

- **Product configuration statements**

  In general, the extreme flexibility of the configuration statements is found to cater for most user requirements.

- **Online and Batch Administration**

  Online Administration: Instead of supplying product configuration statements directly, Online Administration (hereafter "OLA") enables administrators and end-users of Session Manager to tailor the product using a series of menus, lists and attribute display panels.

  Batch Administration: If many changes are required to a large number of configuration definitions, this capability enables administrators and end-users of Session Manager to tailor the product using a batch job.

  For more information, see the *Online and Batch Administration* manual.

- **Panels and Scripts**

  Installation-specific facilities can be created using Session Manager Panels and Scripts, and conditional logic can be incorporated using the product's Panel and Script Language (TPSL). For details, see the *Panels, Scripts and Variables* manual.

  To meet particular needs, a User exit is available, containing several exit points and access to certain variables, enabling user code to be executed. For details, see the chapter 'Session Manager user exit processing' in the *Installation and Customization* manual.

For more information, see 'System definition' on page 15.

External Security Managers

External Security Managers (hereafter "ESMs"), such as RACF®, can be used with Session Manager to authenticate users, set their authorization level and OLA security class, and determine which applications a user can access.

For details, see the 'Defining security and implementing dynamic menus' chapter in the *Installation and Customization* manual.

Session Manager documentation

The following documentation accompanies Session Manager:

Manual	Purpose
Installation and Customization	Goes through the steps required to install the Session Manager software, and provides general information on the methods and options available to configure and operate your system.
User and Administrator	Describes in detail the features and facilities provided by Session Manager.
Online and Batch Administration	Explains the set-up and configuration of OLA, how to use the interface, and how to utilize both OLA and Batch Administration to modify the Session Manager configuration.
Technical Reference	Provides a detailed reference for Session Manager commands and configuration statements, along with problem diagnosis assistance.
Quick Reference	Provides a quick way to find the correct syntax for commands, configuration statements, and variables, without detailed explanations.
Panels, Scripts and Variables	Gives a detailed technical account of defining panels, using scripts and variables, and the product's Panel and Script Language (TPSL).
Messages and Codes	Contains explanations of all messages issued by Session Manager, and the actions that should be taken.

Additionally, the *Program Directory* contains information for systems programmers about the program material and procedures for installing IBM Session Manager under z/OS.

New users should review the *User and Administrator* manual to gain an understanding of Session Manager concepts, and Technical Programmers should review the *Installation and Customization* manual in order to tailor the product to the Installation's requirements.

In general, the extreme flexibility of the configuration statements is found to cater for most user requirements. Panel and script definitions may be provided with conditional logic using the Session Manager Panel and Script Language and numerous variables are provided to view and modify a wide variety of data. If any particular needs cannot be met, a user exit is available, containing several exit points and access to certain variables, enabling user code to be executed. Full details of the User Exit are supplied in the *Installation and Customization* manual.

Conventions

The following typographic conventions are used:

`boldface`	Indicates a command or keyword that you should type, exactly as shown. When mixed case is used, the element in upper case represents the shortest acceptable form. For example, `MSGsuffix` can be abbreviated as far as `MSG`.
`italics`	Indicates a variable for which you should substitute an appropriate value.
`monotype`	Indicates literal input and output.
Ctrl+D	Indicates two or more keys pressed simultaneously.
`[ ]`	Brackets surround an optional value.
`|`	Vertical bars separate alternative values from which you must make a selection.
`...`	Ellipsis indicates that the preceding element may be repeated.
@	Some commands or key sequences make use of the 0x7C (that is, x'7C') character. When using the English language code page, this character is displayed as the @ sign, but may be displayed as a different character in some other code pages. In this document, the 0x7C character is always presented as the @ sign. You should enter the appropriate 0x7C character symbol for the code page you are using.

Session Manager internal names

Session Manager internal names may consist of alphanumeric characters and these special characters:

£ (pound sign)
(hash sign)
% (percent sign)
@ (at sign)
- (dash)
_ (underscore)

Summary of new features

For a summary of changes made to the product in its most recent releases, please refer to the *Installation and Customization* manual.

CHAPTER 1

System definition

IBM Session Manager for z/OS is a VTAM® application which runs in its own z/OS address space, enabling terminals connected to it to run multiple sessions with other applications. Once a user is signed on to Session Manager, it takes on the responsibility for managing and controlling all the sessions to which the user requires access.

This chapter contains these sections:

- A description of the control statement format and some usage rules.
- An explanation of how configuration logic operates.
- A brief description of each of the control statements available in Session Manager.
- Information on configuration defaults and overrides.

The subsequent chapters provide a complete description of all control statements and their associated parameters which should be used when generating the Session Manager environment, and when amending or enhancing it.

Control statement format

(See also 'Session Manager internal names' on page 13.)

Each Session Manager configuration control statement comprises one or more fixed length, 80 byte records of which all columns may be used. If sequence numbers are required at the end of each record, then the numbers should be prefixed with slash asterisk (/*), so that they are treated as comments. Each statement is considered complete when the first field on a subsequent record contains a new statement name. The one exception to this is data specified between the `TEXTSTART` and `TEXTEND` parameters of the `PANEL` statement, since the data may contain Session Manager statement names and keywords.

Control statements and parameters may be specified in mixed case, but for reasons of clarity, statement and parameter names are shown in upper case in this chapter. Any parameters that must be in upper case, for example the output class specification in the `HCROUTE` statement, are automatically converted by Session Manager.

Most keywords may be abbreviated. In this and the following section, the minimum abbreviation is indicated by the upper case portion of the keyword, any amount of the lower case part may also be specified. For example, for the keyword `ALLOWEscape`, `ALLOWE`, `ALLOWES`, `ALLOWESC`, `ALLOWESCA` and `ALLOWESCAP` are all valid.

Parameters and their values specified on the same control statement should be separated by one or more spaces. An equal sign or comma may be used if preferred.

Statement and parameter names must be wholly contained on a record, although parameter values may be placed on a subsequent record and may be continued over several records, provided that the continuation rules are observed.

In all parameters where `Yes` and `No` are valid, `ON` and `OFF` are valid as synonyms. If the parameter is specified without a following `Yes`, `No`, `ON` or `OFF`, `Yes` is assumed. For example, `ALLUSERS YES`, `ALLUSERS ON` and `ALLUSERS` have the same effect.

Records beginning with *

Records beginning with an asterisk (*) or containing only spaces are ignored.

Records containing /* or */

If a slash asterisk (/*) or asterisk slash (*/) is found anywhere within a record, it, and the remainder of the record, is ignored. This gives the system administrator the option of placing descriptive comments among the control statements.

Records beginning with %%

Records beginning with two consecutive percent signs (%%) contain TPSL (the Panel and Script Language) logic statements which are to be executed during configuration processing – see 'Configuration logic' on page 20.

Text data and the use of delimiters

Except when defined within TEXTSTART and TEXTEND segments, all text data must be delimited at start and end with any one of the special delimiting characters – a single quotation mark ('), a double quotation mark ("), an exclamation mark (!), and a question mark (?). Alternatively, matched parentheses (...) may be used.

The choice of text delimiters enables quotes to be specified unambiguously within text data, since there is no need to insert multiple quotes. In fact this may produce an undesirable result.

Variable substitution in control statements

See also 'Variable substitution in configuration statements' on page 21.

Session Manager provides access to a large set of variables, and these are described individually in the chapter 'Session Manager Variables' in the *Panels, Scripts and Variables* manual. The variables may be used in different ways, but one use is to substitute the current value of a variable in a control statement parameter.

The following list shows the control statements and parameters in which variables may be substituted, and the point at which a variable is replaced by a value.

APPL, PROFILE, TERMINAL and USER statements:	
ACB	At session initiation
DATA	At session initiation
DESC	When the description is displayed
CMD	When the menu is displayed
SAUTOSEQ type C	When the menu is displayed
HCFORMAT statement:	
HEADER	When the hardcopy is performed
TRAILER	When the hardcopy is performed
HCROUTE statement:	
CLASS DESTINATION NAME NODE OUTPUT	When the hardcopy is performed (in all cases)
PANEL and SCRIPT statements:	
AUDITMSG	When the script verb is run
CALL	When the panel is displayed
CALLEXIT	When the script verb is run

FIELD	When the panel is displayed
INPUT	When the script verb is run
LET	On execution
TEXTSTART/TEXTEND	When the text is displayed
ISZCMD	When the script verb is run
ISZCMDA	When the script verb is run
USERMSG	When the script verb is run

When variables are used to substitute a current value in a control statement parameter, the variable name must be preceded immediately by an ampersand (&). The variable should be followed immediately either by an ampersand, or a space. These delimiters should appear after the closing parenthesis of a displacement and length qualifier, when one is used. For example a HEADER parameter in an HCFORMAT statement may be coded as follows:

```
Hcformat
    Header 'Session Manager hardcopy at terminal'-
    '&t_termid by user'-
    '&t_user(2,3) at &t_time(5)&'
```

When two or more variables are to be concatenated, there must be two ampersands between each. The first denotes the end of the first variable and the second denotes the start of the second variable. If only one ampersand is specified, then the second variable is taken as a literal, for example:

```
&t_user&t_time
```

would result in 'PLXXt_time' for user PLXX when substituted, but:

```
&t_user&&t_time
```

could result in 'PLXX15.30.05' when substituted.

For simplicity, it is easier to remember that each variable should always be enclosed by ampersands. This rule must be adhered to when two or more variables are to be concatenated.

Continuation rules

The Session Manager configuration control statements consist of a series of keywords and parameter values. These may be placed anywhere on a record, and may continue over several records. For parameter values that are not delimited, it is not necessary to code any continuation markers.

Where a parameter value is delimited text, for example message text, and it continues over more than one record, care must be taken to specify the delimiters and continuation in the correct manner. At a convenient point in the text, the appropriate end delimiter should be coded, followed **immediately** by a hyphen (-) to indicate continuation. The next start delimiter, in the same statement, that matches the previous start delimiter, indicates the beginning of the continued data. Any parameters, data, or spaces between the end delimiter and start delimiter, or the beginning of the next control statement, are ignored. For example:

```
FIELD   (The last parameter)-
        or keyword (entered is not recognized by
        Session Manager)
```

would suppress ‘or keyword’ and the field is output containing:

```
The last parameter entered is not recognized by Session
Manager
```

Continuation may be used on a single record to interrupt and continue delimited text. For example:

```
FIELD ?This message is issued?- sometimes ?at a terminal?
```

produces a string as follows:

```
This message is issued at a terminal
```

Configuration logic

The Session Manager system definition can be modified without changing the definition statements themselves by two methods:

- By providing an exit routine for exit point `E05` of the Session Manager User exit (the Configuration Statement Processing exit point).
- By inserting logic statements into the configuration file.

To use the `E05` exit point you should refer to the chapter 'The Session Manager User Exit' in the *Installation and Customization* manual. However, you may find that using logic statements in the configuration is a simpler and more flexible alternative.

The logic statements available for modifying the configuration are a subset of the Panel and Script Language (TPSL), and comprise the following statements:

```
IF
SELECT
LET
```

For the `IF` and `SELECT` statements, a combined maximum of eight levels of nesting per member are allowed.

Configuration logic statements must appear on records beginning with a double percent sign (%%). Records starting thus must contain **only** logic statements. Logic records can appear anywhere in the configuration, either between configuration control statements or embedded among the parameters of a statement.

The logic statements are processed **after** any modifications have been made to the configuration records by the `E05` exit, and are applied at MEMBER level. (Note that any `IF` or `SELECT` structures which are still active at the end of a member will produce diagnostic messages.) Any control statements which are skipped as a result of the logic are ignored completely when the system definition is built.

Within the logic statements, any system (global) variable and any user-defined variable, either global or locally defined, can be used. Note, however, that local variables and subscript variables apply to a member and are therefore reset at the start of each member.

A likely use of configuration logic is to enable the Session Manager system definition to be modified in a disaster-recovery situation. For example, if normal processing occurs 'onsite' but for disaster recovery the Session Manager system needs to be set up 'offsite', where there are none of the normal users, the following statements could be inserted permanently in the system definition:

```
%% Let gc_machine = 'onsite'
    ...
COPY SYSDEF
%% If gc_machine NE 'offsite'
COPY USERG1
COPY USERG2
COPY USERG3
%% End
```

During normal initialization of Session Manager, the users defined in the members USERG1, USERG2 and USERG3 will be set up in the system. To start up the system 'offsite' without these groups of users, it is then only necessary to insert one additional statement to the definition of the configuration:

```
%% Let gc_machine = 'offsite'
```

Provided this precedes the IF statement which tests the variable gc_machine, the user definitions will be bypassed.

Variable substitution in configuration statements

See also 'Variable substitution in control statements' on page 17.

For Session Manager 1.1.05 and higher, a Session Manager variable can be substituted on *any* configuration statement *before* the contents of the statement are passed to the Session Manager parser. To do so, prefix and suffix the variable with two consecutive percent signs (%%).

For example, if T_CONFIG_SUF is 60, then:

```
PCOPY SYSTEM SYSTEM%%T_CONFIG_SUF%%
```

would be interpreted as:

```
PCOPY SYSTEM SYSTEM60
```

Control statement summary

Below is a summary of the Session Manager configuration control statements with a brief description of the purpose of each. They appear in the same order as the control statement prototypes in the following section. Statements defining similar options appear close to each other. Detailed statement descriptions are given in later chapters.

`OPTION`	This statement specifies some miscellaneous options such as whether control statements are to be printed, whether the system is to be activated after statement validation, or whether the control statements are to be merely validated.
`INSTALLSU`	Specifies that any Selectable Units that have been supplied are to be installed.
`COPY`	If you do not use Online and/or Batch Administration to tailor the product, this statement enables additional Session Manager control statements to be read from the source library during product initiation.
`PCOPY`	If Online and/or Batch Administration is used to tailor the product, this statement loads member(s) of the PDS(E)s allocated to a specified `DDNAME` during product initiation.
`SYSTEM`	This statement defines various options which are system wide, that is, they apply to the complete Session Manager environment.
`PROFILE`	Defines one or a group of applications and their related attributes, which may then be assigned to many users or terminals.
`USER`	Specifies the attributes to be assigned to specified users of the system.
`TERMINAL`	Specifies the attributes to be assigned to specified terminals regardless of their type.
`LU`	Specifies the attributes to be assigned to specified LU-type terminals.
`APPL`	Specifies the characteristics of an application or system. It provides a convenient method of centralizing the definition of application characteristics. This may be a VTAM application that is unable to deal with parallel sessions. In this instance, the `APPL` statement indicates the names of ranges of ACBs to be used for ACB allocation.
`RANGE`	Defines ranges of ACB names which can be allocated by Session Manager when initiating sessions with applications which are unable to deal with parallel sessions.
`LINK`	Defines the characteristics of a link to another Session Manager network node. A Network node can have as many links defined to other network nodes as necessary.
`RUSER`	Specifies users who can issue remote commands.
`GROUP`	Defines a named group, which may consist of users, profiles and terminals, for the purpose of receiving messages and broadcasts, and for use with the SPY facility.

COMMAND	Assigns security codes to the Session Manager operator commands, or denotes a script that is to be run when the command is issued.
MESSAGE	Enables the text of Session Manager error messages to be redefined, and enables specific messages to be directed to selected destinations.
HCPROFILE	Defines a set of hardcopy options for use with the screen Hardcopy facility.
HCFORMAT	Defines a set of heading and trailing lines for screen hardcopies.
HCROUTE	Defines a hardcopy print route.
AUDITROUTE	Defines the Audit file print route.
TRACEROUTE	Defines the trace print route.
TRANSTABLE	Defines a table to translate characters in commands, control statements and panels.
PANEL	Defines screen layouts for the Session Manager system, including the Signon, Menu, Data Display and Help screens. Full details of Panel definitions can be found in the *Panels, Scripts and Variables* manual.
PHEADER	Defines a panel header sub-definition.
PCONTENT	Defines a panel content sub-definition.
PTRAILER	Defines a panel trailer sub-definition.
PPROCESS	Defines a panel process sub-definition.
SCRIPT	Defines automatic input streams. Full details of script definitions can be found in the *Panels, Scripts and Variables* manual.
PATCH	Enables changes to be applied to the executable code. *This should only be used at the request of your local support representative.*
PATCHSU	Enables changes to be applied to Selectable Units. *This should only be used at the request of your local support representative.*
APPLYSU	Applies a Selectable Unit.
REMOVESU	Removes a Selectable Unit.

A list of prototypes for these control statements (and for the Panel and Script statements) is provided in the *Quick Reference* manual.

Configuration defaults and overrides

Session Manager provides considerable flexibility in the definition of the parameters which affect end-user facilities. This is due to Session Manager catering for a wide variation in requirements as regards defaults and overrides.

Most parameters and options that affect each Session Manager user may be defined in a number of statements. These are the `PROFILE`, `USER`, `TERMINAL`, `LU` and `APPL` statements, although the `APPL` statement is only used to specify options affecting a session with an application. Parameters that are common to each of these statements, and which define both enduser options and session options are described separately in 'Common parameters' on page 29 and then referenced from each individual statement description.

Since there are so many places where an option may be specified, it can be very difficult to determine which option will be in force at a particular time if that option has been specified on each statement. This section describes the way in which Session Manager processes the various statements. This should be considered when creating the configuration file, in order to arrive at sensible conventions for parameter specification so that confusion may be avoided.

The processing for profile (or enduser) options is different to that for session definitions and these are therefore described separately in the following paragraphs.

Profile options and limits

For options such as the menu escape id, Session Manager inspects these statements in the order provided until a value is found:

- `USER` or `TERMINAL` (whichever is applicable)

  Note Parameters that are defined as `RETAIN`ed on a `TERMINAL` statement will be kept in force, unless specifically overridden by an identical parameter on an applicable `USER` statement.

- `PROFILE`
- `SYSTEM`
- Session Manager defaults
  (if no value is found on the previous statements)

If a value is supplied by the User exit, then this overrides any supplied by a statement.

Rules for profile selection

The following table shows how a PROFILE name is selected.

For Session Manager 1.1.05 and higher, a user can be associated with multiple profiles – however, if a valid profile is specified at the E21 exit level then *that profile only will be used.*

For Session Manager 2.2.00 the system can be configured to use an External Security Manager to determine users' allocated Profiles (see the 'Defining security' chapter in the *Installation and Customization* manual).

E21 exit	USER statement	SYSTEM statement
1	no	2
1	no	2
1	2	3
1	2	3
no	no	1
no	no	1
no	1	2
no	1	2

Notes

no not specified at that level

1 1st choice if specified at that level

2 2nd choice if specified at that level

3 3rd choice if specified at that level

Session definitions

Session options

Session options may be obtained from several sources: these are the USER or TERMINAL statement, and the PROFILE, APPL and SYSTEM statements. Session Manager supplies default options, and default options can also be defined on the PROFILE, USER, TERMINAL and SYSTEM statements.

A user may find that some of the sessions are obtained from a USER statement and some from a PROFILE statement. Alternatively, some sessions may be obtained from a TERMINAL statement and some from a PROFILE statement. All session definitions on a TERMINAL statement are ignored when there is an applicable USER statement. If there is no USER or TERMINAL statement, but signon is allowed under control of the user exit, and the user exit does not specify a profile, the default profile is used.

A single session definition, however, is obtained solely from one source. Session options are only merged when a set of default options is specified.

Session Manager first locates the session definition by searching these statements:

- USER or TERMINAL

- PROFILE
- APPL
  (only when the session definition contains a REFAPPL parameter)

If the PROFILE statement defines a session for which there is a matching session on the USER or TERMINAL statement, the profile session definition is ignored.

Session defaults

When certain session options have been omitted from the specific session definition, but have been defined as session defaults, the default options are merged with the session definition. Session Manager locates session defaults by searching these statements:

- USER or TERMINAL (whichever is applicable)

  Note Session defaults that are defined as RETAINed on a TERMINAL statement will be kept in force after a signon, unless specifically overridden by an identical parameter on an applicable USER statement.

- PROFILE
- APPL
  (only when the session definition contains a REFAPPL parameter)
- SYSTEM
  (to determine the value of any option that may have been omitted)
- Session Manager defaults
  (if no value is found on the previous statements)

General points

- Several parameters may have any one of four options, Yes (or ON) or No (or OFF).
- Unless otherwise stated, if the parameter is omitted, the default option is No. For example, if PCTRANSFER is not specified on any statement, the setting is PCTRANSFER No.
- If the parameter keyword is specified, but no option is supplied, the default option is Yes. For example, specifying PCTRANSFER with no options, is the same as specifying PCTRANSFER Yes.
- Specifying a parameter on say a USER statement, overrides the same parameter setting on the PROFILE or SYSTEM statement. See the previous section for full details of how to override options for individual users and terminals.
- Several parameters define an ‘escape’ sequence or key, that can be used to invoke a facility, or initiate an action. In all cases this escape must be delimited by a null or a space when issued, and can be any of the following:
  - a character sequence from one to eight characters in length
  - a function key: PF1–PF24
  - an attention key: PA1–PA3, ‘ATTN’
  - the ‘CLEAR’ key (although this latter key should be used with care)
  - the lightpen: ‘PEN’

- Where a parameter definition includes a Session Manager statement name, a parameter name, or spaces, it should be enclosed in quotes, or parentheses.

CHAPTER 2

Common parameters

This chapter contains the parameters that are common to the `SYSTEM`, `PROFILE`, `USER`, `TERMINAL`, `LU` and `APPL` configuration control statements.

The complete set of control statements is listed in 'Control statement summary' on page 22 and each statement is separately described in detail in the chapters that follow.

The common parameters are broken down into two groups; those that can be specified at the end-user level, and those that can be specified at session level:

- 'Common end-user parameters' on page 30
- 'Common session parameters' on page 51

The parameters are listed in alphabetical order within these sections.

'System definition' on page 15 describes the way in which defaults are processed by IBM Session Manager for z/OS and the order in which parameters are implemented when specified on different statements.

Common end-user parameters

Applicable to the PROFILE, SYSTEM, USER, TERMINAL and LU statements.

ACTivesessions *active-session-limit*

Default: ACTIVESESSIONS 9999

Restricts the maximum number of active applications for each user. *active-session-limit* may be any value in the range 1 to 9999.

AFFINITY *affinity_acbname*

Default: *system_acbname*, as defined on the ACB parameter of the SYSTEM statement (see page 89).

Defines the default Local Session Manager node to which users are to be associated. The *system_acbname* identified as *affinity_acbname* is the Session Manager system which receives the signon data. All signon data (for example, password, user name, and so on) is also passed, thereby enabling the signon to complete without further intervention from the user. The signon data is also passed to the E21 exit in the originating Session Manager so that modifications may be made if required.

The *affinity_acbname* can be specified on the separate statements, and in addition by defining the Signon panel appropriately, using a defined input field (variable t_affinity) the *affinity_acbname* can be modified dynamically. See the PANEL statement in the *Panels, Scripts and Variables* manual.

AUTH *auth-level*

Default: AUTH 1

Defines the authorization level for Session Manager users. The setting of *auth-level* determines the commands which users may issue.

auth-level may be any value in the range 1-9. Details of command security can be found in 'Privilege (authorization) levels' on page 216.

Note This setting will be ignored if an AUTHRESNAME is specified under the SECURITY parameter on the SYSTEM statement. For further information, see 'SYSTEM statement' on page 87.

AUTOSELECT
session-id
applname

Default: AUTOSELECT 0

Enables a session to be automatically selected when the user logs on, based on a specified application name or session number (range1 to 9999). Causes the logon request, Initscript or Session Manager command associated with the defined application name or session number to be issued automatically as soon as the user accesses Session Manager. If PF keys have been defined to access an application, the session number is the PF key number. If more than one session is using the specified applname then the first session will be selected.

Since the user is immediately connected to the relevant application without seeing the Menu screen, this facility is particularly useful in a situation when either a user has only a single application, or when the majority of a user's work is done in a single application.

If the specified session defines a Session Manager command, the command is issued immediately after signon or reconnection.

AUTOSELECT=0 can be specified as a default on the SYSTEM statement and then defined specifically for a user using a PROFILE, USER or TERMINAL statement.

AUTOSEQ *escape* [A|I|E|N *scriptname*]

Default: AUTOSEQ #@

Defines the sequence of characters, or the function or attention key or lightpen, which causes Session Manager to send a script to an application. This input stream is defined for each individual key by the AUTOSCRIPT parameter. For a full description, see the 'Using scripts with applications' section in the 'Post-installation configuration issues' chapter of the *Installation and Customization* manual.

Note In this document, the 0x7C (that is, x'7C') character is always presented as the @ sign. It may be displayed as a different character in some non-English code pages. You should enter the appropriate 0x7C character symbol for the code page you are using.

The *escape*, which must be delimited by a null or a space when issued, can be from one to eight characters long, PF1-24, PA1-3, 'ATTN', 'CLEAR' or 'PEN'.

The optional parameters on this statement determine the type of script to be run when the input sequence is entered:

A – AUTOSCRIPT for the visible session. This is the default.

I – Script name entered with the input sequence.

E – ENDSCRIPT for the visible session.

N – Script name on this subparameter.

By defining the Menu panel appropriately, using a defined input field (variable t_auto) the escape sequence can be modified dynamically. See the PANEL statement in the *Panels, Scripts and Variables* manual.

Note The AUTOSEQ parameter is affected by parameters CMDACTIONKEY and COMMANDPRFXVAL (see CMDACTionkey on page 32 and COMMANDPRFXVAL on page 33).

BACKWARD *escape*

Default: BACKWARD <<

Defines the sequence of characters, or the function or attention key or lightpen, which causes Session Manager to transfer control backwards through the active applications for a Menu panel.

Note See also 'PREVIOUS' on page 43, which causes Session Manager to transfer control back to the previous active session.

The *escape*, which must be delimited by a null or a space when issued, can be from one to eight characters long, PF1-PF24, PA1-3, 'ATTN', 'CLEAR' or 'PEN'.

By defining the Menu panel appropriately, using a defined input field (variable t_bwd) the escape sequence can be modified dynamically. See the PANEL statement in the *Panels, Scripts and Variables* manual.

Note The BACKWARD parameter is affected by parameters CMDACTIONKEY and COMMANDPRFXVAL (see CMDACTionkey on page 32 and COMMANDPRFXVAL on page 33).

BRECeive [Yes|No|ON|OFF|WAITinp|BELL|QUEUEd]

Default: BRECEIVE Yes

The usual way in which broadcasted messages are received causes the recipient's current screen to be temporarily replaced by the broadcast 'message'. With the BRECEIVE parameter it is possible to alter the receiving status.

The BRECEIVE command is available on all Menu screens to override the configuration option, unless inhibited by the COMMAND statement described on page 157.

Yes or ON causes the recipient's current screen to be temporarily replaced by the broadcast message.

No or OFF causes broadcast messages to be suppressed. The only messages displayed are those sent using BROADCAST/MSG=URGENT; these cannot be suppressed.

WAITINP causes broadcast messages to be held until the next time the Enter key, or a PA or PF key, is pressed.

BELL causes messages to be stacked until the next escape to the menu. The terminal alarm sounds when each broadcast is received.

QUEUED, like BELL, causes messages to be stacked until the next escape to the menu, but the terminal alarm is not sounded when a message is received.

Note If the user is using the Windows feature with WAITEVENT MSG in effect (the default), then the BRECEIVE option is ignored and the message is received by the windows script. The default action is to display the message in a separate window.

If the PCTRANSFER command has been set to ON, then the BRECEIVE setting is ignored, and broadcasts are queued.

CMDACTionkey *aidkey*

Default: None

This statement specifies a Session Manager command action key. This action key will only operate when a command prefix value (COMMANDPRFXVAL) has been specified (see below) or its value has been set via the t_actprf variable.

The *aidkey* may be any valid aid key PF1-24 or the special value NONE. There is no default *aidkey*. If NONE is specified this is equivalent to not specifying the CMDACTIONKEY. This is useful when you have a user or set of users who do not wish to use a command action key but one is specified at SYSTEM level.

When the CMDACTIONKEY parameter is specified, SAUTOSEQs that have COMMANDPRFX=Y will only be invoked if the SAUTOSEQ sequence is entered, prefixed with the COMMANDPRFXVAL character, and the specified key is pressed.

On SAUTOSEQs that are intended to be controlled by CMDACTIONKEY, specify a sequence (not a PFKEY) and do not specify an ACTIONKEY.

Situations where `CMDACTIONKEY` must be used

If `CMDACTIONKEY` (and therefore `COMMANDPRFXVAL`) is specified then it **must** be used in the following circumstances:

In an application session, for:

- escape sequences
- transids
- `SAUTOSEQ` definitions with `COMMANDPRFX=Y`.

In all Session Manager panels except the menu, for:

- escape sequences
- transids
- `SAUTOSEQ` definitions with `COMMANDPRFX=Y` (only applicable on Session Manager internal sessions such as OLA and the System Management Menu, not `DLOG`, `QUERY` and so on).
- Session Manager commands, including command scripts, with a setting of `ACTKEY=Y` on the `COMMAND` statement.

Situations where `CMDACTIONKEY` may be used

If `CMDACTIONKEY` (and therefore `COMMANDPRFXVAL`) is specified then it **may** be used in the following circumstances:

In the menu, for:

- escape sequences
- transids
- Session Manager commands, including command scripts.

See member `ISZC1ESC` for details on how to define single character synonyms for Session Manager commands with `CMDACTIONKEY` and `COMMANDPRFXVAL`.

Two updateable user variables, `t_actcmd` and `t_actprf`, allow an installation to add input fields to the user's menu, allowing the user to view and change their `CMDACTionkey` and `COMMANDPRFXVAL` respectively. A value of `NONE` can be specified for *`t_actcmd`*, to remove a previously specified value.

Note If these variables are modified then the modification only lasts for the current Session Manager signon. If you require a change for subsequent sign-on sessions then the `CMDACTIONKEY` and `COMMANDPRFXVAL` parameters must be modified in the configuration (OLA users will use the OLA dialogs and Classic users will need to edit the appropriate configuration member). Note also that if these parameters are modified in the configuration and the change is activated, it will not take place until the next sign-on.

See the *User and Administrator* manual for more details on command action key processing. See also `COMMANDPRFXVAL` below.

`COMMANDPrfxval` *`character`*

Default: None.

Specifies a single *`character`*, to be used as the command prefix.

For example, in a USER statement, the following parameter:
COMMANDPRFXVAL /
will define the '/' forward-stroke character as the command prefix. This parameter is useful when defining command synonyms with the SAUTOSEQ statement.

Two updateable user variables, t_actprf and t_actcmd, allow an installation to add input fields to menu, allowing the user to view and change their COMMANDPRFXVAL (command prefix value) and CMDACTionkey (command action key) respectively. A value of NONE can be specified for *t_actcmd*, to remove a previously specified value. COMMANDPRFXVAL can be used without CMDACTIONKEY.

Note If these variables are modified then the modification only lasts for the current Session Manager signon. If you require a change for subsequent sign-on sessions then the CMDACTIONKEY and COMMANDPRFXVAL parameters must be modified in the configuration (OLA users will use the OLA dialogs and Classic users will need to edit the appropriate configuration member). Note also that if these parameters are modified in the configuration and the change is activated, it will not take place until the next sign-on.

See also CMDACTionkey above.

CURESC [Yes|No|ON|OFF]

Default: CURESC No

Defines the method that Session Manager uses to determine when an escape sequence has been entered.

No or OFF causes Session Manager to check for the escape sequence in the first modified field of the terminal display.

Yes or ON enables the escape sequence to be entered into any input field, even if there is already data in that field, provided that the cursor is one position to the right of the escape sequence. The escape sequence may be entered anywhere within the field except in the last position (since the cursor would not then be in the correct position).

If you are running MISER, the original data is replaced in the field after the escape sequence is actioned.

CUT *escape* NFILLC [Yes|No|ON|OFF]

Default: CUT *c NFILLC Yes

Defines the sequence of characters, or function or attention key or lightpen, to start a cut operation using the Cut and Paste feature. It also defines the way in which nulls are interpreted.

escape, which must be delimited by a null or a space when issued, can be from one to eight characters long, PF1-PF24, PA1-3, 'ATTN', 'CLEAR' or 'PEN'. When *escape* is entered, the Session Manager Cut panel is displayed.

By defining the Menu panel appropriately, using a defined input field (variable t_cut) the escape sequence can be modified dynamically. See the PANEL statement in the *Panels, Scripts and Variables* manual.

NFILLC specifies how nulls are interpreted during a cut operation. When specified as Yes any nulls in the cut area are changed to spaces. When No is specified, nulls are used; any gaps between the data are removed.

Note The CUT parameter is affected by parameters CMDACTIONKEY and COMMANDPRFXVAL (see CMDACTionkey on page 32 and COMMANDPRFXVAL on page 33).

DAPPLCheck [Yes|No|ON|OFF]

Default: Yes|ON

A setting of Yes|ON will indicate that a check is made to ensure that the applname entered with the ADDSESS command exists within the Session Manager configuration.

A setting of No|OFF will bypass this check.

When updated, this option comes into effect immediately.

DAPPLESMAUTH [Yes|No|ON|OFF]

Default: Yes|ON

If the E22 exit is installed then a setting of Yes|ON specifies that a call should be issued to the External Security Manager to check if the user has access to the application which has been entered with the ADDSESS command. A setting of No|OFF will bypass this check.

When updated, this option comes in to effect immediately.

```
DEMO [DEMOKEY escape]
    [AUTocopy [Yes|No|ON|OFF]]
    [INTernal [Yes|No|ON|OFF]]
    [DISplay [Yes|No|ON|OFF]]
```

Default: AUTOCOPY Yes
INTERNAL No
Display Yes

Sets defaults for the Demonstration feature.

DEMOKEY *escape*

Defines the sequence of characters, or the function or attention key or lightpen, which causes a screen image to be sent from the demonstrator to the viewers.

If the DEMOKEY parameter is omitted, no demokey is set, and the selective screen copying facility is disabled. This can be overridden on the Demonstration Viewer List. See the chapter 'Demonstration and View Feature' in the *User and Administrator* manual.

AUTOCOPY [Yes|No|ON|OFF]

Determines whether or not output to the demonstrator's screen is automatically sent to the viewers of the demonstration.

If AUTOCOPY=Yes is specified, output screens are automatically sent to the viewers.

If AUTOCOPY=No is specified, only those output screens chosen by the demonstrator (using the demokey) are sent to the viewers. The default is AUTOCOPY=Y.

INTERNAL [Yes|No|ON|OFF]

Determines whether Session Manager internal screens, such as the Demonstration Viewer List, are to be included when automatic screen copying is in effect.

If INTERNAL=Yes is specified, Session Manager internal screens are included. If INTERNAL=No is specified, internal screens are not included. The default is INTERNAL=NO.

DISplay [Yes|No|ON|OFF]

If DISPLAY=Yes is specified, the Demonstration Viewer list is displayed. If DISPLAY=No is specified, then display of this panel is suppressed.

Note The DEMO parameter is affected by parameters CMDACTIONKEY and COMMANDPRFXVAL (see CMDACTionkey on page 32 and COMMANDPRFXVAL on page 33).

DOUBLESC *interval*

Default: DOUBLESC 0

For Session Manager 1.1.10 and higher, this parameter specifies the time interval (in milliseconds) associated with 'double escape' processing (see page 37). Specify a value of 0 (the default) to disable this facility.

The SME command (see page 243) sends the Menu escape key (or sequence) to the last active session.

DSESSRange FROM *fromsessionnumber* TO *tosessionnumber*

Default: DSESSRange FROM 9999 TO 1

When updated, this option comes into effect at next Session Manager signon.

This parameter can be used to specify a range of session numbers to assign to dynamically added sessions. Both *fromsessionnumber* and *tosessionnumber* must be a numeric value between 1 and 9999. So, to specify that users can have a maximum of 10 dynamically added sessions, starting with session number 100 and subsequent dynamic session numbers incremented by 1, you would specify:

```
DSESSRange FROM 100 TO 109
```

To allocate the same session numbers but in a descending order you would specify:

```
DSESSRange FROM 109 TO 100
```

To limit a user to a single dynamically added session which had a session number of 100 then specify:

```
DSESSRange FROM 100 TO 100
```

`ERTIMEOUT` *`seconds`*

Default: `ERTIMEOUT 30`

Specifies a value in the range 0-1440 being the number of seconds the Eclipse Server will wait for a response to a request, after which an error response will be sent to the Client.

When updated, the change comes into effect at next Session Manager signon.

`ESCAPE` *`escape`* `[CURSOR` *`row col`*`]`

Default: `ESCAPE @@`

Defines the sequence of characters, or the function or attention key or lightpen, which causes a return to the Session Manager Menu screen from an active application. *`escape`*, which must be delimited by a null or a space when issued, can be from one to eight characters long, PF1-PF24, PA1-3, 'ATTN', 'CLEAR' (all non-Menu escapes do *not* support this key) or 'PEN'.

Note In this document, the 0x7C (that is, x'7C') character is always presented as the @ sign. It may be displayed as a different character in some non-English code pages. You should enter the appropriate 0x7C character symbol for the code page you are using.

By defining the Menu panel appropriately, using a defined input field (variable `t_esc`) the escape sequence can be modified dynamically. See the `PANEL` statement in the *Panels, Scripts and Variables* manual.

Note If the 'CLEAR' key is used as the Menu escape key (either at logon time or dynamically by the user) then previously typed data will be lost. Also, since CLEAR key escape processing makes use of the MISER facility to reduce data traffic to and from applications, Session Manager will automatically ensure that MISER is active for all sessions.

If the `CURSOR` subparameter is specified, the escape only occurs when the cursor is in the specified row and column. This applies to PF key escape ids, the lightpen id and escape character sequences, but not to PA key escape ids, the Attn key or the Clear key. The effect is to stop Session Manager from interpreting valid input to the session as an escape sequence, but still allows escape from a session.

'Double escape' processing

For Session Manager 1.1.10 and higher, 'double escape' processing is possible – that is, when a Menu escape key is pressed twice within a specified time interval (see '`DOUBLESC`' on page 36) then:

- The first escape will escape to the Menu (previously typed data will be lost).
- The second escape will make the 'escaped from' session visible again and send the escape key (or sequence) to the application.

Note The `ESCAPE` parameter is affected by parameters `CMDACTIONKEY` and `COMMANDPRFXVAL` (see `CMDACTionkey` on page 32 and `COMMANDPRFXVAL` on page 33).

`ESMOLAGROUP [Yes|No|ON|OFF]`

Default: `ESMOLAGROUP No`

A setting of `Yes|ON` will restrict the list of Local USER definitions to userids which reside in the OLA users default External Security Manager group.

A setting of `No|OFF` will not restrict the list.

When updated the change comes into effect at next Session Manager signon.

`EUTIMEOUT minutes`

Default: `EUTIMEOUT 0`

Specifies a value in the range 0-1440 being the number of minutes of inactivity after which an Eclipse User will be logged off.

When updated the change comes into effect at next Session Manager signon.

`FORWARD escape`

Default: `FORWARD >>`

Defines the sequence of characters, or the function or attention key or lightpen, which causes Session Manager to transfer control forwards through the active applications for a Menu panel. *escape*, which must be delimited by a null or a space when issued, can be from one to eight characters long, PF1-PF24, PA1-3, 'ATTN', 'CLEAR' or 'PEN'.

By defining the Menu panel appropriately, using a defined input field (variable `t_fwd`) the escape sequence can be modified dynamically. See the `PANEL` statement in the *Panels, Scripts and Variables* manual.

Note The `FORWARD` parameter is affected by parameters `CMDACTIONKEY` and `COMMANDPRFXVAL` (see `CMDACTionkey` on page 32 and `COMMANDPRFXVAL` on page 33).

`HCPROF hc_profilename`

Specifies the name of a hardcopy profile. The *hc_profilename*, which may be referenced by the variable `t_hcprof`, may be from one to eight alphanumeric characters.

The hardcopy profile determines the set of options, such as the heading and trailing lines and the print routes, which a user may select for a screen hardcopy.

`HCREQUEST escape [CURSOR row col]`

Defines the sequence of characters, or function or attention key or lightpen, which invokes a screen hardcopy. *escape*, which must be delimited by a null or a space when issued, may be from one to eight characters long, PF1-PF24, PA1-3, 'ATTN', 'CLEAR' or 'PEN'.

By defining the Menu panel appropriately, using a defined input field (variable `t_hcmd`) the escape sequence can be modified dynamically. See the `PANEL` statement in the *Panels, Scripts and Variables* manual.

If the CURSOR subparameter is specified, the hardcopy only occurs when the cursor is in the specified row and column. This applies to PF key and lightpen escapes as well as to character sequences. The effect is to stop Session Manager from interpreting valid input to the session as a hardcopy request, but still allows a hardcopy to be taken.

An alternative way to take a screen hardcopy is to use the HARDCOPY command. See 'General user commands' on page 222 for further details of this command.

Note The HCREQUEST parameter is affected by parameters CMDACTIONKEY and COMMANDPRFXVAL (see CMDACTionkey on page 32 and COMMANDPRFXVAL on page 33).

IDLEDisc *minutes* [DIRECTION IN|INOUT][WARN [YES|No|ON|OFF]]

Default: IDLEDISC 0 DIRECTION IN WARN NO

Specifies the interval, in minutes, since the last input was received from the terminal, after which the user is to be disconnected.

minutes may be any value in the range 0-1440 minutes. 0 means that users are not disconnected from Session Manager. If any other value is specified, it should be less than the value specified on the IDLELOGOFF parameter on this statement, (unless the AFTERDISC subparameter is specified on the IDLELOGOFF parameter).

DIRECTION controls whether (activity on) input alone, or on both input and output datastreams will inhibit a timeout. If DIRECTION IN is in effect then the interval counter is not reset by screen updating.

If WARN=Yes is in effect, a warning message is issued to the user when the idle interval has expired. If the user presses the Enter key in the next 30 seconds, the disconnect does not take place.

When WARN=No is specified, disconnection takes effect as soon as the idle interval expires.

IDLELOCK *minutes* [DIRECTION IN|INOUT]

Default: IDLELOCK 0 DIRECTION IN

Specifies the interval, in minutes, since the last input was received from the terminal, after which the 'lock' internal session is activated.

minutes may be any value in the range 0-1440 minutes. 0 means that the timer-based 'lock' function is not used.

DIRECTION controls whether (activity on) input alone, or on both input and output datastreams will inhibit a timeout. If DIRECTION IN is in effect then the interval counter is not reset by screen updating.

When updated, this option comes into effect at the next terminal enter.

IDLELogoff *minutes* [AFTERdisc] [DIRECTION IN|INOUT] [WARN [Yes|No|ON|OFF]]

Default: IDLELOGOFF 0 DIRECTION IN WARN NO

Similar to IDLEDISC, this parameter specifies the interval, in minutes, since the last input was received from the terminal after which the user is to be logged off Session Manager.

minutes may be any value in the range 0-1440 minutes. 0 means that the automatic logoff does not occur. If any other value is specified, it should be greater than the value specified for the IDLEDISC parameter on this statement (unless AFTERDISC is specified).

A user can be logged off even when in a disconnected state. Note that the IDLEDISC interval and the IDLELOGOFF interval run concurrently, except when the AFTERDISC subparameter is specified, in which case the IDLELOGOFF interval starts when the IDLEDISC interval has expired.

DIRECTION controls whether (activity on) input alone, or on both input and output datastreams will inhibit a timeout. If DIRECTION IN is in effect then the interval counter is not reset by screen updating.

If WARN=Yes is in effect, a warning message is issued to the user when the idle interval has expired. If the user presses the Enter key in the next 30 seconds, the forced sign off does not take place.

When WARN=No is specified, sign off takes effect as soon as the idle interval expires.

Note If SHAREDISC or SHARESESS are being used to implement the Shared User Facility (see the *Installation and Customization* manual) then operation of IDLELOGOFF alters slightly.

If the logoff interval has expired for the primary user (the first terminal to be logged on with a given userid) then the primary user will be logged off if no secondary users (terminals subsequently logging on with the same userid) are connected. If any secondary users are connected then the primary user will be disconnected rather than logged off.

If the logoff interval has expired for a secondary user then only that secondary user will be logged off if the primary or other secondary user is connected. However, if the primary or other secondary users are disconnected then those other users will also be logged off.

LANGuage *languageid*

Language Packs (LPs) contain all the Panels, Scripts, Text and Messages that have been translated. For more information on LPs and national language support in Session Manager, see the *Installation and Customization* manual.

The LANGUAGE parameter is used to specify the language in which screen text is displayed for a particular user. Two characters, consisting only of letters of the alphabet or numerals, must be specified for *languageid*, and the specified id should match the LP id.

For additional customer-defined messages and panels:

- A language id can be allocated to appropriate Session Manager messages using the LANGUAGE parameter of MESSAGE statements (for details, see 'MESSAGE statement' on page 163)
- A language id can be allocated to appropriate Session Manager panels using PANEL, PHEADER, PCONTENT, PTRAILER and PPROCESS definitions (for details, see the *Panels, Scripts and Variables* manual).

If a LANGUAGE value is not specified, or an invalid value for *languageid* is specified (that is, there is no corresponding LP) then the default LP (English) is used.

LOGDISC EXIT|SIGNON

Default: LOGDISC EXIT

Specifies the screen to be displayed when the LOGOFF and DISCONNECT commands are issued, or when the interval specified by either IDLEDISC or IDLELOGOFF expires.

EXIT causes the terminal to be dropped from Session Manager, and the VTAM Logon screen is consequently displayed. SIGNON causes the appropriate Session Manager Signon panel to be displayed.

The setting of the LOGDISC parameter can be overridden on the LOGOFF and DISCONNECT commands when issued at a terminal.

To exit from Session Manager, the LOGOFF or DISCONNECT command should be issued with the EXIT operand.

MENU *panelname*

Default: MENU MENU

Specifies the name of the Menu panel id, (menu screen name), into which the data defined for a user is to be formatted. Any combination of up to eight alphanumeric characters is permissible, but the *panelname* specified must match a panel name defined on the PANEL statement.

t_panel is the variable that contains the menu panel name and in certain circumstances may be updated.

MOBILE [Yes|No|ON|OFF] [LOCK [Yes|No|ON|OFF]]

Default: MOBILE Yes

Specifies whether a Menu screen can be acquired by a terminal other than the originator. When specified with the Yes option, any Menu screen may be acquired by a terminal other than the one which originally accessed the screen.

If No or OFF is specified, then no Menu screen can be acquired from another terminal.

LOCK [Yes|No|ON|OFF]

Applies only when MOBILE=Yes is specified, or implied.

When LOCK=Yes is specified, no Menu screens may be acquired by another terminal until the LOCK OFF or LOCK NO command is issued at the original terminal.

When LOCK=No is specified, then a Menu screen may be acquired by another terminal without any intervention from the original terminal.

MSGID [Yes|No|ON|OFF]

Default: MSGID No

Specifies the format of messages when displayed at a terminal. Yes or ON causes messages to be displayed in full at all terminals with the code and text.

When No or OFF is in effect, only the message text is displayed.

The Session Manager user-associated variable t_msgid may be used to refer to the value of the message format; for details, see the product's *Panels, Scripts and Variables* manual.

`NCSESC [Yes|No|ON|OFF]`

Default: `No` - Specified escape sequences will be case sensitive.

This parameter determines whether escape sequences will be case-sensitive.

A value of `NCSESC Yes` will make escape sequences non-case-sensitive.

A value of `NCSESC No` (the default) will make escape sequences case-sensitive.

`OLACLASS` *`ola-class`*

Specifies the user's OLA security class. When a user attempts to invoke a particular OLA task (for example, to display a menu, or list certain attribute values), the security function checks to see if the user's OLA security class allows that task to be performed. Therefore, the user's OLA security class defines what the user can or can't do. For example, if a particular user is authorized to execute just a single item on a particular menu then the menu display is bypassed and the panel for the single item is shown.

The Session Manager configuration as shipped comes with these OLA security classes defined; for details, refer to the *Online and Batch Administration* manual:

OLACLASS	Description
IM	Implementor
BT	Batch Administrator
AD	Administrator
LA	Local Administrator
SU	Super User
US	User
NO	No Access

Notes

a All users in the same OLA security class have the same capability.

b For the Session Manager configuration as shipped, only an Implementor can view *and* update `OLACLASS` attributes. However, an Implementor could update `OLACLASS` attributes such that other OLA classes have this capability.

c OLA security settings are used to define completely what the user is allowed to do. For details, see the *Online and Batch Administration* manual.

d The Session Manager user-associated variable t_security_class may be used to refer to the value of a user's OLA security class; for details, see the product's *Panels, Scripts and Variables* manual.

e The OLA security class will be determined by an ESM if an `OLARESNAME` is specified on the `SECURITY` parameter of the `SYSTEM` statement. In this case, any `OLACLASS` settings in the configuration will be ignored. For further details, see 'SYSTEM statement' on page 87.

`PASTE` *escape* `NFILLP [Yes|No|ON|OFF] SPILLW [Yes|No|ON|OFF]`

Default: `PASTE *p NFILLP Yes`

Specifies the sequence of characters, or function or attention key or lightpen, which initiates a paste operation to paste data collected by a previous cut operation.

escape, which must be delimited by a null or a space when issued, can be from one to eight characters long, PF1-PF24, PA1-3, 'ATTN', 'CLEAR' or 'PEN'. When *escape* is entered, the Session Manager Paste panel is displayed.

By defining the Menu panel appropriately, using a defined input field (variable `t_paste`) the escape sequence can be modified dynamically. See the `PANEL` statement in the *Panels, Scripts and Variables* manual.

The `NFILLP` subparameter specifies how nulls to the left of the paste area are treated. If `Yes` is specified, any nulls are converted to spaces. `No` or `OFF` causes no conversion to occur; pasted data moves to the left.

When `SPILLW` is specified, any spillage during a paste operation causes a message to be displayed in the paste panel, but the paste operation continues if possible. Spillage occurs when the data being pasted is larger than the paste area.

`SPILLW=No` suppresses the warning message, and the paste operation continues.

Note The `PASTE` parameter is affected by parameters `CMDACTIONKEY` and `COMMANDPRFXVAL` (see `CMDACTionkey` on page 32 and `COMMANDPRFXVAL` on page 33).

`PREVIOUS` *escape*

Default: None.

Defines the sequence of characters, or the function or attention key or lightpen, which causes Session Manager to transfer control back to the previous active session for a Menu panel.

Notes

a To transfer control back to the original active session, issue *escape* again; that is, use *escape* repeatedly to 'toggle' between the current active session and the previous active session.

b See also '`BACKWARD`' on page 31, which causes Session Manager to transfer control backwards through the active applications.

The *escape*, which must be delimited by a null or a space when issued, can be from one to eight characters long, PF1-PF24, PA1-3, 'ATTN', 'CLEAR' or 'PEN'.

By defining the Menu panel appropriately, using a defined input field (variable `t_prev`) the escape sequence can be modified dynamically. See the `PANEL` statement in the *Panels, Scripts and Variables* manual.

`PULL` *escape*

Default: `PULL /`.

Specifies the sequence of characters, or function or attention key or lightpen, for retrieving saved, or 'pushed', screens. *escape*, which must be delimited by a null or a space when issued, can be from one to eight characters in length, PF1-PF24, PA1-3, 'ATTN', 'CLEAR' or 'PEN'.

By defining the Menu panel appropriately, using a defined input field (variable `t_pull`) the escape sequence can be modified dynamically. See the `PANEL` statement in the *Panels, Scripts and Variables* manual.

The chapter 'The Push-Pull Facility' in the *User and Administrator* manual describes the use of this facility.

Note The `PULL` parameter is affected by parameters `CMDACTIONKEY` and `COMMANDPRFXVAL` (see `CMDACTionkey` on page 32 and `COMMANDPRFXVAL` on page 33).

`PUSH` *escape*

Default: `PUSH />`

Specifies the sequence of characters, or function or attention key or lightpen, to be used for 'pushing' a screen; that is, saving it dynamically in storage for retrieval by the 'pull' command. *escape*, which must be delimited by a null or a space when issued, can be from one to eight characters long, PF1-PF24, PA1-3, 'ATTN', 'CLEAR' or 'PEN'.

By defining the Menu panel appropriately, using a defined input field (variable `t_push`) the escape sequence can be modified dynamically. See the `PANEL` statement in the *Panels, Scripts and Variables* manual.

Note The `PUSH` parameter is affected by parameters `CMDACTIONKEY` and `COMMANDPRFXVAL` (see `CMDACTionkey` on page 32 and `COMMANDPRFXVAL` on page 33).

`PUSHLimit` *push-limit*

Default: `PUSHLIMIT 12`

Defines the maximum number of screens that may be saved by each Session Manager user at any one time using the push escape sequence.

push-limit can be any value in the range 0-999. If this parameter is omitted, the default is 12. Specifying `PUSHLIMIT=0` on the `SYSTEM` statement suppresses the Push-Pull facility for all users, except those with an override value set on the `PROFILE`, `USER` or `TERMINAL` statement.

Since each screen occupies storage, this parameter may prove useful in preventing problems arising from lack of storage, for example, when screens are frequently saved, but not deleted.

`REBIND [Yes|No|ON|OFF]`

Default: `REBIND Yes`

The characteristics of a physical terminal may be determined from the bind image returned to Session Manager at signon time.

When `REBIND=Yes`, Session Manager attempts to create a session with the terminal reflecting the terminal's characteristics, regardless of any logmode entry name supplied at signon.

When REBIND=No, Session Manager does not attempt to alter the bind parameters specified in the logmode entry supplied with the signon request. REBIND=No is usually specified for terminals and PCs which do not completely emulate the IBM equivalent.

Further information on the effects of using REBIND may be found in 'Logmode Entry Selection and REBIND Implications' in the 'Setting up applications' chapter of the *Installation and Customization* manual.

RECORDLimit *record-limit*

Default: RECORDLIMIT 12

Defines the maximum number of records per Menu screen for recording terminal input and output data streams using the Record-Replay facility. *record-limit* may be any value in the range 0-999.

Since the records use storage, it may help to impose an overall restriction on the SYSTEM statement, which can then be overridden for selected users.

RECOVERYLevel High | Intermediate | None

Default: None

Specifies the Sysplex recovery level. Defines the recoverability of forward sessions to the application when the ACB used is being opened. The parameter only takes effect when an ACB is opened. If the ACB is already opened when the session is started then any change to this parameter will have no effect. This parameter can be defined as a Common End User parameter on the USER, TERMINAL, PROFILE or SYSTEM statements or on the APPL statement. If defined as a Common End User parameter and set to HIGH or INTERMEDIATE then this will override the APPL statement. If NONE is specified then the APPL setting applies.

HIGH = Recover the application session if the owning (target) Session Manager node fails or the Session Manager Controller fails. Can only be used with SYSPLEXTYPE I instances.

INTERMEDIATE = Recover the application session if the owning (target) Session Manager node fails. Can only be used with SYSPLEXTYPE I instances.

NONE = No recovery will be attempted for this application. Can be used with SYSPLEXTYPE I or SYSPLEXTYPE N instances.

See the 'Parallel Sysplex support' chapter in the *Installation and Customization* manual for further information.

REPLAY ASRECord|FMTopts

Default: REPLAY ASRECord

The Record/Replay facility provides three playback formats for recorded terminal input and output data streams; 'As Recorded', 'Dump' and 'Data Stream Analysis'. Specifying REPLAY=ASRECORD suppresses the latter two formats, while specifying REPLAY=FMTOPTS enables any of the three formats to be selected.

Since the Dump and Data Stream Analysis are provided as problem solving tools, it is recommended that ASRECORD should be the SYSTEM option, overridden when required for Technical Staff.

The chapter 'The Record-Replay Facility of the *User and Administrator* manual contains a full description of this facility.

`SENDCDonsrd [Yes|No|ON|OFF]`

Default: `SENDCDonsrd Yes`

This parameter may be applied when using SNA 3270 LU2 protocols, but not LU0 protocols although it is benign. The parameter will cause an SNA Change Direction data stream to be sent if a short data stream is received in response to a Session Manager 3270 Read Buffer command. If a short data stream is received, a race condition may occur when connecting to Session Manager using components that emulate SNA terminals, such as TN3270 servers or other session managers.

This parameter allows the 'terminal' to be given session direction, enabling it to send the Read Buffer reply correctly.

Note IBM Communication Server TN3270 generally only operates as an LU2 when using TN3270E protocols. If standard TN3270 protocols are used, an LU0 session is initiated.

`SESSAUTOSAppl [Yes|No|ON|OFF]`

Default: No

This parameter enables sessions to be Autostarted based on application name (or command). If it is set to `Y` then OLA will generate a `USER` statement parameter `SESSAUTOS A` (if the session specifies an application) or `SESSAUTOS C` (if the session specifies a command).

If neither an application nor a command is specified, or if `SESSAUTOS N` is in effect, then OLA will generate a `SESSAUTOS T` (if the session has a `SESTYPE` associated with it); otherwise it will generate a `SESSAUTOS N`.

See also `SESSAUTOS` on page 129.

Note It is recommended that you allow users only one session with each application, otherwise the effect of Autostarting a session may not necessarily be as you expect.

`SESSPRIAppl [Yes|No|ON|OFF]`

Default: `No`

This parameter enables sessions to be prioritized based on application name (or command). If it is set to `Y` then OLA will generate a `USER` statement parameter `SESSPRI A` (if the session specifies an application) or `SESSPRI C` (if the session specifies a command).

If neither an application nor a command is specified, or if `SESSPRIAPPL N` is in effect, then OLA will generate a `SESSPRI T` (if the session has a `SESTYPE` associated with it), otherwise it will generate a `SESSPRI N`.

Notes It is recommended that you allow users only one session with each application, otherwise the effect of prioritizing a session may not necessarily be as you expect.

This parameter will be ignored for users who have sub-menus configured.

See also `SESSPRI` on page 130.

SHARE [Yes|No|ON|OFF]

Default: SHARE No

The Shared User facility enables multiple users to sign on with the same userid (see the *Installation and Customization* manual). This is achieved by appending a 'user qualifier' to the signon userid. Depending on the value of parameter MULTUSER on the SYSTEM statement, either the 'user qualifier' is created from the last four characters of the terminal LU name, or it is an eight-digit number in the range 1-99999999.

Prior to Session Manager 1.1.10, the Shared User facility was activated by setting the Multiple Users flag to 'M' in the E21 Exit (or the variable ec21_ruser in the E21 exit script) – see the *Installation and Customization* manual. For Session Manager 1.1.10 and higher, you can specify SHARE Yes (or SHARE ON) to define a user as a 'shared user'.

The Session Manager system variable t_share may be used to refer to the value of SHARE; for details, see the product's *Panels, Scripts and Variables* manual.

The RECONANYTERM parameter (see page 100) of the SYSTEM statement enables any 'SHARE=Y' user to reconnect to a disconnected user's sessions with the same name from a different terminal.

Only one of SHARE, SHAREDISC and SHARESESS should be set to Yes at any one time.

SHAREDISC [Yes|No|ON|OFF]

Default: SHAREDISC NO

This parameter enables a single userid to be used to sign on to more than one terminal at once, with each terminal sharing the same menu and session list.

When a SHAREDISC user signs on and there is no other terminal signed on (or disconnected) with the same userid then this becomes the primary user. Subsequent terminals signing on with the same userid become secondary users. All sessions started by primary or secondary users are primary user sessions. If a secondary user takes over a session from a primary user, the primary user will be disconnected. If a primary user takes over a session from a secondary user, the secondary user will be logged off.

For example, if USERA, with SHAREDISC set to YES, signs on to terminal TERM1 and then signs on to terminal TERM2, both terminals will display the same menu and session list. If a session is started on TERM1 then the session will be displayed as active on both terminals. In this case the owner of the session is TERM1, and TERM1 will remain the owner even if the user on TERM1 escapes and starts another session.

If, however, the user on TERM2 selects a session owned by TERM1, then TERM1 is disconnected. Any data that has been typed but not entered on the original terminal will be lost. Depending on the setting of the LOGDISC parameter (see page 41) for the initial terminal (TERM1), either the Session Manager signon screen will be displayed or the terminal will be disconnected.

If a LOGOFF command is entered at any terminal signed on to by a SHAREDISC user then all other terminals signed on with that userid will also be signed off. If, however, a DISCONNECT command is entered at any terminal signed on to by a SHAREDISC user then only that terminal is disconnected.

Only one of SHARE, SHAREDISC and SHARESESS should be set to Yes at any one time.

```
SHARESESS [Yes|No|ON|OFF]
          [MAXUSRLOGIN nnnn]
```

Default: SHARESESS No

This parameter enables a single userid to be used to sign on to more than one terminal at once, with each terminal sharing the same menu and session list. If one terminal takes over a session from another, the first will lose control of the session, and receive a message to that effect.

For example, if USERA, with SHARESESS set to YES, signs on to terminal TERM1 and then signs on to terminal TERM2, both terminals will display the same menu and session list. If a session is started on TERM1, the session will be displayed as active on both terminals. In this case the owner of the session is TERM1, and TERM1 will remain the owner even if the user on TERM1 escapes and starts another session.

If however, the user on TERM2 selects a session owned by TERM1, then TERM2 will take over the ownership of the session. Any data that has been typed but not entered on the original terminal will be lost. If the session is still visible on TERM1, the terminal will be returned to the menu with the following message:

```
ISZ0482W Session nnnn taken over by uuuu at tttt zzzzzzz
```

where *nnnn* is the session number, *uuuu* is the userid of the new owner (always the same as the userid of this user) and *tttt zzzzzzz* are the terminal type and terminal name respectively of the new owner.

This same message will also be written to the audit log, prefixed with the userid and the terminal type and name of the original owner of the session.

If a LOGOFF command is entered at any terminal signed on to by a SHARESESS user then all other terminals signed on with that userid will also be signed off. If, however, a DISCONNECT command is entered at any terminal signed on to by a SHARESESS user then only that terminal is disconnected.

When a SHARESESS user signs on and there is no other terminal signed on (or disconnected) with the same userid then this becomes the primary user. Subsequent terminals signing on with the same userid become secondary users. All sessions started by primary or secondary users are primary user sessions.

Note Only one of SHARE, SHAREDISC and SHARESESS should be set to Yes at any one time.

The subparameter MAXUSRlogin specifies the maximum number of SHARESESS logins permitted for this user. *nnnn* is a value up to 9999.

```
SIMRecon [Yes|No|ON|OFF]
```

Default: SIMRecon NO

This parameter may be used with all the user's CLOSEDISC sessions.

Normally, when the user terminates a CLOSEDISC session, the user is returned to the Session Manager sign-on panel, from which they can re-enter their user and password to reconnect to their disconnected Session Manager menu.

The SIMRecon parameter requests that the existing security access of a user be maintained for later reconnection, following a CLOSEDISC allowing a direct reconnect to their disconnected Session Manager menu, bypassing the Session Manager sign-on panel.

The SIMRecon Yes or SIMRecon ON values will only apply if all of the following conditions are true:

- On return from a CLOSEDISC session.
- When LOGDISC SIGNON is active.
- The user is signed in, rather than a SIGNON NO terminal.
- The CLOSEDISC application is not in the current Session Manager instance.

If any one of these conditions is not true, then on termination of the CLOSEDISC session, the user will see a signon panel as if SIMRecon No was in effect.

SPYABLE [TELL|Yes|No|ON|OFF]

Default: SPYABLE TELL

Specifies whether users can be spied upon. SPYABLE TELL specifies that all users may be spied upon, provided that the user being spied upon accepts the spy operation. SPYABLE TELL also causes a message to be sent to the target user when the spy operation is terminated.

SPYABLE TELL is a prerequisite for the SPYOFF command, which is documented later in this manual.

SPYABLE=Yes specifies that all users may be spied upon without the need for acceptance of the spy operation. SPYABLE=No prevents a user from being spied upon.

Note A shared user is not SPYABLE. Use SPY LU if sharing users. See 'SPY operator command' on page 263 for details.

SPYGROUP *groupname*|*

Specify the name of a group of users, or terminals that may be spied upon. The *groupname* must be defined on a GROUP statement. An asterisk means that all users or terminals specified on **any** GROUP statement can be spied upon.

Note that this means that a user with this option can at any time view the current screen of any of the users, profiles, or terminals in the group identified by the *groupname*. Therefore, due consideration should be given to any security exposure likely to arise before allowing a user this privilege.

TERMERROR EXIT|SIGNON [DISC|LOGOFF|SIGNOFF]

Default: TERMERROR EXIT DISC

Specifies the panel to be displayed in the event of Session Manager encountering an irrecoverable error when attempting to communicate with the terminal.

EXIT causes the terminal to be dropped from Session Manager and the VTAM Logon screen to be displayed.

SIGNON causes the appropriate Session Manager Signon screen to be displayed.

DISC causes the terminal to be disconnected from Session Manager.

LOGOFF causes the user to be logged off. If SHAREDISC or SHARESESS are being used, the operation of LOGOFF is slightly altered. Refer to the IDLELOGOFF parameter on page 39 for details. Also, see the LOGDISC parameter on page 41.

SIGNOFF has the same effect as LOGOFF.

TRANSTab *transtable*

Specifies the translation table to be used, if appropriate. Translation tables are used for parsing Session Manager commands, and for removing invalid characters from panel displays.

The *transtable* specifies a unique one to eight character name as specified on the TRANSTABLE statement. Refer to the definition of the TRANSTABLE statement to see the default settings used by Session Manager.

USERDATA1 *data* - USERDATA5 *data*

These five 32-byte user fields (and corresponding read-only variables *t_userdata1* - *t_userdata5*) are for general use by the user. The 5th field will be displayed as asterisks in an Online Administration display. There is no validation on the data within these fields. The variables are available for use in scripts, exits and so on, and are not used internally by Session Manager. The variables are not available when running a session or command on a remote system.

WINDSCRIPT *scriptname*

Default: WINDSCRIPT WINDOW

Specifies the name of the window script to be executed if no script name is supplied on the WINDOWS command. Refer to 'WINDOWS user command' on page 244 for details of the WINDOWS command, and to the *Panels, Scripts and Variables* manual for information about window scripts.

Common session parameters

Except where stated otherwise, common session parameters are applicable to the `PROFILE`, `SYSTEM`, `USER`, `TERMINAL and LU` statements.

`ACB` *`override_acbname`*

(Not applicable to the `SYSTEM` statement.) Specifies the name of the ACB to be used when establishing a session with the application. It can be up to a maximum of eight bytes long. This parameter provides a method of overriding the normal Session Manager ACB allocation process which is fully described in 'Choosing an ACB' section in the 'Setting up applications' chapter of the *Installation and Customization* manual.

The *`override_acbname`* may contain the special character '?' (question mark symbol) in any number of positions, and Session Manager substitutes these with the corresponding characters from the LU name of the terminal. For example, if the LU name is 'L010ABCD', and an *`override_acbname`* of '????K1' is specified on this parameter, the ACB name 'L010K1' would be generated.

Although it is not a recognized key in all countries, a hash (#) is allowed in place of a question mark, for backwards compatibility.

Note Since '?' is itself a Session Manager quote character, if it is used in an ACB value then the whole value should be placed within another quote character, for example "AC????".

The *`override_acbname`* may also contain the special character '%' (percent symbol) in any number of positions in order to substitute characters from the userid.

See also the section 'The STOPINH and STOP_INHERIT keywords' on page 72.

```
ADDSID text|variable-name AT row col
    [COND text AT row col]
    [MODEL model-number]
```

Default: None

Enables a character string to be added to any session output screens. Optional tests may be made for the presence of a second string (which is the output from the application) using the `COND` subparameter. In addition, a particular terminal model type (for example, a model 5 terminal) can be tested for.

The character string may be a Session Manager variable, or it may be literal text. In each case, the string may contain a maximum of 80 characters; the `COND` subparameter can contain up to 256 characters.

`AT` *`row col`* specifies the row and column where the character string is to be placed on the screen display. When used with `COND` it specifies the place on the screen display where the `COND` string is to be found. The 3270 order SBA will generally be placed immediately before the string on the same row. However, if Col 1 is specified, the SBA occupies Col 1 and the text starts in column 2. The values of *`row col`* **must** be explicitly specified; there is no default.

`MODEL` *`model-number`* specifies the terminal model type the screen must be in for this definition to take effect.

As many ADDSID parameters as required may be specified for a session. If a definition exists without the optional COND and MODEL specified, it will be used if none of the conditional definitions are satisfied. If multiple conditional definitions can be satisfied, the last one tested for is used. The specification is ignored if PCTRANSFER is set to ON.

See also the section 'The STOPINH and STOP_INHERIT keywords' on page 72.

ALARM [Yes|No|ON|OFF]

Default: ALARM No

Specifies whether the terminal alarm is to automatically sound when a non-visible application is modified. ALARM=Yes should be specified for the terminal alarm to sound.

When ALARM=No is specified, the alarm is not sounded.

If the visible session has PCTRANSFER=Yes specified, the alarm is not sounded.

ALLOWEscape [Yes|No|ON|OFF]

Default: ALLOWESCAPE Yes

Specifies whether or not it is possible to escape back to the Menu screen from an application. ALLOWESCAPE=Yes enables escape.

Note The default setting of this parameter is an exception to normal Session Manager default processing. The default setting for ALLOWESCAPE is to be set ON if the parameter is omitted, or if the keyword only is specified.

ALLOWESCAPE=No prevents escape.

APPLSEL *search-text*

Enables invisible sessions to become visible when an application writes to the terminal and the *search-text* is in the data stream.

If the *search-text* is not found, then unless the ONREAD or ONWRITE parameters have been specified, no session swap occurs.

The *search-text* is case sensitive. See the ONWRITE and ONREAD parameters on page 60.

See also the section 'The STOPINH and STOP_INHERIT keywords' on page 72.

AUTOSCRIPT *scriptname*

Optionally defines the name of a script passed to an application associated with a session id, or a PF key, when the *escape* defined using the AUTOSEQ parameter is entered. (See page 31).

Scripts are generated using the SCRIPT statement described in the *Panels, Scripts and Variables* manual. The *scriptname* specified on the AUTOSCRIPT parameter must appear on a SCRIPT statement.

See also the section 'The STOPINH and STOP_INHERIT keywords' on page 72.

`AUTOSTART [Yes|No|ON|OFF]`

Default: `AUTOSTART No`

Specifies whether the session is to be started automatically when a user accesses Session Manager. `AUTOSTART=Yes` starts the session. `AUTOSTART=No` suppresses automatic startup.

`BLANKSCript` *`scriptname`*

Default: none.

`scriptname` identifies the session script to be invoked in a situation where an application session is being made the visible session, MISER is not in effect, but Session Manager is unable to determine what should be written to the screen. If this parameter is not specified then no script will be invoked, and the user will be presented with a blank screen.

A sample script (`ISZBLNKS`) and panel (`ISZBLNKP`) are supplied with the product. For their base member location see the 'Handling upgrades and Functional Enhancement PTFs' section in the 'Post-installation configuration issues' chapter in the *Installation and Customization* manual. If you wish to use these samples then please take a copy and rename them, to avoid clashes in future upgrades.

When updated, this option comes into effect at next session selection or at next signon.

See also the section 'The STOPINH and STOP_INHERIT keywords' on page 72.

`BRDVAR` *`text`*

Specifies a broadcast/message identifier for any session. This then enables a user to receive a broadcast or message which has been sent using either of the following commands:

`BRO` *`'message'`* `BRDVAR` *`id-string`*

`MSG` *`'message'`* `BRDVAR` *`id-string`*

The session does not need to be active for the message to be received. See 'Privileged user commands' on page 245 for details of the `BROADCAST` command.

See also the section 'The STOPINH and STOP_INHERIT keywords' on page 72.

`COMPress [Yes|No|ON|OFF]`

Default: `COMPRESS Yes`

Specifies whether screens received from applications and screens stored when switching between sessions; that is, escape screens, are compressed. `COMPRESS=Yes` causes compression. The saving from this can be evaluated using the `QUERY STATS` command.

If `PCTRANSFER` is set to `ON`, then a `COMPRESS=Yes` setting is ignored; compression is not performed.

`CONCEAL [No|Yes|OFF|ON]`

Default: `CONCEAL NO`

Specifies if the User has chosen to hide the session on their Session Manager menu. User-concealed sessions can subsequently be revealed or concealed by the menu commands `REVEAL` and `CONCEAL`.

Note This parameter will be ignored for users who have sub-menus configured.

When updated this option comes into effect immediately.

`CONDLOGOFF [Yes|No|ON|OFF]`

Default: `CONDLOGOFF No`

Determines whether a `TERMSESS UNCOND` (if `CONDLOGOFF` set to `No`) or a `TERMSESS COND` (if `CONDLOGOFF` set to `Yes`) is sent to an application when it is reset by Session Manager. On occasion, CICS® can respond with an error if a `TERMSESS UNCOND` is sent to it – in this event use `CONDLOGOFF Yes`.

`DATA` *`logon-data`*

Default: None

(Not applicable to the `SYSTEM` statement.) Specifies a string of up to 256 alphanumeric characters to be passed as data to the application when it is initiated, that is, at the start of the session. Substitution variables may be embedded in the string.

The chapter 'Session Manager Variables' in the *Panels, Scripts and Variables* manual lists the valid variables and 'Variable substitution in control statements' on page 17 of this manual gives the rules that apply. If the string contains any Session Manager keywords, or embedded spaces, the complete string should be enclosed in delimiters.

The DATA parameter can contain the Uniform Resource Locator (URL) for connecting to remote host applications using TCP/IP. See 'Session Manager and TCP/IP' in the *Installation and Customization* manual for further details of specifying TCP/IP sessions.

See also the section 'The STOPINH and STOP_INHERIT keywords' on page 72.

`DESCription` *`description`*

Specifies a description which is to be associated with an application or command session. Its maximum length is 256 bytes. However, it may not be desirable to have a description covering several lines on the Menu panel, and a short, concise description may give a more readable display.

If the description contains any Session Manager keywords or embedded spaces, the complete string must be enclosed in delimiters to avoid ambiguity.

See also the section 'The STOPINH and STOP_INHERIT keywords' on page 72.

`DISCACtive [Yes|No|ON|OFF]`

Default: QUITACTIVE setting

`Yes` or `ON` allows disconnection from the Menu while the session is still active.

`No` or `OFF` rejects a `DISCONNECT` command if the session is active and causes message 393 to be issued.

`DROP_SESSION [Yes|No|ON|OFF]`

Default: `DROP_SESSION No`.

Specifies whether the application session is a dropped session. If `DROP_SESSION` is set to `Yes` it will prevent you from:

- Selecting the session in order to make it visible.

Note Unlike `HIDE YES|ON` sessions, if you create your own menus then you will not have to implement TPSL code to prevent dropped sessions from being displayed.

- Escaping to the session using an escape sequence/program function key.

Unlike `HIDE YES|ON` sessions, the `DROP_SESSION` parameter cannot be used for application build sessions that are controlled by application builder scripts.

The Session Manager session variable `s_dropsess` may be used to refer to the value of `DROP_SESSION` for an application session; for details, see the *Panels, Scripts and Variables* manual. However, after the user exit E22 point all `DROP_SESSION YES|ON` sessions will have been dropped and therefore no longer available to access.

If an E22 exit is active the `DROP_SESSION` parameter will be set regardless of the configuration settings, therefore if you want to conceal any sessions to which the user has access then set the `CONCEAL` common session parameter to `YES`. See the 'Defining security' chapter in the *Installation and Customization* manual and the `SYSTEM SECURITY` parameter on page 103.

`ENDSCript` *`scriptname`*

Optionally specifies the name of a script to be passed to a selected application when the user issues the `END` command for the application, or when the system operator command, `CLOSEDOWN END`, is issued. This script is also passed when the user logs off from Session Manager if `LOGOFF END` is specified for an active application.

It would normally be the case that the final input from the script would issue the logoff for the application. If this is not the case, Session Manager automatically issues a VTAM unconditional logoff or resets the logical device after the script has completed.

Scripts are generated using the `SCRIPT` statement described in the *Panels, Scripts and Variables* manual. The *`scriptname`* specified on the `ENDSCRIPT` parameter must appear on a `SCRIPT` statement.

See also the section 'The STOPINH and STOP_INHERIT keywords' on page 72.

`ENVIRONSCript` *`scriptname`*

Optionally specifies the name of an environment-related script. It is invoked after the `INITSCRIPT` and before the `STARTSCRIPT` and is designed to deal with messages issued by the type of environment in which the session is starting.

This means that all application-specific commands and messages can be handled in the `STARTSCRIPT`, and an `ENVIRONSCRIPT` common to several applications can precede it.

Scripts are generated using the `SCRIPT` statement described in the *Panels, Scripts and Variables* manual. The *`scriptname`* specified on the `ENVIRONSCRIPT` parameter must appear on a `SCRIPT` statement.

See also the section 'The STOPINH and STOP_INHERIT keywords' on page 72.

`HIDE [Yes|No|ON|OFF]`

Default: `HIDE No`.

Specifies whether the application session is a hidden session. If `HIDE` is set to `Yes`, it will prevent you from:

- Selecting the session in order to make it visible.

  Note If you create your own menus then you will have to implement some TPSL code to prevent hidden sessions from being displayed. To do this, use the sample menus as a model.

- Escaping to the session using an escape sequence/program function key.

The `HIDE` parameter could be used for application build sessions that are controlled by application builder scripts.

The Session Manager session variable `s_hidden` may be used to refer to the value of `HIDE` for an application session; for details, see the product's *Panels, Scripts and Variables* manual.

If an E22 exit is active the `HIDE` parameter will be set regardless of the configuration settings, therefore if you want to conceal any sessions to which the user has access then set the `CONCEAL` common session parameter to `YES`. See the 'Defining Security' chapter in the *Installation and Customization* manual and the `SYSTEM SECURITY` parameter on page 103.

`ILU [Yes|No|ON|OFF]`

Default: `ILU No`

This parameter is primarily for AS/400 support of ILU-type sessions, where `ILU` is an 'Initiating Logical Unit'. This type of session is for SNA Primary LU2 Support (SPLS) and improves the AS/400 system's participation in an SNA backbone network. The initial bind from this type of session is received by Session Manager as an unsolicited bind which is not accepted as the session start unless `ILU=Yes` is in effect.

SPLS requires one dependent logical unit (LU) definition and one `ILU` definition in VTAM. The `ILU` can be manually created, or dynamically created, depending on the release of VTAM being used.

`IMSConvert [Yes|No|ON|OFF]`

This parameter may be used to convert Erase Write commands to Erase Write Alternate commands.

If the parameter is set to `YES`, and the terminal is a 3270 model 3, 4 or 5, then the conversion will take place.

This parameter would normally only be used when Session Manager definitions and 3270 models are mismatched.

See also the `IMSCOVERTC` parameter (below) and *`s_imsconvertc`* session variable, documented in the *Panels, Scripts and Variables* manual.

`IMSCONVERTC [Yes|No|ON|OFF]`

This parameter may be used to convert Erase Write commands to Erase Write Alternate commands. If the value is set to Y the processing is similar to configuring `IMSCONVERT Y` although, unlike `IMSCONVERT`, the processing is conditional on the BIND for the session not specifying a default 3270 screen size of a model 2.

If the value is set to `N` then no special processing is performed. The default value, if `IMSCONVERTC` is not specified, is `N`.

See also the `IMSCONVERT` parameter and the *`s_imsconvertc`* session variable, documented in the *Panels, Scripts and Variables* manual.

`INITSCript` *`scriptname`*

Optionally specifies the name of an 'initialization' script to be executed *before* a session is started. The script does not communicate with the application, so any parameters contained in it (such as `INPUT`, `WAITAPPL`, `OUTSCAN`, and so on) are ignored. This type of script is generally used to modify session options before the session begins.

For Session Manager 1.1.10 and higher, the `INITSC` command enables a suitably authorized user to disable or (re-)enable the running of the session 'initialization' script. For details, see page 228.

Scripts are generated using the `SCRIPT` statement described in the *Panels, Scripts and Variables* manual. The *`scriptname`* specified on the `INITSCRIPT` parameter must appear on a `SCRIPT` statement.

See also the section 'The STOPINH and STOP_INHERIT keywords' on page 72.

`INTERNALSess [Yes|No|ON|OFF]`

This parameter specifies that the session is an internal session provided with Session Manager. Internal sessions provided include the Online Administration session, the System Management Menu and the Broadcast Menu. The appearance when `INTERNALSess` is set to `YES` is as follows:

Status	Meaning
`ACTIVE`	Feature is available, and currently active.
`AVAIL`	Feature is available, but not currently active.

The internal sessions specified in the sample profiles provided with Session Manager have `INTERNALSess` set to `YES` by default.

If the Online Administration session is created by the OLA Enabler, it will be created with `INTERNALSess` set to `YES`.

`LOGMODE` *`logmode-entryname`*

(Not applicable to the `SYSTEM` statement.) Overrides the default characteristics of the terminal. The default characteristics are derived from the bind image received from the terminal logon request when the user first establishes the session with Session Manager. The default logmode table entry names and the terminal types to which they apply are listed below.

Entry-name	Terminal type
S3270	Non SNA 3277-1
S3270	Non SNA 3277-2
D4B32781	Non SNA 3278-1
D4B32782	Non SNA 3278-2
D4B32783	Non SNA 3278-3
D4B32784	Non SNA 3278-4
D4B32785	Non SNA 3278-5
D4B3278X	Non SNA 3278-X
D4A32781	SNA 3278-1
D4A32782	SNA 3278-2
D4A32783	SNA 3278-3
D4A32784	SNA 3278-4
D4A32785	SNA 3278-5
D4A3278X	SNA 3278-X
VT220	VT220
VT100	VT100
ANSI	ANSI
LINEMODE	NVT

Additionally, Session Manager is supplied with a logmode table which contains entries for 3278, 3279 and 3193 terminals which have extended attributes as follows.

Entry-name	Terminal type
D4B32792	Non SNA 327x-2
D4B32793	Non SNA 327x-3
D4B32794	Non SNA 327x-4
D4B32795	Non SNA 327x-5

Entry-name	Terminal type
D4B3279X	Non SNA 3279-X
LOGM9DN	Non SNA 3279-D ISZDYNAM queriable
D4A32792	SNA 327x-2
D4A32793	SNA 327x-3
D4A32794	SNA 327x-4
D4A32795	SNA 327x-5
D4A3279X	SNA 3279-X
LOGM9DS	SNA 3279-D ISZDYNAS queriable

Further details on the relationship between the physical terminal and the characteristics held in the logmode entry can be found in the 'Setting up applications' chapter of the *Installation and Customization* manual.

See also the section 'The STOPINH and STOP_INHERIT keywords' on page 72.

`LOGOFF END|RESET`

Default: `LOGOFF RESET`

Specifies the action which Session Manager is to take for any selected session if it is still active when the user logs off from Session Manager.

Specifying `RESET`, causes either a VTAM unconditional logoff for the application to be issued, or the logical device to be reset.

Specifying `END` causes Session Manager to pass the script specified by the `ENDSCRIPT` parameter to the application. It provides a means of passing input to the application without actually issuing the `END` command.

```
MISER [Yes|No|ON|OFF] |
    [INput]
    [OUTput]
    [REA]
    [ERB]
```

Default: `MISER No`

The Session Manager Network Data Minimizer, otherwise known as MISER, is essentially a data reduction facility which when enabled can considerably reduce data traffic to and from applications. To achieve this, complex compare techniques are employed to ensure that only those portions of the data stream which have changed are updated, as compared to the data currently appearing on the screen.

It should be noted that there is a CPU overhead in employing these techniques, therefore it is recommended that each Installation examine their own requirements, and experiment with various settings to find the optimum solution. The `Q STATS` command displays statistics on the volume of data processed. See the *Installation and Customization* manual for further details.

MISER may be switched on or off, or used selectively by specifying this parameter on appropriate statements. In this way data reduction may be used for those applications or terminals for which it is most effective.

The `INPUT`, `OUTPUT`, `REA` and `ERB` operands would not usually be specified except when advised by a local support representative. These parameters are used if a problem is encountered using MISER and it is necessary to eliminate data reduction for certain areas. These parameters can only be positively specified, they cannot be switched off, although any specification completely overrides any previous specification. If `MISER=Yes` is specified on the `SYSTEM` statement and `MISER=INPUT` is specified for a session defined on an `APPL` statement, for example, then `OUTPUT`, `REA` and `ERB` are effectively turned off for that application. The meanings of these operands are as follows:

`INPUT` restricts data reduction to inbound data streams only.

`OUTPUT` restricts data reduction to outbound data streams only.

`REA` requests the removal of extended attributes.

`ERB` requests emulation of the Read Buffer command.

Note The `REA` operand, although valid, is managed internally, it does not have to be explicitly defined.

When MISER is in force for a session, the queue specified by the `WORKQUE` parameter is not used because MISER keeps an image of the screen and this is updated.

`NETID` *`network-id`*

Specifies the name of the network to be used when establishing a session with the application. *`network-id`* can be up to a maximum of eight bytes long.

If this parameter is not specified, the default is the name of the local network.

See also the section 'The STOPINH and STOP_INHERIT keywords' on page 72.

`ONESCape CONTinue|END|RESET`

Default: `ONESCAPE CONTINUE`

Specifies the action Session Manager takes when the escape command is issued.

`END` causes Session Manager to pass the script named on the appropriate `ENDSCRIPT` parameter to the application. As with the `LOGOFF` parameter, it provides a means of passing input to an application without issuing the `END` command.

`RESET` causes either a VTAM unconditional logoff for the application escaped from or the logical device for the application escaped from to be reset.

`CONTINUE` enables applications to remain active when escape has been issued.

`ONREAD IGNore|SELect|DISPMenu`

Default: `ONREAD IGNORE`

Provides control over the screen display when a system or application, which is not currently displayed, issues a terminal read.

IGNORE causes Session Manager to update the status field for the system issuing the read. This can be recognised by '!!!' if the Menu panel has been set to display the s_aa variable. It is displayed on the next return to the Menu. For DISPMENU, the Menu is displayed immediately. SELECT causes an immediate transfer to the system requiring intervention.

ONWRITE IGNore|SELect|DISPMenu

Default: ONWRITE IGNORE

Provides control over the screen display when a system or application, which is not currently displayed, issues a terminal write. IGNORE causes Session Manager to update the status field for the system issuing the write. This can be recognized by '***' if the Menu panel has been set to display the s_aa variable. It is displayed on the next return to the Menu. For DISPMENU, the Menu is displayed immediately. SELECT causes an immediate transfer to the system requiring intervention.

OUTPUTWarn *nnnn* **[Minutes** *mmm***]**

Default: OUTPUTWARN 100 Minutes 5

Enables identification of application sessions that are in a screen output loop. It specifies the rate of session outputs per session inputs that are to be permitted. If the specified limit is exceeded, message 739 is output.

If message 739 is repeatedly issued for a session, the problem should be investigated and if necessary the session should be stopped to prevent it impacting other users. It is recommended that a global limit be specified on the SYSTEM statement, and then individual limits be set for applications that normally produce a rate higher than the global limit.

nnnn is the number of session outputs per session inputs and *mmm* is the number of minutes. If either value is set to zero (0), then no checking is performed.

PASSTIMEOUT DISPMenu|LOGAPPL

Default: PASSTIMEOUT DISPMENU

Specifies the action that Session Manager should take after the BIND when it has received an unbind request, as a result of the application issuing a 'CLSDST PASS' command, but the expected bind has not been received. DISPMENU causes the Menu screen to be displayed. LOGAPPL requests Session Manager to attempt to re-logon to the application.

PASSTRAnsid [Yes|No|ON|OFF]

Default: PASSTRANSID No

Specifies the action that Session Manager should take when a transid is entered that is assigned to the currently visible session. Yes or ON causes the transid and the aid (Enter, PF or PA key) to be passed onto the session. If the parameter is allowed to default, or if NO or OFF are specified, then the escape sequence is removed from the screen and the keyboard is unlocked.

Note This parameter applies to transids only, it does not apply to other forms of escape sequences.

This parameter is useful where a user's session uses all 24 PF keys and the Session Manager transid for the session is a PF key. Alternatively, the user may wish to enter into the application the transid for that session.

`PCTRANSFER [Yes|No|ON|OFF]`

Default: `PCTRANSFER No`

This parameter should be specified when a file is to be transferred to, or from, a PC which is connected to Session Manager.

When `PCTRANSFER=Yes` is specified, MISER is switched off for this session, datastream compression is not used, addname, addnode and addsid are ignored, the alarm is not sounded, broadcasts are queued, and Session Manager does not intercept any WSF RP(Q) since these are known to interfere with PC emulators. A transfer should not be attempted when Session Manager Windowing is being used at the terminal. Windows must be ended prior to the transfer.

`PSTKAppl` *`passticketapplname`*

Default: None.

Specifies the value to use when generating a passticket for the application (some applications do not use their VTAM applid during authentication of passtickets but use a different value).

The *`passticketapplname`* parameter is a one to eight character field.

See also the section 'The STOPINH and STOP_INHERIT keywords' on page 72.

`PSTKUser` *`alternativeuserid`*

Default: None.

Specifies the value to use in place of the original userid when generating a passticket for the application. If `PSTKUSER` contains a value then the sample passticket exit `ISZEPTKT` will check if the original userid has read access to a resource held in the `PTKTVAL` ESM resource class (exit script samples are in member `ISZSPSTK`). If the user has read access then the passticket will be generated using the userid held in `PSTKUSER`.

The *`alternativeuserid`* parameter is a one to eight character field.

See also the section 'The STOPINH and STOP_INHERIT keywords' on page 72.

`QUITACtive [Yes|No|ON|OFF]`

Default: `QUITACTIVE Yes`

`QUITACTIVE=Yes` allows users to quit or logoff while they have active sessions. Alternatively, `QUITACTIVE=No` prevents users quitting while they have active sessions.

When `QUITACTIVE=Yes` is specified for all sessions, the user may quit from the Menu when any of the systems are active. Specifying `QUITACTIVE=Yes` for an individual session enables a default of `QUITACTIVE=No` to be overridden for that key only. The user cannot quit while any of the systems are active apart from the one after which `QUITACTIVE` is specified.

Similarly, `QUITACTIVE=No` may be specified for an individual session to override an overall option of `QUITACTIVE=Yes`, whether defaulted or explicitly defined. This enables the user to quit at any time, unless the system for which `QUITACTIVE=No` has been defined is active.

`REJBB [Yes|No|ON|OFF]`

Default: `REJBB No`

If the value is set to Yes and a request is received from an application with an SNA Begin Bracket and the session is already in a bracket, then the request will be rejected by sending a negative response containing a 08130000 sense code. The default behavior, or if `REJBB` is set to `No`, for this eventuality, is to accept the request and if required send a positive response with no sense code.

See also the *s_rejbb* session variable as documented in the *Panels, Scripts and Variables* manual.

If updated, changes are applied at the next session start.

`REMOTE` *nodename*

Specifies that the application associated with the session id is running on a remote node, not the local system. *nodename* should be the name as specified on the `LOCALNODE` parameter of the remote system's `SYSTEM` statement.

If the `REMOTE` specification needs to be overridden, then specify *nodename* as being the same as that entered on the local `SYSTEM LOCALNODE` parameter.

See also the section 'The STOPINH and STOP_INHERIT keywords' on page 72.

```
RMISER [Yes|No|ON|OFF] |
    [INput]
    [OUTput]
    [ERB]
```

Default: `RMISER No`

Specifies that MISER data reduction is to be applied to the remote connection for an application session. Data reduction for the inbound and outbound data streams on the path between the terminal and the local Session Manager is controlled by the `MISER` parameter, described on page 59. The `RMISER` parameter controls data reduction for inbound and outbound data streams on the path between the local and remote Session Manager.

See the chapters on the Session Manager networking feature and the MISER feature in the *Installation and Customization* manual for further details and examples.

```
SAUTOSEQ escape [ A | C ismcommand | E | I | J | N scriptname]
   [PASS REMESC] [NOCURESC]
   [ ACTIONKEY aidkey ] [ COMMANDPRFX Y | N | ON | OFF ]
   [ PARM Y | N ]
```

Specifies a sequence of characters, a function or attention key, or the lightpen, for performing a task such as invoking a script in the same way as the `AUTOSEQ` parameter, but for a session.

escape, which must be delimited by a null or space if issued, can be from one to eight characters long, PF1-24, PA1-3, 'ATTN', 'CLEAR' or 'PEN'. The parameters `A`, `I`, `E` and `N` specify the type of script to be run when *escape* is entered. The parameters `C` and `J` enable other actions to be performed when *escape* is entered.

The subparameters have the following meanings:

A – The AUTOSCRIPT is run, for this session.

C – Create a synonym called *escape* for the supplied valid Session Manager command *ismcommand*. This synonym may be issued within sessions. The *ismcommand* can be any Session Manager command the user could enter on the menu, except command scripts. Session selections and session TRANSIDs can be entered, causing the user's visible session to be switched to the requested session. If the Session Manager command has operands, the entire *ismcommand* must be enclosed in single quotes, for example:

```
'STOP USER USER1'
```

If the Session Manager command contains a character string normally enclosed in single quotes, the entire *ismcommand* must be enclosed in double quotes:

```
"BROADCAST 'Good Morning' ALL"
```

The *ismcommand* may contain variables which are substituted before the Session Manager command is invoked:

```
'QUERY USER &T_USER&'
```

where &T_USER& is replaced by the user's identity when the Session Manager command is invoked.

If the *ismcommand* is rejected the user will receive two broadcasts: firstly message 526, and then the resulting error message.

E – The ENDSCRIPT is run, for this session.

I – The scriptname supplied with *escape* is run.

J – Defines a character string alias that represents an instruction to jump to an applid. Session numbers can be entered as well as a session's applid. The alias is specifed in the *escape* value, it must be one to eight characters long and has no default value. For example, an alias may be defined as follows:

```
SAUTOSEQ .J J
```

If an example session had an applid CICSA, the user could jump to the session by entering:

```
.J CICSA
```

If the destination applid is invalid, message 527 is issued in a Session Manager broadcast sent to the user.

N – The script specified on the SAUTOSEQ parameter is run.

PASS – Passes the complete datastream containing the SAUTOSEQ *escape* sequence onto the script. Script logic can then be used to determine if the sequence is to be passed onto the application. For example, it can be useful when used with a PCTRANSFER script driven by a SAUTOSEQ escape setting of IND£FILE. The datastream containing the IND£FILE and its parameters can then be passed to the session.

NOCURESC – If CURESC YES has been specified in the configuration, the NOCURESC parameter allows SAUTOSEQ sequences to be recognized and actioned immediately they are entered in an input field, regardless of the cursor position. This is particularly useful if you are running the supplied script GOPCTRAN, which automatically detects IND£FILE and issues the PCTRANSFER command.

REMESC – Specifies that the escape sequence should be removed from the terminal screen. If REMESC is not used then the escape sequence will remain displayed on the screen when the user returns to the session. Optionally, the variables ucrestesc, unrestescrow and unrestesccol can, if required, be read by the script and used to reinstate the sequence that has been removed. See the *Panels, Script and Variables* manual for further information.

Note If the MISER function (see the *Installation and Customization* manual) is not enabled on sessions where REMESC is being used, returning to the session after an escape may result in incomplete screens being displayed.

Note The REMESC sub-parameter of the PASS parameter only applies to SAUTOSEQ options which cause a script to run, that is options A, E, I and N. REMESC may not be used with the C or J options, as these will automatically remove any entered escape sequence.

ACTIONKEY - Used to specify an *aidkey* that will invoke the synonym. *aidkey* may be any valid aid key PF1-24. If ACTIONKEY is specified then pressing the Enter key will not invoke the SAUTOSEQ *escape* sequence. If an ACTIONKEY is used, the SAUTOSEQ *escape* sequence cannot be a function or attention key. If the user tries to define an ACTIONKEY sequence when the *escape* sequence is a function or attention key, error 4011 is issued.

See also CMDACTionkey on page 32.

COMMANDPRFX - If COMMANDPRFX Y or COMMANDPRFX ON is specified in the statement, a command prefix character is enabled. By default, COMMANDPRFX has a value of N. If a command prefix character is enabled, the actual prefix character to use is specified in the COMMANDPRFXVAL parameter of the USER statement. For example, if a command prefix has been defined in a USER statement:

```
COMMANDPRFXVAL /
```

and the following statement is given:

```
SAUTOSEQ D C DISCONNECT COMMANDPRFX Y ACTIONKEY PF10
```

... the user will be able to disconnect from the session by entering /D and pressing PF10.

See also COMMANDPRFXVAL on page 33.

PARM - If PARM Y is specified in the statement, then Session Manager will scan for a parameter of up to eight characters entered after the sequence. If one is found then the variable ucsautoparm will contain the value. The entered parameter will be treated as part of the sequence and so removed with it. This parameter should only be used on SAUTOSEQ definitions that invoke scripts; the variable would normally be accessed in these scripts. By default, PARM has a value of N.

Any number of SAUTOSEQ parameters can be specified for each session. However, since they are session-specific, the scripts can only be run when the session is active.

Note Escape Sequence Matching

When evaluating an escape sequence, Session Manager will check the sequences configured by the user in the order they were entered. The first sequence that matches is the one that will be actioned.

It is recommended that escapes driven by a character sequence should appear first, sorted in descending order by sequence length. This means that the longest sequences are at the top of the list. Next, escapes invoked by a function or attention key should appear at the end of the list. This order is particularly important when cursor sensitive escapes are enabled (see the `CURESC` parameter).

For example, consider the following escape definitions:

Order	Escape definition	Required outcome
1	`SAUTOSEQ pf5 n aascript`	Run the script aascript when the user presses the PF5 key.
2	`SAUTOSEQ xx n xxscript actionkey pf5`	Run the script xxscript when the user keys in xx and presses the PF5 key.
3	`SAUTOSEQ bs n bscript`	Run the script bscript when the user keys in bs and presses the Enter key.
4	`SAUTOSEQ abs n abscript`	Run the script abscript when the user keys in abs and presses the Enter key.

If the above example definitions are entered in a system where cursor sensitive escapes are active, two problems could occur. If the user types in `xx` and presses the PF5 key, the aascript will run instead of the xxscript. If the user types in `abs` and presses the Enter key, the bscript will run instead of the abscript.

The solution is to change the order of the escape definitions:

Order	Escape definition	Required outcome
1	`SAUTOSEQ abs n abscript`	Run the script abscript when the user keys in abs and presses the Enter key.
2	`SAUTOSEQ bs n bscript`	Run the script bscript when the user keys in bs and presses the Enter key.
3	`SAUTOSEQ xx n xxscript actionkey pf5`	Run the script xxscript when the user keys in xx and presses the PF5 key.
4	`SAUTOSEQ pf5 n aascript`	Run the script aascript when the user presses the PF5 key.

See also the section 'The STOPINH and STOP_INHERIT keywords' on page 72.

`SEQUENCE` *nnnn*

Default: `SEQUENCE 0`.

Specifies the menu sequence number for the session, which is used by the program code for Session Manager 1.1.05 and higher to order sessions.

Notes Sessions that do not have a sequence number appear on Session Manager menus in session/PF Key order *after* sessions with a sequence number.

This parameter will be ignored for users who have sub-menus configured.

To establish the menu sequence number, the usual precedence is used – that is, Session Manager searches these statements in turn:

- User-customized 'priority' sessions set up through the Session List in OLA, identified by the `SESSPRI` parameter of the `USER` statement. For further details, refer to the *Online and Batch Administration* manual.
- `USER` or `TERMINAL`
- `PROFILE`
- `APPL`
  (only when the session definition contains a `REFAPPL` parameter)

Note The Session Manager session variable `s_sequence` may be used to refer to the value of the menu sequence number for an application session; for details, see the product's *Panels, Scripts and Variables* manual.

`SESSDATA1` *`data`* - `SESSDATA5` *`data`*

These five 32-byte user fields (and corresponding read-only variables *`s_sessdata1`* - *`s_sessdata5`*) are for general use by the user. The 5th field will be displayed as asterisks in an Online Administration display. There is no validation on the data within these fields. The variables are available for use in scripts, exits and so on, and are not used internally by Session Manager. Since they are associated with sessions, they may optionally be defined with subscripts. For example, *`s_sessdata1.12`* refers to the value of `SESSDATA1` specified for session 12. The subscript may be a four digit number from 1 to 9999. The variables are not available when running a session or command on a remote system.

See also the section 'The STOPINH and STOP_INHERIT keywords' on page 72.

`SESSPROGMSG [Yes|No|ON|OFF]`

Default: `SESSPROGMSG Yes`

By default, when a session with an application is started, progress messages are issued on panels SESS1 and SESS2 at each terminal. These panels can be suppressed by specifying the No option, so that the first output appearing at a terminal is that from the application. This option is not usually required, except where there is a need to reduce network traffic at session initiation. The system-wide setting may be overridden for individual sessions.

`SESTYPE` *`nnnn`*

Default: `SESTYPE 0`

Specifies the session type, which is used by Session Manager to eliminate duplicate sessions. A user can be associated with multiple profiles (the maximum was increased from nine to eighteen at Session Manager 1.3.05) so the ability to eliminate duplicate sessions is especially useful when assigning more than one profile to a `USER` definition or a `TERMINAL` definition. See also 'Rules for profile selection' on page 25.

Notes (Applicable to existing users of a Classic configuration – that is, all configuration definitions are stored in members of PDS(s) allocated to the DDNAME of CONFIG.) If you run the OLA Enabler to implement the new OLA format configuration, session types can optionally be assigned to sessions automatically (see the *Online and Batch Administration* manual).

This parameter will be ignored for users who have sub-menus configured.

To establish the session type, the usual precedence is used – that is, Session Manager searches these statements in turn:

- USER or TERMINAL
- PROFILE
- APPL
  (only when the session definition contains a REFAPPL parameter)

Examples

If a user's definition contains a session with a SESTYPE of 6, and the user is associated with a profile which also contains a session with a SESTYPE of 6, then it is considered a duplicate.

Suppose a user's definition is associated with two profiles – PROFILE1 and PROFILE2. If PROFILE1 and PROFILE2 both contain a session with a SESTYPE of 25 then it is considered a duplicate.

Notes

a If the session number is zero (also the default if a SESTYPE is not specified) then the session is used; that is, session types of zero are *not* regarded as duplicates.

b The Session Manager session variable s_sestype may be used to refer to the value of the session type for an application session; for details, see the product's *Panels, Scripts and Variables* manual.

c The SESTYPE parameter is important when the user-customizable 'priority' session facility is being used and a user has a common enduser parameter SESSPRIAPPL set to NO (the default) or when the user-customizable 'autostart' session facility is being used and a user has a common enduser parameter SESSAUTOSAPPL set to NO (the default). See page 46 for further details of the SESSPRIAPPL parameter and page 130 for further details of the USER statement SESSPRI parameter. See page 46 for further details of the SESSAUTOSAPPL parameter and page 129 for further details of the USER statement SESSAUTOS parameter. The SESTYPE parameter can be set automatically by the OLA Enabler, or by running the sample job ISZSTJOB.

d If a session is allocated a SESTYPE value, all the user's other session SESTYPE values will be checked for duplicates. If a user has a large number of sessions it may affect performance, especially when the user signs on to Session Manager. Please bear this in mind when implementing SESTYPE, and test on a non-production system to ensure the performance is at an acceptable level. Please contact your local Support Representative if further assistance is needed.

`SIDLTime` *`minutes`* `[DIRECTION IN|INOUT]`

Default: `SIDLTIME 0 DIRECTION IN`

Specifies the interval, in minutes, since the last input was received from the terminal after which the session is to be terminated.

`minutes` may be any value in the range 0-1440 minutes. If 0 is specified (the default) no timeout occurs.

`DIRECTION` controls whether (activity on) input alone, or on both input and output datastreams will inhibit a timeout. If `DIRECTION IN` is in effect then the interval counter is not reset by screen updating.

When the session is timed out, normal session-end processing is performed, including execution of an `ENDSCRIPT` (if one is defined for the session).

Note For HLA, sessions will not independently time out at the Standby Instance, since they will mirror what happens at the Active Instance. If an E36 exit script has been coded, the exit script will run at the Standby Instance, however a return code of zero (honour timeout) will not be actioned. Session detail variable `s_standby` may be inspected in the E36 exit to prevent unnecessary E36 exit script processing.

`SNABUSY [Yes|No|ON|OFF]`

Default: `SNABUSY Yes`

This parameter is useful for non-visible SNA sessions which issue unsolicited writes. When `SNABUSY=Yes`, or allowed to default, Session Manager indicates to the application that it is busy, and the application then ceases to send output until the application screen is next displayed.

For applications which are unable to accept the busy indication, `SNABUSY=No` should be specified. In this instance the `WORKQUE` parameter may be used to store a number of the output screens.

When output screens are suppressed using `SNABUSY`, the status variable shows `s_aa` as set to `-Q-` and `s_a` to `70` on the Menu screen. Normally, `s_aa` would be set to `***`, and `s_a` to `80`.

`STARTSCript` *`scriptname`*

Optionally specifies the name of a script which is passed to any selected application when Session Manager starts (that is, activates) the session. Scripts are generated using the `SCRIPT` statement described in the *Panels, Scripts and Variables* manual. The *`scriptname`* specified on the `STARTSCRIPT` parameter must appear on a `SCRIPT` statement.

For Session Manager 1.1.10 and higher, the `STARTSC` command enables a suitably authorized user to disable or (re-)enable the running of the session start script. For details, see page 243.

See also the section 'The STOPINH and STOP_INHERIT keywords' on page 72.

`STATs [Yes|No|ON|OFF]`

Default: `STATS No`

Specifies whether session statistics are to be collected for this session. When `STATS=Yes` is specified, the address of the session statistics block is available to the User exit E39.

A sample E39 exit (`ISZE39SM`) is supplied with the product. This sample exit may be tailored to produce SMF records containing data to meet your installation requirements.

If session statistics are requested, a small, but accumulative, amount of processing and storage is required. The collection of session statistics can be stopped by specifying `STATS=No`.

`TERMSCript` *`scriptname`*

Optionally specifies the name of a script to be executed when a session has terminated. The script does not communicate with an application, so any parameters such as `INPUT`, `WAITAPPL`, `OUTSCAN...`, contained in it are ignored. This script is generally used to reset session options after termination of a session.

Scripts are generated using the `SCRIPT` statement described in the *Panels, Scripts and Variables* manual. The *`scriptname`* specified on the `TERMSCRIPT` parameter must appear on a `SCRIPT` statement.

See also the section 'The STOPINH and STOP_INHERIT keywords' on page 72.

`UNBIND [WAIT|IMMEDiate]`

Default: `UNBIND IMMEDIATE`

Determines how Session Manager handles an UNBIND request from an application.

`IMMEDIATE` causes each UNBIND to be processed as it is.

`WAIT` causes each UNBIND received from the application to be treated as UNBIND with BIND forthcoming.

`WAIT` may be used for applications such as Logon Managers which transfer between different secondary applications. This is usually achieved by the Logon Manager issuing an UNBIND with BIND forthcoming, and the secondary application issuing an unsolicited BIND.

The problem occurs at log off from the secondary application if it issues an UNBIND without BIND forthcoming. Session Manager assumes that the session has ended, so the unsolicited BIND from the Logon Manager cannot be matched and message `268I` is issued. When `UNBIND WAIT` is specified, Session Manager maintains the session for one minute and waits for a BIND.

When a primary application issues an UNBIND, Session Manager makes no change, even when `WAIT` is specified.

If the final UNBIND which terminates the session is not issued by the primary application, then the name of the application can be specified on the `UNBINDAPPL` parameter so that Session Manager processes the UNBIND immediately. If this is not done, the session would hang for one minute until the UNBIND timed out and the session would then terminate.

`UNBINDAPPL` *`applid`*

(Not applicable to the `SYSTEM` statement.) This parameter is used in conjunction with the `UNBIND=WAIT` specification. `UNBIND=WAIT` causes each UNBIND from secondary level applications to be treated as UNBIND with BIND forthcoming.

In some instances, a secondary application may validly issue an UNBIND to cause a logoff, but if `UNBIND=WAIT` is specified, Session Manager maintains the session for one minute.

`UNBINDAPPL` specifies the name of a secondary application for which UNBIND will not be treated as UNBIND with BIND forthcoming, and so an UNBIND will cause immediate session termination.

For further explanation and examples, refer to the section 'UNBIND WAIT Applications' in the 'Setting up applications' chapter of the *Installation and Customization* manual.

See also the section 'The STOPINH and STOP_INHERIT keywords' on page 72.

`UNDERISZSMGR|CLOSEDISC|CLOSELOGOFF`

Default: `UNDERISZSMGR`

`CLOSEDISC` causes the physical terminal to be disconnected from Session Manager when a target application is selected from the Menu panel. As a result, any other active sessions remain active and may be re-entered exactly where they were left when the user quits from the selected session. This parameter should therefore be used instead of `CLOSELOGOFF` in a situation where a user normally has multiple sessions active.

When `CLOSELOGOFF` is specified, Session Manager relinquishes control of the physical terminal to the target application as soon as the application is selected. The logmode entry name and any data defined using the `DATA` parameter are passed to the target application. If the target application is not currently accepting logons, the request is rejected and an error message issued.

This parameter would normally be used in conjunction with the `LOGAPPL` parameter of the `VTAM LU` statement which defines the terminal to VTAM. In this way, the terminal can be logged back on to Session Manager as soon as the user quits from the application and have the Menu screen displayed again.

For terminals at which a single application only is usually active, this technique may be useful since it cuts out the overhead of Session Manager processing all the data streams passing between the physical terminal and the application. It does however have the disadvantage that no other sessions can be run simultaneously to which the user can transfer control as would normally be the case. No Session Manager messages or broadcasts will be received at the terminal while it is controlled by the target application and no scripts may be run.

Note If `SHAREDISC` or `SHARESESS` are being used to implement the Shared User Facility (see the *Installation and Customization* manual) then `CLOSELOGOFF` will behave as `CLOSEDISC` for the primary user (the first terminal logged on with the userid), but for secondary users `CLOSELOGOFF` will behave as usual.

Further details regarding the implications of using the CLOSEDISC and CLOSELOGOFF parameters can be found in the *Installation and Customization* manual in 'Using Session Manager as a Front-End'.

If UNDERISZSMGR is specified, or allowed to default, the physical terminal remains under the control of Session Manager, and the user can switch between sessions without the need to quit from any session.

The STOPINH and STOP_INHERIT keywords

The majority of Common Session Parameters have a mechanism to prevent inheritance of a value from the SYSTEM, PROFILE, USER, and TERMINAL statements by specifying No or OFF to take the default value instead. The following, however, do not have this mechanism.

```
ACB
ADDSID
APPLSEL
AUTOSCRIPT
BLANKSCRIPT
BRDVAR
ENDSCRIPT
ENVIRONSCRIPT
DATA
DESCRIPTION
INITSCRIPT
LOGMODE
NETID
PSTKAPPL
PSTKUSER
REMOTE
SAUTOSEQ
SESSDATA1 - SESSDATA5
STARTSCRIPT
TERMSCRIPT
UNBINDAPPL
ADDSID
SAUTOSEQ
```

To enable the adoption of default values rather than inheritance values, you can now use two new keywords, STOPINH and STOP_INHERIT.

STOPINH will stop the value of a keyword from being inherited and revert to the default. For example,

```
ACB STOPINH
```

specified on a session definition will stop the inheritance of an ACB value specified on a USER statement and revert to the SESACB value on the SYSTEM statement (the default).

The STOP_INHERIT keyword can be used for the ADDSID and SAUTOSEQ parameters which can have multiple values when you want to suppress some while keeping others. For example, consider the following definitions on a SYSTEM statement:

```
SAUTOSEQ gb1 n cmd1
```

```
SAUTOSEQ gb2 n cmd2
SAUTOSEQ gb3 n cmd3
```

By specifying:

```
SAUTOSEQ gb3 STOP_INHERIT
```

on a `USER, PROFILE` or session definition, the sequence `gb3` will be suppressed while the sequences `gb1` and `gb2` will be retained.

CHAPTER 3

OPTION statement

The `OPTION` statement is used to define various options for the execution of IBM Session Manager for z/OS, such as whether all input control statements are printed, and whether the Session Manager system is to be activated when validation is complete. It may also be used to specify the id of the configuration source file, and the name of the User exit.

Multiple `OPTION` statements may be specified, but if an `OPTION` statement specifies the configuration file suffix or the User exit name, the statement **must** appear before any other statements.

An alternative method of supplying the options to Session Manager is to specify the options as a parameter value using the `PARM` operand of the EXEC job control statement.

When this alternative method is used, the `OPTION` keyword may be omitted. The maximum length of this parameter value is 73 bytes. This method enables the system administrator to easily activate the system with a different system configuration.

Syntax and parameters

The `OPTION` statement takes a range of parameters:

```
OPTION
    CONFIG
    EXIT
    Enn
    GFS
    MDPROF
    MSGUPPER
    PRINT
    SECURITY
    START
    TEST
```

Parameters

`CONFIG` *xx*

Default: `CONFIG 00`

Specifies the name of the source module which contains the configuration control statements. Session Manager forms the name of the source module by concatenating:

- the string 'ISZCON'
- *xx* – the specified (or default) suffix, which uniquely identifies the instance. A value of `BT` is not allowed as this is used by Batch Administration.

The default configuration source module is therefore ISZCON00. To process a source module with a suffix other than '00', it should be specified on this parameter. Any two alphanumeric characters are valid.

The Session Manager global variable t_config_suf may be used to refer to the `CONFIG` *xx* value; for details, see the *Panels, Scripts and Variables* manual.

`EXit` *exitname*

Default: none

Specifies the name of an external module. Session Manager loads this module immediately after the successful validation of the `OPTION` statement, and thereafter calls the module under a number of different circumstances. The chapter 'Session Manager User Exit' in the *Installation and Customization* manual describes the User exit and its invocation points. See also the `Enn` parameters below.

```
E05 S|E
E06 S|E
E08 S|E
E11 S|E
E21 S|E
E22 S|E
E26 S
E29 S|E
E31 S|E
E33 S|E
E36 S
E39 S|E
E79 S|E
E99 S|E
```

Default: `S` for E26 and E36 exit points, `E` for all other exit points.

If an exit name is specified, **all** exit points (except E26 and E36) are invoked. By default, a User exit module is loaded.

Exit points E26 and E36 can only be specified as exit scripts.

To combine some Assembler/COBOL User exit processing with some script processing, an exit name must be specified and the script exit points must be specified with 'S'. For example:

```
EXIT ISZEXT00
E31 S
E39 S
```

Session Manager looks for scripts called `EXIT31` and `EXIT39` for the E31 and E39 exit points. All other exit points are handled solely by `ISZEXT00` (supplied sample `ISZE00DR`). The E31 and E39 script exits may also pass control on to `ISZEXT00` by using the appropriate return code. This is just an example and exit `ISZEXT00` could be replaced by another supplied exit or a customer exit.

If **only** script processing is to be used, then an exit name should **not** be specified. The option 'S' should be specified for each of the required exit points. For example:

`E21 S` to run `EXIT21` script
`E29 S` to run `EXIT29` script

Note If the 'S' option is specified for an exit point, and the corresponding script has not been successfully defined, then Session Manager will fail to initialize.

The exit point parameters only take effect on the initial load of Session Manager, not on update. To update a script or exit process see the 'UPDATE administrator command' on page 270 and 'Using the Update Facility' on page 199.

Exit Points E01, E09 and E71 are only applicable to the User exit. They cannot be specified as scripts.

If a User exit has been loaded, it is always called with E99, the closedown exit point, even if `E99 S` is specified. When this is the case, the script `EXIT99` is also invoked.

```
GFS Yes|No|ON|OFF
    [STORLim nnnnnnnn|nnnnnnnnK|nnnnM]
    [CUSHion nnnnnnnn|nnnnnnnnK|nnnnM]
    [THREShold Yes|No|ON|OFF|LOG]
    [WARNing nnn]
    [SEVEre nnn]
    [CRITical nnn]
    [MAXstor Yes|No|ON|OFF]
```

Default: `GFS Yes`

Specifies that the Get/Free Storage option is to be used. The `GFS` option enables Session Manager to manage its own storage in a more efficient way than the operating system does. It therefore reduces CPU usage. GFS supports storage above the 16M line.

`STORLim` *nnnnnnnn*|*nnnnnnnn*K|*nnnn*M

Default: `STORLIM 10M`

When `GFS=Y` is specified, this parameter may be used to control the amount of storage managed by GFS. It may be specified as a number of bytes, or in Kilobytes, or in Megabytes. The default is 10 Megabytes. It is suggested that the `STORLIM` operand be set to 72M. Use the `Q STOR` command during a period of high system usage, after Session Manager has been installed, to check storage usage and alter the value if required.

Maximum value is 2046M.

`CUSHion` *nnnnnnnn*|*nnnnnnnn*K|*nnnn*M

Default: `CUSHION 1024K`

When `GFS=Y` is specified, Session Manager always reserves a cushion of storage for system use, outside of GFS. This parameter may be used to control the size of the cushion. It may be specified as a number of bytes, or in Kilobytes, or in Megabytes.

`THREShold Yes|No|ON|OFF|LOG`

Default: `THRESHOLD Y`

When `GFS=Y` is specified, thresholds may be defined to determine the action to be taken if an excessive storage usage condition arises.

If `THRES=N` is specified, threshold control is not active. No actions occur.

If `THRES=Y` is specified, threshold control is active, as defined by the parameters `WARNING`, `SEVERE` and `CRITICAL`, outlined below.

If `THRES=LOG` is specified, threshold control is inactive, but the relevant messages are output to the Log when storage usage exceeds the levels defined by `WARNING`, `SEVERE` and `CRITICAL` below.

Note The percentages specified in the following parameters are percentages of the space managed by GFS; that is, the value of `STORLIM` (or its default). The actual storage available in the machine may be larger.

Storage shortage is unlikely to be a problem. However, THRES=LOG may be used to judge whether the storage managed by GFS should be increased for performance reasons.

WARNing *nnn*

Default: WARNING 85 (percent)

The percentage of storage use at which warning message 571I is issued to the Log. No further action is taken at this point.

SEVEre *nnn*

Default: SEVERE 90 (percent)

The percentage of storage use considered severe. If THRES=Y has been specified the following actions are taken:

UPDATE commands entered at this point are rejected until the storage shortage is relieved.

TRACE commands entered at this point are rejected until the storage shortage is relieved.

All active traces are terminated.

Startup of new sessions is prevented.

New attempts to logon to the system are rejected.

BROADCASTs are suspended until the storage shortage is relieved.

The text of message 577I is assigned to the variable t_sos_msg, to inform users of the situation.

CRITical *nnn*

Default: CRITICAL 95 (percent)

The percentage of storage use considered critical. If THRES=Y is specified, in addition to the actions outlined in SEVERE above, any UPDATE commands currently executing are terminated.

MAXstor Yes|No|ON|OFF

Default: MAXSTOR No

In systems where storage-usage is not a constraint, MAXSTOR=Y further reduces the possibility of storage fragmentation; however, it may increase storage used by up to 100 percent. If used, the value set for STORLIM should be doubled. MAXSTOR can only be specified when GFS=Y is specified.

MDPROF [L|M]

Default: MDPROF L

If MDPROF M is specified each DEFPROFILE defined on the SYSTEM statement will be added as a default PROFILE. Up to eighteen DEFPROFILEs may be specified. If MDPROF L is specified and multiple DEFPROFILEs defined then only the last DEFPROFILE specified will be used as a default PROFILE. If MDPROF is not specified then MDPROF L is assumed.

If updated, then the new value comes into effect when system is recycled.

`MSGupper [Yes|No|ON|OFF]`

Default: `MSGUPPER No`

Specifies whether Session Manager messages should be output in upper case, rather than mixed case (the default). In some instances, for example Kanji and Hebrew terminals, messages must be output in upper case only. In this instance, `MSGUPPER=Yes` should be specified.

Note This parameter only effects Session Manager hardcoded messages. Messages defined using the `MESSAGE` statement are not changed.

`Print [Yes|No|ON|OFF]`

Default: `PRINT No`

Specifies the Session Manager control statement print option. When `Yes` is specified, all control statements defined in the configuration source file are output to the log and the Audit file by default. The `MESSAGE` statement (MSG 862I) may be used to cause the control statements to appear on the Audit file only, to avoid filling the console log.

When `No` is specified, or allowed to default, only statements in error are reported.

Note The `PRINT` option specified or defaulted to on the `OPTION` statement defines the default for the print option of the `UPDATE` command.

`Security [Yes|No|ON|OFF]`

Default: `ON`

This parameter is only operative if the ISZE21SF or ISZE21PH security exit is being used; in all other cases it will be ignored.

If the `SECURITY` parameter is `Yes` or `ON` then ISZE21SF or ISZE21PH will call the active ESM for validation, and the ISZE22DM exit will call the ESM for dynamic menu processing.

If the `SECURITY` parameter is `No` or `OFF` then ISZE21SF or ISZE21PH will not call the active ESM for validation but will assume that there are no security restrictions in place. The authorization level and OLA class will not be set. In addition, the ISZE22DM exit for dynamic menu processing will not call the ESM.

`Start [Yes|No|ON|OFF]`

Default: `START Yes`

To prevent Session Manager being activated after validation, specify `START=No` or `START=OFF`. This provides a means of validating a new or amended configuration deck.

`Test [Yes|No|ON|OFF]`

Default: `TEST No`

Activates various diagnostic processes when `Yes` or `ON` are specified. This should only be done at the request of your local support representative when information is required to solve a problem.

CHAPTER 4

INSTALLSU statement

The `INSTALLSU` statement is used to specify that the selectable units (SUs) supplied with the latest version of IBM Session Manager for z/OS be installed. If an `OPTION` statement has been specified in the Session Manager configuration, then `INSTALLSU` should be the next statement specified. If the `OPTION` parameters have been specified as part of the Session Manager startup deck, then `INSTALLSU` should be the first statement in the configuration file.

Syntax and parameters

The `INSTALLSU` statement takes no parameters:

```
INSTALLSU
```

See 'Selectable Units' on page 195 for further details of Selectable Units and their use.

CHAPTER 5

COPY statement

The `COPY` statement enables additional source members to be processed at Session Manager initiation time.

If you are using Online and/or Batch Administration, see also 'PCOPY statement' on page 85.

Syntax and parameters

The `COPY` statement takes a single parameter:

`COPY` *`membername`*

`membername` specifies the name of an additional source member to be processed during product initiation. Any valid combination of alphanumeric characters may be specified for the member name, up to a maximum length of eight bytes.

Session Manager searches for a source member in the partitioned dataset associated with the DDNAME 'CONFIG'.

Note that additional source members must contain complete control statements.

As many `COPY` statements as required can appear in the configuration source members. It may be advantageous to divide the control statements between a number of source members so that if an update is required while Session Manager is active, the `UPDATE MEMBER` command can be issued for the source member affected. The update will complete more rapidly than if the whole configuration file had to be processed, and less storage will be used.

CHAPTER 6

PCOPY statement

This statement is relevant only if you are using Online and/or Batch Administration to tailor IBM Session Manager for z/OS.

If Online and/or Batch Administration is used to tailor the product, configuration data is stored in several PDS(E)s. In this configuration, each PDS(E) is allocated to a particular `DDNAME` and *must be maintained exclusively by Online and/or Batch Administration.*

The `PCOPY` statement loads member(s) of the PDS(E)s allocated to a specified `DDNAME` at Session Manager start up.

Note If you are *not* using Online and/or Batch Administration, see instead 'COPY statement' on page 83.

Syntax and parameters

The PCOPY statement takes these parameters:

PCOPY *ddname* [*member-name*]

ddname

Specifies a DDNAME to be processed. For a list of possible DDNAMEs, see the *Installation and Customization* manual. Unless *member-name* is specified (see below), all members of the PDS(E)s allocated to *ddname* will be loaded during Session Manager start up.

member-name

Supplies the name of the member to be loaded during Session Manager start up and may be any valid combination of alphanumeric characters up to a maximum of eight.

An ISZCONxx configuration member is required for each Session Manager instance that shares the configuration, where *xx* is the unique identifier for the instance. Member ISZCONxx, which is stored in the CONFIG PDS(E)s, should contain a COPY statement for member ISZCOMON, which is processed at Session Manager start up. Member ISZCOMON should contain a number of PCOPY statements, each of which loads all members of the PDS(E)s allocated to a specified DDNAME.

Note OLA recognizes only DDNAMEs that are associated with the Session Manager configuration (see the *Online and Batch Administration* manual). Therefore, do *not* add PCOPY statements to member ISZCOMON (or any other member) for DDNAMEs other than those that are associated with the configuration.

CHAPTER 7

SYSTEM statement

The SYSTEM statement defines various system-wide options, that is, they apply to all terminals and users in the IBM Session Manager for z/OS environment. Many of the values defined for parameters on the SYSTEM statement may be overridden for individual users or terminals by specifying the USER, TERMINAL and/or PROFILE statements appropriately. In addition, session options may be centralized by specifying them on the APPL statement. In this way, the system can be configured to cater for the specific requirements of groups of terminals or users, and for individual sessions. For this reason, it is logical to place the SYSTEM statement at the front of the configuration deck.

The SYSTEM statement is completely optional and Session Manager defaults to the values given in the individual parameter descriptions if the statement is omitted. See also 'General points' on page 26.

In an OLA system there is a SYSTEM*xx* configuration member (stored in the PDS(E) allocated to the DDNAME of SYSTEM) for each Session Manager instance that shares the configuration, where *xx* is the unique identifier for the instance. A value of BT is not allowed as this is used by Batch Administration and a value of CM is not allowed as this is the reserved name for the common SYSTEM statement SYSTEMCM.

The SYSTEMCM member contains SYSTEM parameters common across all Session Manager instances and the SYSTEM*xx* member just contains the SYSTEM parameters unique to that Session Manager instance. By default, that is, as generated by the Enabler or by the sample batch job ISZBACOM, the SYSTEM*xx* member will contain the ACB, LOCALNODE, SESACB and TCP parameters; however the SYSTEM*xx* and SYSTEMCM members may subsequently be modified, using either the Online Administration or Batch facility, to reflect the installation's requirements.

Any updates to the SYSTEMCM member may also affect other Session Manager instances using this shared configuration. You should ensure that you modify and optionally Activate the correct SYSTEM member.

Syntax and parameters

The SYSTEM statement takes a range of parameters:

```
SYSTEM
    ACB
    ALLUSERS
    ATTR
    AUDITOGDG
    BINDTIMEOUT
    CLOSEACBINACT
    CV64
    DEFAPPL
    DEFMENU
    DEFPROFILE
    DLOGLIMIT
    DOMAX
    DUMPGDG
    EXITWALEN
    GENERICACB
    GENRESNAME
    HARDENUSER
    INITIAL_CMD
    INPUTEXIT
    INQINTERVAL
    LOCALNODE
    LOGCMDAUTH
    LOGMNNX
    MDY|DMY|YMD|YDM
    MSGSUFFIX
    MULTUSER
    OLA_DEFER_USERS
    OPEROLACLASS
    OUTPUTEXIT
    PANELID
    PASSTRY
    PREFLANGUAGE
    RCMDTIMEOUT
    RETRCMDS
    RTMTn
    SCREENMODE
    SECURITY
    SESACB
    SHAREAPPL
    SIGNON
    SIGNONPANEL
    SRBUFSIZE
    STANDBY
    SYSDUMP
    SYSPLEXGROUP
```

(contd.)

```
SYSPLEXTYPE
TCP
TN3270E
TN3270E_CONNECT
TRBUFSIZE
TRNUMBER
VERBOSE
WORKQUE
```

Parameters

See also 'Common end-user parameters' on page 30 for Systemwide Defaults for enduser options, and 'Common session parameters' on page 51 for Systemwide Defaults for Session Definitions.

`ACB system_acbname|0`

Default: `ACB ISZSMGR`

Defines the name of the ACB by which Session Manager and VTAM communicate with one another. The name must also be defined on the appropriate VTAM APPL macro.

If `0` is specified, then Session Manager makes no attempt to open an ACB. A VTAM acbname can subsequently be specified on an `ACB` parameter in a different configuration file and this file can then be loaded using the `UPDATE CONFIG` command, Session Manager then attempts to open this new ACB.

If the *`system_acbname`* becomes inactive, Session Manager continues to process and attempts to re-open the ACB every two minutes. In this circumstance, when the ACB is not open, an `UPDATE` command can be issued specifying a configuration file with a new acbname on the `ACB` parameter. Session Manager then attempts to open this new ACB.

If the *`system_acbname`* has been opened by another application, then Session Manager continues to process, but without the terminal acb. This means that users will be unable to continue. A new acbname can be specified using the `UPDATE` command as in the previous two examples.

`ALLUSERS [Yes|No|ON|OFF]`

Default: `ALLUSERS No`

`ALLUSERS=Yes` enables all users to sign on to the Session Manager system, irrespective of whether matching `USER` statements are defined.

External security may be controlled by the Signon Validation exit point (E21) of the User exit documented in the *Installation and Customization* manual.

`ALLUSERS=No` restricts users signing on to those which have a matching userid defined by a `USER` statement. This enables the effect of the `ALLUSERS` parameter to be reversed when using the Update facility.

`ATTR` *`attr-char`* `OFF|`*`field-attribute-parameters`*

Fields specified on panel definitions use special symbols to provide the field attributes. Session Manager itself supplies default attribute definitions for specific characters. These are:

#	UNPROT,NORM,IN
%	UNPROT,HIGH,IN
"	UNPROT,NOND,IN
?	PROT,NORM,OUT
&	PROT,HIGH,OUT
<	PROT,NORM,LIT
@	PROT,HIGH,LIT

Note In this document, the 0x7C (that is, x'7C') character is always presented as the @ sign. It may be displayed as a different character in some non-English code pages. You should enter the appropriate 0x7C character symbol for the code page you are using.

The `ATTR` parameter of the `SYSTEM` statement may be used to override the Session Manager supplied values, giving alternative values to apply globally throughout the panel definitions, although these in turn may be overridden for individual panels. The specified value should be enclosed in quotes. For example:

```
ATTR '?' PROT,NORM,OUT
```

`OFF` may be used to clear the setting for a character, if the character itself is to be displayed within text on a panel.

Acceptable field attribute parameters are as follows:

For protection and content:	ASKIP
	DETect|NODETect
	UNPRotected|PROTected
	NUMeric|ALPHanumeric
	NONdisp
For color and effects:	RED|YELlow|WHIte|GREen|BLUe|PINk |TURquoise
	HIgh|NORMal
	REVerse|BLInking|UNDErscore
Special attributes:	ETMODE|DBCS
	ICursor
	BLANK|NOBLANK
	IN|OUT|LITeral

Synonyms may be used. Valid synonyms are:

`SKIP` for `ASKIP`

`NODisp` for `NONdisp`

`USCOre` for `UNDErscore`

Refer to the *Panels, Scripts and Variables* manual for further details of these attributes.

When `ATTR` is omitted for a field, the default attributes are:

```
PROTected
ALPHanumeric
NODETect
NORMal
SKIP
LITeral
Default color - usually green
```

AUDITOGDG `[Yes|No|ON|OFF]`
 `ATGBASE` *gdgbasename*
 `ATGUNIT` *unitdevicetype*
 `ATGSMSCLASS` *smsdataclas*
 `ATGPSPACE` *numberofcylinders*

Default: `AUDITOGDG No`

If set to `Yes` the parameter specifies that audit log records should be written to the audit GDG dataset.

The `AUDITOGDG` parameter specifies whether the audit log is written to a Generation Data Group dataset (GDG).

During start up the (+1) generation level dataset within the GDG cycle will be allocated, catalogued and opened. If a failure is encountered during this then an error message will be issued and no Audit log records will be written to the GDG. In order to close the current GDG dataset and allocate, catalogue and open the next (+1) generation without re-starting, the `SPIN` command should be issued. This will take the following format:

```
SPIN AUDITGDG
```

The Audit GDG will be allocated as a variable blocked (VB) dataset which has a record length of 260 and a block size of 3384. If a failure is encountered whilst writing out a block then an error message will be issued and no further writes will be attempted until the next `SPIN AUDITGDG` command is issued.

Sub-parameters under the `AUDITOGDG` parameter are:

`ATGBASE` *gdgbasename*

Default: None.

This is a mandatory parameter if `AUDITGDG` is set to `Yes`. This specifies the audit GDG base name which forms the prefix of the audit GDG dataset name to be allocated during Session Manager start up, or on issuing the SPIN AUDITGDG command (that is, *basname*(+1). Must be one to thirty five characters in length.

`ATGUNIT` *`unitdevicetype`*

Default: SYSDA

This specifies the type of device upon which the audit GDG dataset will reside. Must be between one and eight characters in length.

`ATGSMSCLAS` *`smsdataclas`*

Default: None

This specifies the SMS data class if the audit GDG is SMS-managed. This must be between one and eight characters in length.

`ATGPSPACE` *`numberofcylinders`*

Default: 5

This specifies the primary space allocation, in cylinders, to assign to the audit GDG dataset. This must be a numeric value between 1 and 999.

`BINDTIMEOUT` *`seconds`*

Default: `BINDTIMEOUT 30`

The value set with this parameter is used when processing BINDs after UNBINDs with BIND forthcoming marked, or applications with `UNBIND WAIT` specified. *`seconds`* is the time that Session Manager waits, after receiving an UNBIND request, for a matching BIND to appear. *`seconds`* can be any value in the range 0-999 seconds.

`CLOSEACBInact [Yes|No|ON|OFF]`

Default: `CLOSEACBINACT No`

The `CLOSEACBINACT` parameter with the `Yes` or `ON` option causes the closure of a back end session ACB if the ACB has no active sessions. The default behavior is for all ACBs to remain open until Session Manager terminates or the ACB fails, which avoids the need to re-open the ACB when it is next used. Alternatively, specifying `CLOSEACBINACT=Yes` may decrease the time taken to terminate Session Manager as inactive ACBs will not need to be closed.

`CV64 [Yes|No|ON|OFF]`

Default `CV64 No`

When a VTAM application is accessed by a TN3270 client directly via the z/OS Communication Server TN3270 Server then the application receives a CV64 structure in the VTAM session's CINIT. The CV64 structure contains the TN3270 client's IP address and port which can be used by the application to help identify the origin of the TN3270 Client. Prior to this version of Session Manager, if any Session Manager was placed between the TN3270 Server and the VTAM application, then no CV64 structure was supplied to the application in the session's CINIT. Providing z/OS is at Release 12 or higher and the CV64 parameter is specified, Session Manager will supply a CV64 structure to the VTAM application in the session's CINIT.

The CV64 structure is associated with the VTAM ACB used to access the application. This association is constructed when the ACB is opened. The association is lost when the ACB is closed. Therefore you should ensure your configuration provides an ACB that is not used by any other user whilst this association between the TN3270 client and the ACB exists. Consider using the

SYSTEM parameter CLOSEACBINACT to close an ACB when the last user's session using an ACB has terminated. So if a user accesses an application with an ACB that was opened by another user (and not subsequently closed then re-opened), the original user's CV64 will be passed to the application.

If the CV64 parameter is specified then TN3270 users who access Session Manager directly via its TN3270 Server will also provide a CV64 structure to their VTAM applications.

If CV64 is not specified or CV64 No or OFF is specified, no CV64 structure is provided to the VTAM application.

DEFAPPL *applname*

Default: None

Allow users to specify a default APPL statement. This will be used to supply an ACB range to applications that are known to VTAM but do not have a specific APPL statement within the Session Manager configuration.

DEFMENU *panelname*

Default: DEFMENU MENU

Specifies the name of the default Menu panel name. The *panelname* can be any combination of alphanumeric characters up to eight characters long. The name entered must have a matching panel name on a PANEL statement.

DEFPROFILE *profilename*

Default: DEFPROFILE PROFILE

Specifies the name of the profile which Session Manager is to apply to a terminal or a user, if one is not defined on the appropriate TERMINAL or USER statement, or specified via the E21 exit. The *profilename* can be any combination of alphanumeric characters up to eight bytes long.

For Session Manager 3.100 and higher, multiple DEFPROFILE statements, up to a maximum of eighteen, can be configured by specifying MDPROF M on the OPTION statement and configuring multiple DEFPROFILE parameters.

DLOGlimit *dlog-message-limit*

Default: DLOGLIMIT 60

Specifies a value to restrict the number of entries that may be displayed when the DLOG command is used to display Session Manager log information. *dlog-message-limit* may be any value in the range 0-999.

Domax *domax-limit*

Default: DOMAX 100

Limits the number of iterations in any DO loop that may be specified in any Panel and Script Language (TPSL) logic structure. This is to prevent erroneous logic from running forever. *domax-limit* may be any value in the range 1-9999.

```
DUMPGDG [Yes|No|ON|OFF]
   DUMPBASE gdgbasename
   DUMPUNIT unitdevicetype
   DUMPSMSCLS smsdataclass
   DUMPPSPACE numberofcylinders
   DUMPSSPACE numberofcylinders
```

Default: `DUMPGDG No`

If set to Yes or `ON` the parameter specifies that dumps should be written to a GDG dataset via UNIX services.

When an abend occurs, a UNIX call will be made to clone the address space. The cloned child will then allocate the (+1) generation level dataset within the GDG and write a SNAP dump to it. If a failure is encountered during this then an error message will be issued and a dump will be taken to spool using the normal non-GDG method. Note that, because this dump will be taken in a UNIX cloned address space, it will have the UNIX name BPXAS on the spool. The header line of the SNAP dump, however, will contain the name of the parent address space.

When a dump is taken, message 133 will be issued. Because the UNIX call is almost instantaneous, we recommend adding an `INFORM` group to message 133 so that the necessary people are informed that an ABEND has occurred.

The Dump GDG will be allocated as a variable blocked (VB) dataset with a record length of 125 and a block size of 1632. If a failure is encountered whilst writing out the dump then an error message will be issued and a dump will be taken using the normal non-GDG method.

The subparameters which may be supplied are shown below:

`DUMPBASE` *`gdgbasename`*

Default: `None`

This is a mandatory parameter if `DUMPGDG` is set to `Yes`. This specifies the dump GDG base name which forms the prefix of the dump GDG dataset name to be allocated during Session Manager dump taking i.e. DSN=*`gdgbasename`*`(+1)`

`gdgbasename` specified must be 1 to 35 characters in length.

`DUMPUNIT` *`unitdevicetype`*

Default: SYSDA

This specifies the type of device upon which the dump GDG dataset will reside. Must be between 1 and 8 characters in length.

`DUMPSMSCLS` *`smsdataclass`*

Default: None

This specifies the SMS data class if the dump GDG is SMS-managed. This must be between 1 and 8 characters in length.

`DUMPPSPACE` *`numberofcylinders`*

Default: 50

This specifies the primary space allocation, in cylinders, to assign to the dump GDG dataset. This must be a numeric value between 1 and 999.

`DUMPSSPACE` *`numberofcylinders`*

Default: 5

This specifies the secondary space allocation, in cylinders, to assign to the dump GDG dataset. This must be a numeric value between 1 and 999.

`EXITWALen` *`nnnnn`*

Default: `EXITWALEN 0`

The size of the work area made available to the Input and Output 3270 datastream exits (see '`INPUTEXit`' on page 96 and '`OUTPUTEXit`' on page 101). Specify a value in the range 0-16384.

It is essential that the exit point code checks that it has been passed a work area of sufficient size for its needs. Take care not to overlay storage immediately following the exit work area, since this will result in unpredictable results.

`GENERICACB` *`generic-acbname`*

Default: none

A common (or generic) ACB name can be specified in a Parallel Sysplex® system where a Session Manager system is run on each clone. The *`generic-acbname`* does not need to be defined to VTAM.

Session Manager opens the *`acbname`* specified by the `ACB` parameter and also the *`generic-acbname`*.

Users should sign on using the *`generic-acbname`* and the sysplex will attempt to balance the signons across the clones. When necessary, users can sign on to a particular Session Manager system by using the `ACB` parameter *`acbname`*.

`GENRESName` *`ismgenericresourcename`*

Default: none

Contains the generic resource name (a one to eight character field) to be used, instead of the Session Manager instance name, when verifying the user to the External Security Manager (ESM). This is used when Session Manager is set up as a generic resource and a single signon is being used for the Express Logon Facility (ELF).

An associated variable, `T_GENRESNAME`, is checked by the E21 Exit when issuing a `VERIFY` to the ESM.

`HARDENUser Yes | No | ON | OFF`

Default: `No`

This parameter changes the behavior of OLA. Normally, a user who signs on to OLA with a generic userid and selects 'My User' will see all fields as read-only. If the `HARDENUSER` parameter is set to `Yes` or `On`, a new non-generic user is created when a generic userid is used for signon. The non-generic user is a clone of the original generic user, and inherits the same user settings. However, the new user can modify their fields (subject to security permissions).

The variable T_HARDENU is set to the read-only value 'Y' or 'N' to indicate the value of the HARDENUSER parameter.

INITIAL_CMD *initial_cmd*

Default: none

Specifies a script (which may be a command script specified with parameters) to run at product startup. This function enables Session Manager to run in 'batch' mode with the sole purpose of executing the initial command.

If *initial_cmd* returns a non-zero return code in t_rc then startup of Session Manager will be aborted. An alternative to setting a non-zero return code is to use the START N parameter of the OPTION statement. For a list of return codes, please refer to the section titled 'Starting the system – initiating Session Manager' in the 'Performing a basic installation' chapter of the *Installation and Customization* manual.

INPUTEXit *exitname*

Specifies the standard 1-8 character name of the Input 3270 datastream exit load module. (See the *Installation and Customization* manual for details on Input and Output 3270 datastream exits.) If you do not specify a value for this parameter then the Input 3270 datastream exit will not be invoked.

The size of the work area made available to the Input and Output 3270 datastream exits is controlled by parameter EXITWALEN (see page 95).

INQINTerval *minutes*

Default: INQINTERVAL 1

Controls the interval at which Session Manager checks on the status of applications. See also the INQUIRE parameter on the APPL statement.

A value of 0 indicates that no 'status-enquiries' are to be performed; any other value indicates the number of minutes between status-enquiries. *minutes* may be any value in the range 0-1440 minutes.

The value of INQINTERVAL may be changed at any time by an update. If a value of 0 is changed to a number of minutes, status enquiries are started immediately and are repeated at the specified interval thereafter.

Session Manager only performs the status enquiry if terminal input has been received and an application status field is being displayed.

See also 'INQUIRE administrator command' on page 257.

LOCALnode *nodename*

Identifies the name of the node for this system, when the Session Manager Networking feature is used to link several Session Manager systems. Each system is termed a node in the network and each node name specified must be unique. This parameter may only contain the characters @ $ A-Z 0-9.

Note In this document, the 0x7C (that is, x'7C') character is always presented as the @ sign. It may be displayed as a different character in some non-English code pages. You should enter the appropriate 0x7C character symbol for the code page you are using.

For Session Manager networking to function, the local node must be defined, together with one or more links to other systems in the network. (See the LINK statement definition on page 147.) The NETMAN names of other nodes in the network and routes to these remote nodes are determined dynamically by Session Manager.

When a remote session is specified, the remote *nodename* is used to identify on which system the remote session should be started.

If Eclipse is being used, then a LOCALNODE parameter must be specified.

LOGCmdauth *command-auth-level*

Default: LOGCMDAUTH 7

Each Session Manager command has a numeric security code, which may be allocated by the COMMAND statement, or allowed to default. The LOGCMDAUTH parameter may be used to specify a security level such that any command issued of that security level, or above, will be logged on the Console and possibly the Audit file.

command-auth-level may be any value in the range 1-9.

LOGMnnx *logmode-entryname*

When presenting any application with a logon request, the characteristics of the terminal are acquired from an entry in the VTAM logmode table. Normally, a standard name is generated according to the bind image which Session Manager receives when a user first logs on.

If a logmode entry name other than the standard name is required however, it may be specified using this parameter. The table below gives the valid settings for the LOGM*nnx* parameter, the entry-names that Session Manager selects by default for each one, and the type of terminal to which that entry relates:

Parameter	Entry-name	Terminal type
LOGM71N	S3270	Non SNA 3277-1
LOGM72N	S3270	Non SNA 3277-2
LOGM81N	D4B32781	Non SNA 3278-1
LOGM82N	D4B32782	Non SNA 3278-2
LOGM83N	D4B32783	Non SNA 3278-3
LOGM84N	D4B32784	Non SNA 3278-4
LOGM85N	D4B32785	Non SNA 3278-5
LOGM8XN	D4B3278X	Non SNA 3278-X
LOGM81S	D4A32781	SNA 3278-1
LOGM82S	D4A32782	SNA 3278-2
LOGM83S	D4A32783	SNA 3278-3
LOGM84S	D4A32784	SNA 3278-4
LOGM85S	D4A32785	SNA 3278-5
LOGM8XS	D4A3278X	SNA 3278-X

Additionally, Session Manager is supplied with a logmode table which contains entries for 3278, 3279 and 3193 terminals which have extended attributes as follows:

Parameter	Entry-name	Terminal type
LOGM92N	Non SNA 327x-2	D4B32792
LOGM93N	Non SNA 327x-3	D4B32793
LOGM94N	Non SNA 327x-4	D4B32794
LOGM95N	Non SNA 327x-5	D4B32795
LOGM9XN	Non SNA 3279-X	D4B3279X
LOGM9DN	Non SNA 3279-D	ISZDYNAM queriable
LOGM92S	SNA 327x-2	D4A32792
LOGM93S	SNA 327x-3	D4A32793
LOGM94S	SNA 327x-4	D4A32794
LOGM95S	SNA 327x-5	D4A32795
LOGM9XS	SNA 3279-X	D4A3279X
LOGM9DS	SNA 3279-D	ISZDYNAS queriable

`MDY|DMY|YMD|YDM`

Default: `MDY`

Specifies the format of date fields which appear in panels built by Session Manager and in messages directed to SYSLST. Date fields are displayed either as *mm/dd/yyyy*, *dd/mm/yyyy*, *yyyy/mm/dd* or *yyyy/dd/mm* according to the specification of this parameter.

```
MSGsuffix I|E|W|*
    AUDIT [Yes|No|ON|OFF]
    CONTent [Yes|No|ON|OFF|CODE|TEXT]
    INFORM users-inform-list
    LOG [Yes|No|ON|OFF]
    DESCriptor descriptor-code-list
    ROUTE route-code-list
```

Controls the destination and content of console messages with a common suffix.

The `I` suffix denotes informatory messages, `E` denotes serious error messages, `W` denotes warning messages, and `*` denotes all messages.

Default: `*`

More than one `MSGSUFFIX` parameter may be specified if it should be necessary to define different options for `I`, `E` and `W` messages. The subparameters which may be supplied are shown below:

`AUDIT [Yes|No|ON|OFF]`

Default: `AUDIT Yes`

The `AUDIT` subparameter with the `Yes` or `ON` option causes all messages with the specified suffix to be directed to the Session Manager Audit file. `No` or `OFF` stops messages in the group appearing in the audit.

`CONTent [Yes|No|ON|OFF|CODE|TEXT]`

Default: `CONTENT Yes`

The `CONTENT` subparameter controls the content of the messages with the specified suffix.

`Yes` and `ON` options cause both the message number and the text of the message to be printed.

`No` and `OFF` suppresses both the message number and the text.

`CODE` causes only the message number to be printed.

`TEXT` causes only the text of the message to appear.

`INFORM` *`users-inform-list`*

Specifies a list of userids to which messages in the group are directed. The messages are sent in full, even though the original message appearing on the console may have been modified in accordance with `MSGSUFFIX` subparameters. The userid is that which is entered at Session Manager signon time.

If the first entry in the list matches a Session Manager statement or parameter name, the entry should be enclosed in inverted commas, or parentheses. Otherwise the entry will be interpreted as a statement.

`LOG [Yes|No|ON|OFF]`

Default: `LOG Yes`

By default, messages are directed to the system console. This is the same as specifying the `ON` or `Yes` options. Specifying `OFF` or `No`, prevents messages appearing on the system console.

`DESCriptor` *`descriptor-code-list`*

Each descriptor code has a value from 1 to 16 and corresponds to one of the z/OS action descriptor codes. Some of the codes are mutually exclusive, but Session Manager does not perform any checking. The IBM manual describing z/OS supervisor services and macro instructions gives a list of codes and their associated actions. Each code in the list may be separated by a comma, a blank, or an equal sign.

`ROUTE` *`route-code-list`*

Each route code may have a value between 1 and 16 and equates to an z/OS message routing code. Session Manager does not check the mutual exclusivity of the codes. Reference should be made to the appropriate IBM manual describing z/OS supervisor services and macro instructions for a list of routing codes.

```
MULTUser TERML4
         COUNT RECONAnyterm [Yes|No|ON|OFF] PORTNumber
         [Yes|No|ON|OFF]
```

The Shared User facility enables multiple users to use the same userid (for details, see the *Installation and Customization* manual). The `MULTUSER` parameter specifies how the 'user qualifier' should be created, which is used to:

- Distinguish multiple users signed on with the same userid.

- Enable reconnection of users of the Shared User facility.

Default: `MULTUSER TERML4`

`TERML4` indicates that the last four characters of the terminal LU name should be used to create the 'user qualifier'.

`COUNT` indicates that a unique number in the range 1-99999999 should be used to create the 'user qualifier'.

`RECONAnyterm [Yes|No|ON|OFF]`

Default: `RECONANYTERM No`

If `COUNT` is specified (which is not the default) then the `RECONANYTERM` subparameter with the `Yes` or `ON` option enables any '`SHARE=Y`' user to reconnect to a disconnected user's sessions with the same name from a different terminal. The default behavior is to only allow '`SHARE=Y`' users to reconnect to disconnected sessions that were previously connected to the same terminal name as the new user.

Typically, this parameter would be used when all users with the same name were in fact initiated by one person – for example, a user uses more than one TN3270 client simultaneously.

This parameter would not normally be used when more than one person uses the same userid as it can allow reconnection to another user's disconnected sessions.

Note The `RECONANYTERM` parameter cannot be specified on the `TERML4` sub-parameter, only on the `COUNT` sub-parameter.

`PORTNumber [Yes|No|ON|OFF]`

Default: `PORTNUMBER No`

If `COUNT` is specified (which is not the default) then the `PORTNUMBER` subparameter with the `Yes` or `ON` option specifies that the Telnet client port number should be used during sign-on to uniquely identify the user when the Session Manager TELNET server is in use (that is, when `TCP=Yes` has been configured).

The `PORTNUMBER` option is only available to users connected via the Session Manager TELNET server and not to users connected via the IBM TELNET server.

The `PORTNUMBER` option will only allow users to reconnect to their previously disconnected Session Manager sessions if the user's Telnet client port number is maintained after a disconnect. The user's client software may have this capability, or the user can issue the `DISC LOG` command when disconnecting from the system.

Note The `PORTNUMBER` parameter cannot be specified on the `TERML4` subparameter, only on the `COUNT` sub-parameter.

Do *not* change the value of the `MULTUSER` parameter once a shared user has signed on. If you were to change the value of this parameter then:

- You would not be able to terminate a session associated with a shared user by means of the '`STOP USER` *`userid`*' command.

- You would not be able to distinguish multiple users with the same user name by means of the 'QUERY USER' command, because the 'user qualifier' would not be displayed correctly.
- A shared user would not be able to reconnect after a network failure or after a DISCONNECT command had been issued.

OLA_DEFER_USERS [Yes|No|ON|OFF]

Default: No

A setting of Yes specifies that no OLA USER statements will be loaded at Session Manager Start time and will only be loaded when each user signs on.

When updated any change comes into effect when the system is recycled.

OPEROLACLass *olaclassname*

Default: None

Specifies the OLA class to be attributed to the console operator. It is recommended that this is set to the OLA class which defines the category of 'Implementor'. This is to allow the console operator full OLA authority, but it can be set to any OLA class if there is a desire to limit the OLA authority of the console operator.

OUTPUTEXit *exitname*

Specifies the standard 1-8 character name of the Output 3270 datastream exit load module. (See the *Installation and Customization* manual for details on Input and Output 3270 datastream exits.) If you do not specify a value for this parameter then the Output 3270 datastream exit will not be invoked.

The size of the work area made available to the Input and Output 3270 datastream exits is controlled by parameter EXITWALEN (see page 95).

PANELID Yes | No | ON | OFF

Default: OFF

This parameter controls whether a panel will display the panel ID in the upper left-hand corner of the panel, instead of the product name.

If PANELID is Yes or ON, the panel ID will be displayed.

If PANELID is No or OFF, the product name will be displayed.

Note If the PANELID command is used, it will override any SYSTEM PANELID setting.

PASSTry *passtry-limit*

Default: PASSTRY 0

Specifies a limit to the number of times any user can attempt to sign on to Session Manager. *passtry-limit* may be any value in the range 0-999. When the limit is exceeded, the terminal task for the user is suspended, the terminal is locked, and message 5Ø3I is sent to the console.

For the terminal to be used again, a PASSFREE command must be issued.

If PASSTRY 0 is specified, there is no limit to the number of times the user can attempt to sign on.

`PREFLANGuage` *`languageid`*

Specifies the default language in which screen text for Session Manager messages is displayed at users' terminals, and in which messages are issued to the console.

Two characters, consisting only of letters of the alphabet or numerals, must be specified for the language id.

A language id can be allocated to appropriate Session Manager messages using the `LANGUAGE` parameter of the `MESSAGE` statement. For details, see 'MESSAGE statement' on page 163.

`RCMDTimeout` *`timeout_value`*

Default: 1 minute

`timeout_value` is the number of minutes, in the range 0-1440, after which a command sent to a remote system will time out.

When effective: immediately.

`RETRCMDS` *`nn`*

Default: `RETRCMDS 1`

Specifies the number of Session Manager commands that can be stored for retrieval. *`nn`* can be a value in the range 1-30. The default is set at 1 for backward compatibility.

`RTMT1` *`nnnnn`*

Default: `RTMT1 1000`

Specifies the first threshold for the Response Time Monitor routine. The parameter is used in conjunction with the `RTMT2` parameter to provide threshold values used when calculating the three counts for fast, medium and slow responses output by the Response Time Monitor.

`nnnnn` can be in the range 0-32767 milli-seconds. The value specified for `RTMT1` should be less than the value specified for `RTMT2`.

There is no internal validation for the values entered, so you should ensure the values entered are valid. If a value greater than that in `RTMT2` is specified, the session statistics produced by the Response Time Monitor will not be as expected.

`RTMT2` *`nnnnn`*

Default: `RTMT2 5000`

Specifies the second threshold for the Response Time Monitor routine. The parameter is used in conjunction with the `RTMT1` parameter to provide threshold values used when calculating the three counts for fast, medium and slow responses output by the Response Time Monitor.

`nnnnn` can be in the range 0-32767 milli-seconds. The value specified for `RTMT2` should be greater than the value specified for `RTMT1`.

There is no internal validation for the values entered, so you should ensure the value entered are valid. If a value less than that in `RTMT1` is specified, the session statistics produced by the Response Time Monitor will not be as expected.

`SCREENMODE Normal|Alternate`

Default: `SCREENMODE Normal`

On any Session Manager panel, by default, 24 lines and 80 columns are displayed. If there is a requirement to display as many lines and columns as possible, then the `ALTERNATE` operand may be specified.

```
SECURITY
    [AUTHCLASsname authclassname]
    [AUTHRESname authresname]
    [DYNMClass dynamicmenuclassname]
    [DYNMDROPSESSION Yes | No | ON | OFF]
    [DYNMResnm dynamicmenuresourcename]
    [DYNMALog Yes | No | ON | OFF]
    [DYNMAUtsthid Yes | No | ON | OFF]
    [DYNMHide Yes | No | ON | OFF]
    [DYNMLogmax nnnn]
    [DYNMTYPE appl | vtamappl]
    [ESMPRFCLNM esmprofileclass]
    [ESMPRFRSNM esmprofileresname]
    [ESMPRFACC Yes | No | ON | OFF]
    [OLARESname olaresname]
    [PASSPHrase Yes|No|ON|OFF]
    [SIGNONClass signonclass]
    [SIGNONAccess Yes | No | ON | OFF]
    [SIGNONResname signonresname]
    [TERMINALClass terminalclass]
    [TERMINALResname terminalresname]
    [TERMINALAccess Yes | No | ON | OFF]
```

`AUTHCLASsname`

Default: none

This parameter sets the name of the ESM class in which resources will be queried to determine security levels. The classname can be up to 8 alphanumeric characters.

A global read-only variable `T_AUTHCLASS` is created and is available for use by the E21 exit. The variable will contain an ESM class name up to eight characters long.

If specified, then parameters `AUTHRESNAME` and `OLARESNAME` (OLA systems only) are mandatory.

If not specified, then parameters `AUTHRESNAME` and `OLARESNAME` (OLA systems only) must not be specified

`AUTHRESname`

Default: none

This parameter, along with `AUTHCLASSNAME`, allows a user's authorization class to be set in an ESM.

The resource name can be made up of any alphanumeric characters, including periods ('.'), and can be up to the length permitted by the defined ESM class, minus one character.

The supplied security exit will determine the user's authorization by checking against this parameter within the defined ESM class, AUTHCLASSNAME. Starting at 9 and working down to 1 , the exit will add the authorization level to the AUTHRESNAME specified and will check if the user has read access to the resource within the specified AUTHCLASSNAME.

Validation will stop as soon as read access is granted and the user's authorization level will be set to this value. If no access is granted then the user's authorization level will be set to 1. A user should only have read access to one authorization resource within the ESM.

This process will only occur if an AUTHRESNAME is defined. If a definition exists then any authorization settings in the configuration will be ignored.

For example, if *authresname* is set to ISM.AUTH and *authclassname* is set to ISMCLASS, the supplied security exits ISZE21SF and ISZE21PH will check whether the user has read access to ISM.AUTH9 in ISMCLASS, then whether the user has access to ISM.AUTH8 and so on, until a resource to which the user has read access is found. The user's authorization level will then be set to the appropriate value.

A global read-only variable T_AUTHRESN is created and is available for use by the E21 exit. The variable will contain the resource name, which will be appended with the AUTH value. The resource name can be made up of any alphanumeric characters, including periods ('.'), and can be up to the length permitted by the defined ESM class, minus one character.

If specified, then parameters AUTHCLASSNAME and OLARESNAME (OLA systems only) are mandatory.

If not specified, then parameters AUTHCLASSNAME and OLARESNAME (OLA systems only) must not be specified.

DYNMClass

Default: FACILITY

This parameter, effective when dynamic menus are being used, sets the name of the ESM class in which resources will be queried to determine which applications the user has authority to access. The classname can be up to 8 alphanumeric characters.

A global read-only variable T_DYNMCLASS is created and is available for use by the ISZE22DM exit. The variable will contain an ESM class name up to eight characters long.

See the *Installation and Customization* manual for more comprehensive details on defining security and implementing dynamic menus.

DYNMDROPSESSION [Yes | No | ON | OFF]

Default: OFF

This parameter is used by the supplied ISZE22DM exit to determine whether sessions to which the user has no access are dropped or hidden. If DYNMDROPSESSION is set to Yes or ON then sessions the user can access are not dropped by setting DROP_SESSION No and sessions the user can not access are dropped by setting DROP_SESSION Yes causing the internal storage for these sessions to be deleted.

Whilst this has a benefit in storage usage and performance there are some drawbacks; you will not be able to autoselect or autostart these sessions (also see the DYNMAUTSTHID parameter).

If DYNMDROPSESSION is set to No or OFF then sessions to which the user has no access will have HIDE Yes set, and you will be able to autoselect or autostart these sessions (also see the DYNMAUTSTHID parameter) and sessions to which the user has access will have HIDE No set. Also see the DYNMHIDE parameter.

A global read-only variable T_DYNMDROPSESS is created and is available for use by the ISZE22DM exit. The variable will contain the drop attribute value 'Y' or 'N'.

See also the DROP_SESSION common session parameter on page 55.

See the *Installation and Customization* manual for more comprehensive details on defining security and implementing dynamic menus.

DYNMResnm

Default: none

This parameter, along with DYNMCLASS, sets the name of the ESM resource which, when appended with either the APPL name or APPLID name (see DYNMTYPE on page 107), will be queried to determine which applications the user has authority to access when dynamic menus are being used.

The resource name can be made up of any alphanumeric characters, including periods ('.'), and can be up to the length permitted by the defined ESM class, minus one character.

A global read-only variable T_DYNMRESNM is created and is available for use by the ISZE22DM exit. The variable will contain the resource name, which will be appended with either the APPL name or APPLID name. The resource name can be made up of any alphanumeric characters, including periods ('.'), and can be up to the length permitted by the defined ESM class, minus the length of the APPL name or APPLID name.

See the *Installation and Customization* manual for more comprehensive details on defining security and implementing dynamic menus.

DYNMAUtsthid Yes | No | ON | OFF

Default: OFF

This parameter determines whether sessions hidden from a user may be autostarted or autoselected. A setting of Yes or ON means that hidden sessions can be autostarted or autoselected. A setting of No or OFF means that hidden sessions cannot be autostarted or autoselected.

A global read-only variable T_DYNMAUTSTHID is created and is available for use by the ISZE22DM exit. The variable will contain the value 'Y' or 'N'.

See the *Installation and Customization* manual for more comprehensive details on defining security and implementing dynamic menus.

`DYNMALog Yes | No | ON | OFF`

Default: `OFF`

This parameter determines if message 4028 will be recorded in the audit log whenever the ESM cannot determine whether a user should have access to a session. A setting of `Yes` or `ON` will cause the message to be recorded. A setting of `No` or `OFF` will prevent a message from being recorded.

A global read-only variable `T_DYNMALOG` is created and is available for use by the ISZE22DM exit. The variable will contain the log attribute value '`Y`' or '`N`'.

See the *Installation and Customization* manual for more comprehensive details on defining security and implementing dynamic menus.

`DYNMHide Yes | No | ON | OFF`

Default: `OFF`

This parameter is used by the supplied ISZE22DM exit to determine whether sessions are hidden or dropped (also see the `DYNMDROPSESSION` parameter), if the ESM can not determine if a user would have access to the sessions.

If RACF `PROTECT ALL` is active any undefined resources will return a SAF return code of `8 - access denied`.

When using the standard supplied ISZE22DM exit the session will be hidden or dropped. If RACF `PROTECT ALL` is not active any undefined resources will return a SAF return code of `4,4 - RACF can not make a decision`.

When using the standard supplied ISZE22DM exit a setting of `Yes` or `ON` will cause these sessions to be hidden or dropped. A setting of `No` or `OFF` will cause the sessions to be visible.

The ISZE22DM exit settings will override any configuration common session `HIDE` or `DROP_SESSION` settings.

A global read-only variable `T_DYNMHIDE` is created and is available for use by the ISZE22DM exit. The variable will contain the hide attribute value '`Y`' or '`N`'.

See the *Installation and Customization* manual for more comprehensive details on defining security and implementing dynamic menus.

`DYNMLogmax` *nnnn*

Default: `0`

This parameter sets a limit on the number of `DYNMALog` 4028 messages written to the audit log during each user signon. Any value for *nnnn* from 0 to 9999 may be specified.

A global read-only variable `T_DYNMLOGMAX` is created and is available for use by the ISZE22DM exit. The variable will contain the maximum permitted number of log entries, from `0` to 9999.

See the *Installation and Customization* manual for more comprehensive details on defining security and implementing dynamic menus.

`DYNMTYPE appl | vtamappl`

Default: `appl`

This parameter determines whether the VTAM applid or appl should be used when checking a user's access authorization with the ESM. A `DYNMTYPE` of `appl` will cause the appl name to be used. A `DYNMTYPE` of `vtamappl` will cause the VTAM applid to be used.

A global read-only variable `T_DYNMTYPE` is created and is available for use by the ISZE22DM exit. The variable will contain the resource type value `APPL` or `VTAMAPPL`.

See the *Installation and Customization* manual for more comprehensive details on defining security and implementing dynamic menus.

`ESMPRFCLNM esm`*`profileclass`*

Default: None

This parameter indicates whether profiles should be assigned to a user based on resource access rules held in the ESM. If set (by containing the ESM class name under which profile resources are defined) then this indicates that the profile name(s) will be assigned to a user by interrogating the ESM.

An associated global read-only variable `T_ESMPRFCLNM` is created. The variable will contain an ESM class name up to eight characters long.

A user can be associated with multiple profiles with the maximum being 18. Their sequence is important. If a particular attribute value (common enduser or common session) has not been defined by the user, the profiles will be searched in ascending input order and the first value defined explicitly for that attribute will be used. The order in which a user's access to profiles is checked is controlled via the `ESMLEVEL` parameter on the PROFILE statement (see page 122).

Note If using the ESM to assign PROFILE(s) then the `PROF` parameter of USER statement is ignored. If the ESM prevents access to all profiles for a user then the default profile, as specified under the SYSTEM `DEFPROFILE` parameter, is used.

See the *Installation and Customization* manual for more comprehensive details on defining security and implementing dynamic menus.

`ESMPRFRSNM esm`*`profileresname`*

Default: None

This parameter sets the name of the External Security Manager resource, which will be queried to determine if the user has authority to use this `PROFILE`. The resource name can be made up of any alphanumeric characters, including periods ('.'), and can be up to the length permitted by the defined ESM class, minus the length of the Session Manager `PROFILE` name. An associated global read-only variable `T_ESMPRFRSNM` is created. The variable will contain the resource name, which will prefix the Session Manager `PROFILE` name on issuing the check with the ESM.

Note If using the ESM to assign PROFILE(s) then the PROF parameter of USER statement is ignored. If the ESM prevents access to all profiles for a user then the default profile, as specified under the SYSTEM DEFPROFILE parameter, is used.

See the *Installation and Customization* manual for more comprehensive details on defining security and implementing dynamic menus.

ESMPRFACC Yes | No | ON | OFF

Default: No

This parameter determines whether the user is granted access which allows them to use this Session Manager PROFILE, if the External Security Manager cannot determine if a user would have access to the resource.

If RACF PROTECT ALL is active any undefined resources will return a SAF return code of 8 - access denied and the user is not granted access to the PROFILE.

If RACF PROTECT ALL is not active any undefined resources will return a SAF return code of 4,4 - RACF can not make a decision. An ESMPRFACC setting of Yes or ON will allow the user to use the PROFILE. A setting of No or OFF will not allow the user to use the PROFILE.

An associated global read-only variable T_ESMPRFACC is created.

See the *Installation and Customization* manual for more comprehensive details on defining security and implementing dynamic menus.

OLARESname

Default: none

This parameter, along with AUTHCLASSNAME, allows a user's OLA security class to be set in an ESM.

The resource name can be made up of any alphanumeric characters, including periods ('.'), and can be up to the length permitted by the defined ESM class, minus two characters.

The supplied security exit will determine the user's OLA security class by checking against this parameter within the defined ESM class, AUTHCLASSNAME. Starting at AD and working alphabetically through the OLA class definitions to US, the exit will add the OLA class name to the resource name specified and will check if the user has read access to the resource within the specified AUTHCLASSNAME.

Validation will stop as soon as read access is granted and the user's OLA security class will be set to this value. If no access is granted then the user's security class level will be set to NO. A user should only have read access to one OLA security class resource within the ESM.

This process will only occur if an OLARESNAME is defined. If a definition exists then any OLA security class settings in the configuration will be ignored.

For example, if *olaresname* is set to ISM.OLA. and *authclassname* is set to ISMCLASS, the supplied security exits ISZE21SF and ISZE21PH will check whether the user has read access to ISM.OLA.AD in ISMCLASS, then whether the user has access to ISM.OLA.BT and so on, until a resource to which the user has read access is found. The user's OLA security class will then be set to the appropriate value.

A global read-only variable T_OLARESN is created and is available for use by the ISZE21SF and ISZE21PH exits. The variable will contain the resource name, which will be appended with the OLACLASS value. The resource name can be made up of any alphanumeric characters, including periods ('.'), and can be up to the length permitted by the defined ESM class, minus two characters.

See the *Installation and Customization* manual for more comprehensive details on defining security and implementing dynamic menus.

If specified, then parameters AUTHRESNAME and AUTHCLASSNAME are mandatory.

If not specified, then parameters AUTHRESNAME and AUTHCLASSNAME must not be specified.

PASSPHrase Yes|No|ON|OFF
PASSWORDREQ Yes|No|ON|OFF

Default: No

If set to yes or ON then specifies that a password phrase can be used, in place of a password, if authenticating the user with the External Security Manager during Session Manager signon. If the password data entered is less than nine characters then it will be treated as a password, otherwise it will be treated as a passphrase. A new global variable called t_passphrase will represent this value. Any update to this parameter will take effect at the next Session Manager signon.

The sub-parameter under the PASSPHRASE parameter is:

PASSWORDREQ Yes|No|ON|OFF

Default: No

If set to Yes|ON then specifies that a password must also to be entered during signon.

The signon screen will also contain an area for the user to enter their password and if the user has entered a passphrase then they must also enter their password in this field.

This password will be used when Session Manager has been configured to automatically initiate sessions that do not support passphrases.

SIGNONClass *signonclass*

Default: None

This parameter sets the name of the ESM class in which generated resources (a combination of the setting in SIGNONRESNAME and the Session Manager ACB name) will be queried to determine if the user is allowed to logon to this application. We would recommend using the APPL class. The class name can be up to 8 alphanumeric characters.

If this parameter is not set then the user is not checked against the generated resource name and they are granted access.

The specified class is `RACLISTED` into storage on the first invocation of the ISZE21SF and ISZE21PH exits. The SECFRESH command must be issued if any changes made to the class are to be reloaded into storage.

A global read-only variable `T_SIGNONCLASS` is created and is available for use by the ISZE21SF and ISZE21PH exits. The variable will contain an ESM class name up to eight characters long.

See the *Installation and Customization* manual for more comprehensive details on defining security and implementing dynamic menus.

`SIGNONResname` *`signonresname`*

Default: None

This parameter, along with `SIGNONCLASS`, sets the name of the ESM resource, which will be queried to determine if the user has authority to access this Session Manager. The resource name can be up to 31 alphanumeric characters. The resource name can be made up of any alphanumeric characters, including periods ('.'), and can be up to the length permitted by the defined ESM class, minus the length of the Session Manager ACB name.

A global read-only variable `T_SIGNONRESNAME` is created and is available for use by the ISZE21SF and ISZE21PH exits. The variable will contain the resource name, which will be appended with the Session Manager ACB name. The resource name can be made up of any alphanumeric characters, including periods ('.'), and can be up to the length permitted by the defined ESM class, minus the length of the Session Manager ACB name.

See the *Installation and Customization* manual for more comprehensive details on defining security and implementing dynamic menus.

`SIGNONAccess Yes | No | ON | OFF`

Default: No

This parameter is used by the supplied ISZE21SF and ISZE21PH exits to determine whether the user is granted access which allows them to sign on to the Session Manager application, if the ESM can not determine if a user would have access to the generated resource name (a combination of the setting in `SIGNONRESNAME` and the Session Manager ACB name).

If RACF `PROTECT ALL` is active any undefined resources will return a SAF return code of `8 - access denied`.

When using the standard supplied ISZE21SF and ISZE21PH exits the user is not granted access.

If RACF `PROTECT ALL` is not active any undefined resources will return a SAF return code of `4,4 - RACF can not make a decision`.

When using the standard supplied ISZE21SF and ISZE21PH exits a setting of `Yes` or `ON` will allow the user to be signed on. A setting of `No` or `OFF` will cause the sign on to be revoked.

A global read-only variable `T_SIGNONACCESS` is created and is available for use by the ISZE21SF and ISZE21PH exits. The variable will contain the value 'Y' or 'N'.

See the *Installation and Customization* manual for more comprehensive details on defining security and implementing dynamic menus.

`TERMINALClass` *`terminalclass`*

Default: None

This parameter sets the name of the ESM class in which generated resources (a combination of the setting in `TERMINALRESNAME` and the terminal name) will be queried to determine if the user is allowed to logon to this application from this terminal. We would recommend using the `TERMINAL` class. The class name can be up to 8 alphanumeric characters.

If this parameter is not set then the user is not checked against the generated resource name and they are granted access.

The specified class is RACLISTED into storage on the first invocation of the `ISZE21SF` and `ISZE21PH` exits. The `SECFRESH` command must be issued if any changes made to the class are to be reloaded into storage.

A global read-only variable `T_TERMINALCLASS` is created and is available for use by the `ISZE21SF` and `ISZE21PH` exits. The variable will contain an ESM class name up to eight characters long.

See the *Installation and Customization* manual for more comprehensive details on defining security and implementing dynamic menus.

`TERMINALResname` *`terminalresname`*

Default: None

This parameter, along with `TERMINALCLASS`, sets the name of the ESM resource, which will be queried to determine if the user has authority to access this Session Manager from this terminal. The resource name can be up to 31 alphanumeric characters.

The resource name can be made up of any alphanumeric characters, including periods ('.'), and can be up to the length permitted by the defined ESM class, minus the length of the terminal name.

A global read-only variable `T_TERMINALRESNAM` is created and is available for use by the `ISZE21SF` and `ISZE21PH` exits. The variable will contain the resource name, which will be appended with the terminal name. The resource name can be made up of any alphanumeric characters, including periods ('.'), and can be up to the length permitted by the defined ESM class, minus the length of the terminal name.

See the *Installation and Customization* manual for more comprehensive details on defining security and implementing dynamic menus.

`TERMINALAccess Yes | No | ON | OFF`

Default: No

This parameter is used by the supplied `ISZE21SF` and `ISZE21PH` exits to determine whether the user is granted access which allows them to sign on to the Session Manager application from this terminal, if the ESM can not determine if a user would have access to the generated resource name (a combination of the setting in `TERMINALRESNAME` and the terminal name).

If RACF `PROTECT ALL` is active any undefined resources will return a SAF return code of `8 - access denied`.

When using the standard supplied ISZE21SF and ISZE21PH exits the user is not granted access.

If RACF PROTECT ALL is not active any undefined resources will return a SAF return code of 4,4 - RACF can not make a decision.

When using the standard supplied ISZE21SF and ISZE21PH exits a setting of Yes or ON will allow the user to be signed on. A setting of No or OFF will cause the sign on to be revoked.

A global read-only variable T_TERMINALACCESS is created and is available for use by the ISZE21SF and ISZE21PH exits. The variable will contain the value 'Y' or 'N'.

See the *Installation and Customization* manual for more comprehensive details on defining security and implementing dynamic menus.

SESACB *default_acbname*

Default: SESACB ISZ001

Specifies the name of the ACB which Session Manager is to use for parallel sessions. Any combination of up to eight alphanumeric characters may be specified. Further details on the implications of the allocation of ACBs and parallel sessions can be found in the 'Setting up applications' chapter of the *Installation and Customization* manual.

SHAREAPPL *appl-statement-name*

Default: None

References an APPL statement which specifies a range of virtual terminal ACBs that are to be allocated on a user basis. A full description of ACB allocation based on this parameter is provided in the 'Choosing an ACB' section in the 'Setting up applications' chapter of the *Installation and Customization* manual.

SIGNon [Yes|No|ON|OFF]

Default: SIGNON Yes

When SIGNON=No is specified, any terminal which does not match a TERMINAL statement bypasses the signon procedure. Instead, the Menu panel is displayed. This provides a means whereby a Menu panel may be displayed at the terminal without having to define all the terminals with TERMINAL statements, and without the user being forced to sign on to Session Manager.

To cause the Signon panel to be displayed at those terminals which are not defined by a TERMINAL statement, SIGNON=Yes should be specified.

SIGNONPANEL *panelname*

Default: SIGNONPANEL SIGNON

Defines the name of the Signon panel to be displayed when a user or terminal with signon support connects to the Session Manager system.

The *panelname* may be any combination of up to 50 alphanumeric characters. The *panelname* must also be defined on a PANEL statement. For further details of the PANEL statement see the *Panels, Scripts and Variables* manual.

`SRBUFsize` *`buffer-size`*

Default: `SRBUFSIZE 1536`

Specifies the size of buffer that is required to receive any output from any application. *`buffer-size`* can be any value in the range 1-32000. The higher the value, usually the less time output needs to wait to be buffered, but this depends on the transaction rate. A higher value necessarily uses more storage.

The `Q STATS` command can be issued to display the number of session receives. These are shown as overall figures in message 308I.

`STANDBY N | nodename`

Default: `N`

Specifies a 1 to 8 character name, *`nodename`*, which defines the node for which this instance will act as a Standby. If set to `N` then this instance is not acting as a Standby instance.

Note This node must be a `SYSPLEXTYPE I` instance. If the node is a `SYSPLEXTYPE N` instance this instance will fail to synchronize with it and will not be able to recovery its users and their `RECOVERYLEVEL HIGH` and `INTERMEDIATE` sessions.

`SYSDUMP [NODUMP|DUMP|BYPASS|RTMDUMP]`

Default: `RTMDUMP`

`DUMP` If possible, recover from abends and take a SNAP dump of all storage when an abend occurs.

`NODUMP` If possible, recover from abends but do not take a dump of storage when an abend occurs.

`BYPASS` (Not recommended.) By pass any attempt to recover and then abend.

`RTMDUMP` If possible, recover from abends and request that z/OS takes a dump of storage when an abend occurs.

```
SYSPLEXGroup xxxx
   GLOBALMessages Yes|No|ON|OFF
   LINKTRace Yes|No|ON|OFF
   LINKNEtctl Yes|No|ON|OFF
   LOGSTREAMName logstreamname
   PSTIMER 1 to 32767
   STANDBYTaketime 0 to 32767
   USERStructure Yes|No|ON|OFF
   WAITFORCntltime 0 to 32767
```

Default: none

Enables a Session Manager Sysplex group name to be specified.

`xxxx` is the 1 to 4 character Session Manager Sysplex group suffix. The name will be prefixed with `ISM` to form a Sysplex group name of `ISMxxxx`.

If a SYSPLEXGroup is specified on start-up the Session Manager instance will join this group. If a number of other Session Manager instances specify the same group name, they will automatically connect to each other using the z/OS XCF (syspleX Coupling Facility) mechanism. Note that each Session Manager instance must be in the same Sysplex, either in the same instance of z/OS or other z/OS instances.

In a Sysplex environment where the multiple Session Manager instances are assigned to the same SYSPLEXGroup it is mandatory that all the Session Manager instances must share the same configuration datasets. Also any given SYSPLEXGroup must *not* have a mix of Classic and OLA Session Manager instances.

If the group name is altered via an Update or Pupdate or in OLA then the change will have no effect but message 499 will be issued. To implement the change Session Manager must be recycled.

GLOBALMessages controls the scope of all broadcasts and messages sent with the BROADCAST and MESSAGE commands. If *GLOBALMessages* has been set to (or defaults to) YES or ON, then broadcasts and messages will be distributed across the Sysplex. For example, if a broadcast command was issued in one Session Manager instance the command would be sent to all Session Manager instances in the same Sysplex group. By specifying *GLOBALMessages* NO or OFF the broadcasts and messages are restricted to the issuer's instance.

If *GLOBALMessages* YES or ON is in effect then a script or panel can set the variable t_global_msg to N for a task, and any subsequent BROADCAST or MESSAGE command issued on the task will cause the message to be restricted to the issuer's instance. Setting t_global_msg to Y sets the scope of further BROADCAST and MESSAGE commands issued on the task to the *GLOBALMessages* setting.

The *LINKTRace* and *LINKNEtctl* parameters can be specified if your local representative requires a trace of all Sysplex links. See 'TRACE LINK administrator command' on page 267.

logstreamname is a 1 to 26 character name which corresponds to the z/OS system logger log stream defined by an installation, to capture audit messages across a Sysplex group. If the *logstreamname* parameter is not specified then Session Manager will not store audit messages on a common Sysplex log that can be viewed by any instance in the group.

The PSTIMER parameter specifies the time, in seconds, that data required for session recovery will be retained by VTAM after a Session Manager Controller failure (for application sessions specified with a recovery level of High, Session Manager utilizes the VTAM Multi Node Persistent Sessions (MNPS) facility.). The value can be 1 to 32767. Default 120.

The STANDBYTaketime parameter is the time in seconds that the Session Manager Standby Controller will wait before assuming the role of the Session Manager Controller, if the Sysplex Controller is not active. The value can be 0 to 32767 (no default).

The USERStructure parameter is used to switch off the Coupling Facility User List structure. The default is Yes which means that by default, if SYSPLEXGroup is specified, Session Manager will maintain a User List structure, to be accessed by all Session Manager instances in the Sysplex group. The structure ensures all users having VTAM-connected terminals can reconnect to their disconnected

sessions. Note that this mechanism replaces the "Sample signon validation exit script (E21)" used in previous versions (see the "Parallel Sysplex support" section of the *Installation and Customization* manual). The structure is also used to support the SPLXLOCUSER TPSL function (see *Panels, Scripts and Variables* manual).

The WAITFORCntltime parameter is used when running in an environment which has been configured for application recovery. Here, the Session Manager instance will, during its initialization, ask the Controller to identify whether there are any open ACBs that are in the instance's RANGE tables.

An instance's ACBs can be open either if the instance was the subject of a SWITCHPLX command or if the instance failed and its Standby had taken over its RECOVERYLevel ACBs and sessions. If the instance is unable to contact the Controller within the WAITFORCntltime then the instance concludes that the Controller has yet to be started and therefore none of the ACBs will be in use.

The WAITFORCntltime time should be sufficient to allow the network session between the instance and the Controller to become active. If the Controller has not started then the instance initialization will be suspended for the WAITFORCntltime before finalizing its initialization.

The value can be 0 to 32767. Default 60.

SYSPLEXType C | S | I | N

Default: N

The type of Sysplex instance.

C = Controller

S = Standby Controller.

N = Normal instance, allows conventional VTAM sessions RECOVERYLEVEL NONE (and TELNET client sessions and so on) to be used by its users.

I = Recovery Instance, as for N above but also allows its users to use RECOVERYLEVEL HIGH and RECOVERYLEVEL INTERMEDIATE sessions.

```
TCP [Yes|No|ON|OFF]
    [DISPLAY [Yes|No|ON|OFF]]
    [ECLIPSEServer port-no]
        [DIAGS [Yes|No|ON|OFF]]
    [STN3270 port-no]
        [TN3270_MSG4049 [Yes|No|ON|OFF]]
        [TN3270E [Yes|No|ON|OFF]]
        [TN3270E_CONNECT [Yes|No|ON|OFF]]
    [[IBM] [IUCVname iucv-resource-id]
    [TRACE [Yes|No|ON|OFF] ]
```

Default: TCP NO

TCP YES specifies that the Session Manager TCP/IP manager is to be started as part of the Session Manager startup process. The manager is started at the end of Session Manager startup.

The TCP parameter **must** be specified for TCP/IP support and if the Eclipse server is to be used and any other relevant parameters must be actioned.

The subparameters have the following meanings:

`DISPLAY [Yes|No|ON|OFF]`

Default: `DISPLAY No`

This can be used if a TCP/IP error is encountered. If `DISPLAY` is set to `YES` and the error is recreated, it displays the TCP/IP control block that is in error.

`ECLIPSEServer` *`port-no`*
`[DIAGS [Yes|No|ON|OFF]]`

Default: None

Specifies the port number on which the Eclipse Server listens for connection requests. `TCP YES` **must** be specified to support the Eclipse Server.

When updated the change comes into effect when the system is recycled.

The subparameter which may be supplied is shown below:

`DIAGS [Yes|No|ON|OFF]`

Default: `DIAGS No`

Additional diagnostic messages and trace output will be produced if Yes or `ON` is specified. This option should only be set to `Yes` or `ON` under the guidance of your local support representative.

`STN3270` *`port-no`*`]`
`[TN3270_MSG4049 [Yes|No|ON|OFF]]`
`[TN3270E [Yes|No|ON|OFF]]`
`[TN3270E_CONNECT [Yes|No|ON|OFF]]`

Specifies a TN3270 Server is to be started at port *`port-no`* (range 1-32767). All non-3270 TELNET users will be rejected.

Note If you are replacing IBM's TELNET then specify port 23, and disable IBM's TELNET. Details on how to do this are contained in the 'Session Manager and TCP/IP' chapter of the *Installation and Customization* manual.

This parameter **must** be specified if you wish to use Session Manager without running VTAM. If the parameter is omitted, then the Session Manager TCP manager does not start the TN3270 Server.

`TN3270_MSG4049 [Yes|No|ON|OFF]`

Default: (if not specified), or `TN3270_MSG4049 No` or `OFF` is specified, is to issue message 249.

Users who access Session Manager via the Session Manager TN3270 Server would normally cause message 249 to be issued when they sign on. However, if `TN3270_MSG4049` is specified then message 4049, containing the TN3270 Client's IP address and port, will be issued in place of message 249.

TN3270E [Yes|No|ON|OFF]

Default: TN3270E No

If you specify TN3270E Y then the Session Manager TELNET server will negotiate with TN3270 clients to use certain TN3270E protocols.

TN3270E_CONNECT [Yes|No|ON|OFF]

Default: TN3270E_CONNECT No

If you specify TN3270E_CONNECT Y, which implies TN3270E Y, then TN3270E support is taken a step further. When TN3270E_CONNECT Y is in effect, the device name supplied by the client will be used to nominate a virtual terminal (VTAM ACB) which may be used when subsequently establishing forward sessions. For more information, refer to the section 'Session Manager and TCP/IP' in the *Installation and Customization* manual.

IBM [IUCVname *iucv-resource-id*]
TRACE [Yes|No|ON|OFF]

IUCVname *iucv-resource-name*

Default: TCPIP

Specifies the IUCV resource name that is to be used for TCP/IP connection.

TRACE [Yes|No|ON|OFF]

Default: TRACE NO

Specifies whether an internal trace is to be performed for TCP/IP connections.

TN3270E [Yes|No|ON|OFF]

Default: TN3270E NO

If you specify TN3270E Y then the Session Manager TELNET server will negotiate with TN3270 clients to use certain TN3270E protocols. Forward sessions subsequently initiated will, by default, use LU2 protocols (as opposed to LU0 protocols). In addition, the client can pass a device name which can optionally be used by Session Manager when allocating ACBs for forward sessions (see TN3270E_CONNECT below).

TN3270E_CONNECT [Yes|No|ON|OFF]

Default: TN3270E_CONNECT NO

If you specify TN3270E_CONNECT Y, which implies TN3270E Y, then TN3270E support is taken a step further. When TN3270E_CONNECT Y is in effect, the device name supplied by the client will be used to nominate a virtual terminal (VTAM ACB) which may be used when subsequently establishing forward sessions. This device name may relate to either a specific ACB or a range of ACBs.

For more information on TN3270E support and user-level session ACB support, see the *Installation and Customization* manual.

`TRBUFsize` *`buffer-size`*

Default: `TRBUFSIZE 128`

Specifies the size of buffer that is required to receive any input from any terminal and is used as a tuning parameter. *`buffer-size`* may be any value in the range 1-32000. See also the `TRNUMBER` parameter which specifies the number of such buffers to be allocated.

The `Q STATS` command can be issued to display the number of terminal receives in various ranges. These are given in message 491I. Message 307I gives overall figures.

`TRNUMber` *`buffer-number`*

Default: `TRNUMBER 4`

Specifies the number of buffers of `TRBUFSIZE` to be allocated. The higher the value the less time input needs to be buffered (depending on the transaction rate) however, this necessarily uses more storage. *`buffer-number`* can be any value in the range 1-32.

`VERBOSE [Yes|No|ON|OFF]`

Default: `NO`

The `VERBOSE` parameter controls the number of messages issued during a SWITCHPLX process or during VTAM application session recovery. To streamline the process, and to control the volume of potentially unnecessary messages to the audit log, the default is set to `NO`. This option should only be set to `YES` under the guidance of your local support representative.

`WORKQUE` *`workque-limit`*

Default: `WORKQUE 20`

Session Manager displays output at a terminal from one application at any one time. For each of the other active applications on a Menu screen, it can store up to 48 I/O operations, both reads and writes, in a wraparound queue. When a user returns to a system which has many outstanding I/O operations on the queue, Session Manager executes them all very rapidly one after the other to bring the display 'up to date'. For example, if there are many output screens stored in the I/O operations queue owing to a timer updated system left running, the user will see each screen flash past as Session Manager 'catches up'. On remote terminals this could take some time.

If this situation proves inconvenient or disconcerting to users, the `WORKQUE` parameter may be used to limit the I/O operations queue for applications which are not visible. *`workque-limit`* can be any value in the range 4-48.

It is recommended that applications with timer updating should not be left active while not being displayed.

Some queue entries cannot be overwritten. These belong to data streams that use the 'write structured field' function. Since these entries may contain state or function changes not detected by Session Manager, they must be kept in the queue so that the data stream eventually reaches the terminal. In some circumstances, when WSF is frequently used, a considerable amount of storage could be taken up with these entries.

This problem can be circumvented by using the MISER feature. When this is used, all writes are put into an image buffer, and the work queue becomes irrelevant. It is also possible to make the sessions SNA, and set `SNABUSY=ON`.

For SNA sessions, if `SNABUSY=ON` is set or allowed to default, unsolicited output screens are not stored. This reduces the amount of storage required by Session Manager, although display of the non-displayed screens may be slightly slower when the application is next accessed. The `SNABUSY` parameter is described later.

CHAPTER 8

PROFILE statement

The PROFILE statement can be used to specify the characteristics, facilities, and applications available to a number of users or terminals. It can also be used to override certain system options for those users and terminals using the profile.

Individual USER or TERMINAL statements can then be used to specify only those parameters that are to be different to, or additional to, those on the PROFILE. This method of coding reduces the administrative load and reduces storage usage.

Note For Session Manager 1.1.05 and higher, a user can be associated with multiple profiles (the maximum was increased from nine to eighteen at Session Manager 1.3.05). See also 'Rules for profile selection' on page 25.

The way in which the various session names, PF keys, application names and descriptions appear at the terminal depends upon the definition of the panel which is assigned to the PROFILE. This layout, defined by the PANEL statement, described in the *Panels, Scripts and Variables* manual, is known as the Menu Screen, and each PROFILE is only allowed to have one Menu screen layout assigned to it. This menu can be overridden on the USER/TERMINAL statements, if required.

Syntax and parameters

The `PROFILE` statement takes a range of parameters:

```
PROFILE profilename
    ESMLEVEL
    PASSTRY
    SIGNON
    SIGNONPANEL
    RETAIN
    SESSION|KEY
    APPLID
    CMD
    TRANSID
```

`profilename` defines the name with which the following attributes are associated. A minimum of one alphanumeric character and up to a maximum of eight may be specified. This name is used on the `USER` and `TERMINAL` statements to associate the applications defined later to the users and terminals.

Parameters

PROFILE options and limits

In addition to the parameter described below, the parameters described in 'Common end-user parameters' on page 30 are applicable.

`ESMLEVEL` *`esmlevel-value`*

Default: `0`

This is a numeric value in the range 0-9999 which controls the order in which access to ESM-managed profiles is checked for a user. All profiles with an `ESMLEVEL` of 0 will be checked in ascending alphabetical order, followed by all profiles with an `ESMLEVEL` of 1, then 2 and so on. The first 18 profile names to which a user has access will be assigned in the sequence in which they were checked. So, if USERA has ESM authority to access the following profiles:

PROFILE PROF**A** ESMLEVEL **3**

PROFILE PROF**B** ESMLEVEL **2**

PROFILE PROF**C** ESMLEVEL **2**

PROFILE PROF**D** ESMLEVEL **1**

PROFILE PROF**E** ESMLEVEL **0**

PROFILE PROF**F**

Then the profiles would be assigned to USERA in the following sequence:

PROF**E**

PROF**F**

PROF**D**

PROF**B**

PROF**C**

PROFA

Users who do not have ESM authority for any of the profiles will be assigned the supplied DEFPROFILE value from the configuration.

See also ESMPRCLNM parameter on page 107.

PASSTry *passtry-limit*

Default: SYSTEM statement value, or product default: 0

Specifies a limit to the number of times any user can attempt to sign on to Session Manager. *passtry-limit* may be any value in the range 0-999. When the limit is exceeded, the terminal task for the user is suspended, the terminal is locked, and message 503I is sent to the console.

For the terminal to be used again, a PASSFREE command must be issued.

If PASSTRY 0 is specified, there is no limit to the number of times the user can attempt to sign on.

RETAIN [Yes|No|ON|OFF]

Default: RETAIN No

Any 'Profile Options and Limits' parameters and any session default parameters may be placed between a RETAIN and a RETAIN OFF pair. Parameters enclosed in this way are kept in force after a signon, unless specifically overridden by defining the same parameters on the USER statement. For example:

```
PROFILE ABCD
RETAIN
MISER ALARM OFF
RETAIN OFF
```

would cause MISER to be set on and the terminal alarm set off, unless specifically overridden by a USER statement.

The end of the PROFILE statement, or the presence of a SESSION or KEY definition, implies RETAIN OFF.

Note (Applicable to existing users of a Classic configuration – that is, all configuration definitions are stored in members of PDS(s) allocated to the DDNAME of CONFIG.) If you upgrade toSession Manager 1.1.10 or higher, and then run the OLA Enabler to implement the new format configuration, then any RETAIN parameters will be *removed* from any TERMINAL or PROFILE statements found.

SIGNon Yes|No|ON|OFF

Default: SYSTEM statement value, or product default: Yes

Specifies the way in which the user accesses the Session Manager system and the applications which are therefore available.

A specification of SIGNON=N prevents the signon panel being displayed. The Menu Panel is then, by implication, associated with the terminal (through the associated profile) rather than the user.

If SIGNON=N is specified, the Menu Panel implied by the associated profile is displayed as soon as a user logs on to Session Manager. If, however, the associated profile has AUTOSELECT specified, the terminal will immediately be logged onto the application associated with the defined session id, bypassing the Menu screen completely.

A specification of SIGNON=Y causes the appropriate signon panel to be displayed at the terminal. The implication in this case is that the Menu panel is to be associated with the user signing on. Therefore, Session Manager would expect to find a matching USER statement for that user; PROFILE options are lost unless RETAIN is in effect.

The Session Manager user associated variable t_signed_on may be used to refer to the SIGNON value; for details, see the product's *Panels, Scripts and Variables* manual.

SIGNONPANEL *signon-panel*

Default: SYSTEM statement value, or product default: SIGNON

Specifies the name of the signon panel which is to be displayed at the terminal when a user signs on to the Session Manager application. This panel is only displayed when SIGNON=Y has been specified or defaulted.

This *signon-panel* should be defined using the PANEL statement which is documented in the *Panels, Scripts and Variables* manual.

PROFILE session definitions

In addition to the parameters described below, the parameters described in 'Common session parameters' on page 51 (which are the session defaults) are applicable.

```
SESSION DEFAULTS                  |
SESSION nnnn [selection-commands] |
KEY PFnn
```

The SESSION or KEY parameter may be used to denote the start of a Menu panel detail definition, or to specify a Session Manager command to be executed when the session detail number is entered, a *selection-command* is issued, or the PF key is pressed. The SESSION parameter can also be used to specify default values for all sessions in this PROFILE definition.

The subparameters following the SESSION or KEY parameter describe either the session number or PF key usage, and the action which Session Manager takes when the session is selected or the PF key is pressed, or the default actions for all sessions in the profile.

Each defined session may be allocated a session detail number, *nnnn*, from 1 to 9999. Each session may also be allocated a list of *selection-commands* or pseudonyms which, when a pseudonym is issued from the Menu panel, starts the associated session, or issues the associated command. Each pseudonym may also be a PA or PF key, or up to eight alphanumeric characters, and may be a Session Manager keyword, provided that it is enclosed in quotes. For example, a session could be defined as:

```
SESSION 1 PROD CICS APPLID CICS01
```

so you can enter either the session detail number 1, or either of the selection commands PROD or CICS, to access the application CICS01 from the Menu panel.

Any of the following session parameters can be placed before any specific session definitions. When this is done, the parameter is taken as defining a default value which is used whenever a session definition does not contain that parameter. This is the only instance in which session options are merged. A session definition parameter always overrides a session default parameter.

When DEFAULTS is specified, the parameters of this SESSION definition override any default parameters which have been specified **before** the first SESSION or KEY definition.

APPLID *VTAM-applid*|*appl-statement-name*|[REFappl [Yes|No|ON|OFF]]

Specifies the name of the application that Session Manager is to log onto when the session is selected.

VTAM-applid

This can be used to specify directly the VTAM application that Session Manager is to log onto when this session is selected. A name up to eight alphanumeric characters long can be specified. It follows that each application name specified should be defined to VTAM by an APPL macro.

appl-statement-name

The APPL name can be specified. This ensures that a particular set of options is always used for a particular application, unless specifically overridden.

REFappl [Yes|No|ON|OFF]

The REFAPPL option can be used to indicate which of the two options is being used. Specifying REFAPPL=N means that Session Manager does not attempt to find an APPL statement. REFAPPL=Y means that Session Manager searches for a matching APPL statement, and if found, uses the parameters from that statement. An error is not generated by an unmatched reference, but a warning message is issued.

CMD *command*

Defines a Session Manager command to be executed when the PF key is pressed, or the session id is selected either by number or by synonym. Valid commands are described in 'Commands' on page 215.

The maximum length for the executed command is 80 bytes and substitution variables may be used in the command string.

The chapter 'Session Manager Variables' in the *Panels, Scripts and Variables* manual lists the valid variables and 'Variable substitution in control statements' on page 17 of this manual gives the rules that apply. If the string contains any Session Manager keywords, or embedded spaces, the complete string should be enclosed in quotes.

When the CMD subparameter is used, all other subparameters, apart from DESC, are ignored for the session.

It can also be used to indicate that the session is a sub-menu placeholder by specifying a value of `CMD 'ISZSM` *`profilename`*'. For more information see the description of 'Session Manager Submenus' in the *User and Administration* manual.

`TRANSID` *`transid-list`*

Specifies one or more Session Manager transids to be associated with the session id. A transid may then be used to transfer between applications rather than the escape command sequence. It may also be entered into the command line of the Menu screen, rather than a session id being selected or a PF key being pressed.

Each transid in the list can be any combination of eight alphanumeric characters, a PA or PF key, the ATTN or CLEAR key, or the lightpen (PEN). The list is delimited by the presence of a Session Manager parameter or subparameter. Alternatively, the list may be enclosed in quotes, or any pair of text delimiters.

CAUTION If CLEAR is used, the application's screen will be cleared before escaping.

Whenever Session Manager detects a transid character sequence at the start of the first input field in a screen, it immediately transfers control to the application associated with the session for which it has been specified.

When a transid is specified as a key (PF*n*, PA*n*, ATTN or CLEAR) or a lightpen (PEN), control is transferred to the application associated with the session whenever the key is pressed or the lightpen detected. Even when the session is defined as being invoked by a PF key from the Menu screen, the PF key used as a transid does not need to be the same. For example, if PF5 is defined as accessing a CICS application, PF17 may be defined as providing immediate access by coding the definition as follows:

```
Key PF5 Applid CICSA
Transid PF17
```

Care must be taken when using PF keys or sequences/character strings for transids, since no system which is running from this particular Menu screen will be able to use them, unless the `PASSTRANSID` has been specified for the relevant sessions.

CHAPTER 9

USER statement

The USER statement can be used to override certain System and Profile characteristics, facilities, and applications. It can also be used to alter only certain parameters, or provide additional parameters to those specified on a PROFILE statement that is to be used by a user. By using a PROFILE statement to define the basic characteristics available to a number of users, and specifying only those parameters that are to be unique for the user defined by a USER statement, both administrative effort and storage usage is reduced.

If a user is to access the Eclipse PC front-end, then a USER statement must be defined for that user.

Syntax and parameters

The USER statement takes a range of parameters:

```
USER userid|userid_pattern
    PASSWORD
    PROF
    RENUMDUP
    SESSAUTOS
    SESSPRI
    TRACE
```

userid specifies the name of the user with which the attributes that follow, are to be associated. Each *userid* must be defined only once and may be any combination of up to eight alphanumeric characters. Generic userids can also be specified.

Generic user names may contain the '*' and '+' characters as wild character arguments. The '*' character indicates that any number of any character in this position is acceptable, and the '+' character indicates that one of any character in this position is acceptable. For example:

USER E+G – allows any users who have the character E in the first position, G in the third position, and any character in the second position, to log on–user EPG, for example, could log on.

USER E* – allows any users whose userid starts with the character E to signon.

When a user signs on to Session Manager, a search is performed for a matching non-generic USER statement. If one is not found, then the generic USER statements are searched. The generic USER statements are arranged according to the length of the user name specification; the longest specification is searched first. In the above example, E+G would be searched before E* if, for example, user EBT signed on.

Parameters

Parameters specific to USER statement

PASSWORD *password*

Specifies a *password* which the user accessing Session Manager must enter before gaining entry to a Menu screen. This password is ignored when the Session Manager User exit Signon Validation exit point (E21) is used.

PROF *profilename*
[PROF *profilename*]

Default: SYSTEM statement value, or product default – PROFILE.

Specifies the name(s) of the profile(s) to be associated with this user.

For Session Manager 1.1.05 and higher, a user can be associated with multiple profiles (the maximum was increased from nine to eighteen at Session Manager 1.3.05) and their sequence is important. If a particular attribute value (common enduser or common session) has not been defined by the user, the profiles will be searched in the following order and the first value defined explicitly for that attribute will be used:

First PROFILE in the list

Second PROFILE in the list

…

Last PROFILE in the list

Note If using the ESM, by setting the ESMPRFCLNM sub-parameter on the SECURITY parameter on the SYSTEM statement (see page 103), to assign PROFILE(s) then the PROF parameter of USER statement is ignored.

See also 'Rules for profile selection' on page 25.

RENUMDUP *nnnn*

Default: none

If sessions are duplicated, they will normally be excluded from the main menu display. The RENUMDUP parameter allows the user to specify a starting number *nnnn* for renumbering any duplicate sessions encountered during display on the main menu. No permanent change is made to the configuration files. *nnnn* can be a value from 1 to 9999.

If this parameter is not present, duplicate sessions will be ignored.

For example, assume the user has three profiles:
PROFA with sessions 101-105
PROFB with sessions 101-105
PROFC with sessions 201-205
If the user applies the RENUMDUP 201 configuration parameter, the PROFB sessions would be renumbered to 206-210. This is because the 201-205 sessions already exist in PROFC and are not duplicates, therefore the first available session number to use for renumbering of duplicates is 206.

If the number of duplicate sessions encountered has caused the next available number to exceed 9999 (the upper limit for RENUMDUP) then any further duplicate sessions encountered will be ignored and message 4236 will be issued to the audit log when the user signs on. To see all duplicate sessions, if possible, reduce the starting point number on the RENUMDUP parameter to allow for the number of duplicate sessions required.

SESSAUTOS Yes|No|ON|OFF T|N|A|C
sestype|session_number|application|command

Default: none

This parameter identifies sessions and their current 'Autostart' user setting for both Classic and OLA systems. It is specific to the USER definition (which may contain multiple SESSAUTOS parameters) although OLA does not display it in the 'Enduser specific parameters' list.

The SESSAUTOS parameters will determine if the session is Autostarted.

In a Classic system, if a session is found that satisfies the SESSAUTOS criteria, that session, whether USER defined or inherited from a PROFILE statement, will be either Autostart or not Autostart depending on the setting. If no session that is applicable to the user matches the SESSAUTOS criteria, then the SESSAUTOS parameter will be ignored.

In the case of OLA, the value of `Yes|No|ON|OFF` is saved from the 'AUTOSTART' column on the Session List panel. `Yes` and `On` are displayed as `Y`. `No` and `OFF` are displayed as `N`.

In a Classic system the `USER` statements are edited in the usual manner. Any changes made that might affect the session selected, for example the administrator adding or deleting `PROFILE` defined sessions, will have no effect on the `USER SESSAUTOS` definitions. When the user signs on, whichever sessions satisfy the `SESSAUTOS` criteria will be checked against the Autostart setting.

A common enduser parameter `SESSAUTOSAppl` (see page 46) controls whether or not the setting of Autostart by other than `SESTYPE` and session number is to be allowed in an OLA system.

Note If more than one session has a `SESSAUTOS C` specified then only the last session that has a `SESSAUTOS C` specified will be autostarted, as only one Command session can be autostarted at logon. Please note that any `SEQUENCE` and/or `SESSPRI` parameters will affect the order of your sessions. If you need to autostart more than one Command then we recommend that you create a Command Script that contains calls to all the required Commands and define this Command Script to a session that is autostarted.

`SESSPRI T|N|A|C` *`sestype|session_number|application|command`*

Default: none

This parameter affects the sequencing of sessions on the user's Session List, allowing specified sessions to be given 'priority' so that they appear at the top of the user's list.

The parameter specifies whether the session is being identified by session type ('T'), number ('N'), application ('A') or command ('C') with the relevant value appearing alongside.

The `SESSPRIAPPL` common enduser parameter (see page 46) controls whether sessions can be prioritized only by session type and session number, or additionally by application and command.

If `SESSPRIAPPL` is set to `No`, or neither 'A' nor 'C' is specified, and the session being given priority has a non-zero `SESTYPE` (see page 67) then it will be identified by its session type; otherwise the session number will be used.

One `SESSPRI` parameter will be specified for each session that is to be designated as a 'priority' session.

The sessions will appear on the user's menu in the same order that their `SESSPRI` parameters appear on the `USER` statement, whether the `SESSPRI` sessions are user-specific or inherited.

If more than one session satisfies a type A or C `SESSPRI` (for example, if there are multiple sessions defined with the same application) then the first occurrence only will be prioritized.

If using OLA-enabled Session Manager, the `SESSPRI` parameter will be created, modified, and deleted automatically when sessions are specified as priority sessions within the Session List in OLA; the parameter itself will not appear in OLA.

The ability to set priority sessions is designed for OLA-enabled systems (see the section on setting priority sessions in the *Online and Batch Administration* manual for details). But if you are using a Classic configuration, priority sessions may be configured with the help of your local Support Representative (sessions may be prioritized by coding `SESSPRI` parameters of any valid type on the `USER` definition).

Note This parameter will be ignored for users who have sub-menus configured.

`TRACE INTernal|DATa|MISER|Vtam|NETDATA`

This optional parameter would normally only be specified at the request of your local support representative.

`INTERNAL` activates the Session Manager internal storage tracing routines for the user defined by this statement.

`DATA` activates the tracing routines for the data flowing between the terminal and the application.

`MISER` activates the tracing routines for the operation of MISER between the terminal and Session Manager.

`VTAM` activates the tracing routines for the VTAM data tracing. This shows all VTAM events and the data flowing between the terminal and the application.

`NETDATA` activates the tracing routines for the operation of link traffic for network sessions. `DATA` tracing is implied. Tracing is performed for all active tasks.

The internal trace output is written to the SYSOUT job control statement associated with the DDNAME 'DUMP'.

Trace output is written to the route designated by the `TRACEROUTE` statement.

Other parameters that may be specified on USER statement

All other parameters and keywords that may be supplied on the `USER` statement are equivalent to those specified in 'PROFILE options and limits' on page 122 (excluding `PASSTRY`, `RETAIN`, `SIGNON` and `SIGNONPANEL`) and 'PROFILE session definitions' on page 124.

Any of these options and definitions, when specified on the `USER` statement, override any corresponding ones specified on the associated `PROFILE` statement (or `SYSTEM` statement).

There may be some variation on this rule regarding session defaults and definitions. For details, refer to 'Configuration defaults and overrides' on page 24.

CHAPTER 10

TERMINAL and LU statements

The TERMINAL statement can be used to associate specific attributes with a particular terminal in the system. Like the SYSTEM statement, the TERMINAL statement is completely optional. If it is omitted, IBM Session Manager for z/OS would expect to associate characteristics with the user who signs on, rather than the terminal at which the signon is issued. In such a situation, USER statements should be defined instead of TERMINAL statements. It is, however, possible to use parameters from a TERMINAL statement even when there is an applicable USER statement. This is achieved by the RETAIN parameter described on page 135 in this chapter.

The LU statement performs a similar function to the TERMINAL statement in that it associates specific attributes with a particular terminal in the system.

When a terminal is started, or a user signs on, Session Manager searches for a LU statement to match the terminal name and type. If one cannot be found, Session Manager then searches for a TERMINAL statement to match the terminal name.

The LU statement is also completely optional. If omitted, Session Manager would expect to associate characteristics with a TERMINAL statement, or the user who signs on, rather than the terminal at which the signon is issued. In such a situation, USER statements should be defined.

TERMINAL syntax and parameters

The TERMINAL statement defines a terminal name regardless of the terminal type.

```
TERMINAL termid|termid_pattern
    PASSTRY
    PROF
    RETAIN
    SIGNON
    SIGNONPANEL
    TRACE
```

This specifies the name of the terminal with which the attributes that follow are to be associated. The *termid* can be an LU name.

Generic *termids* may be specified, enabling groups of terminals to be defined. This method enables a profile to be associated with many terminals using only a single statement, and is therefore useful in reducing the number of control statements required to generate the Session Manager system.

Creating generic terminal groups is achieved by using the special characters '*' and '+' when specifying the *termid*. An '*' means 'any number of any character', while a '+' means 'one of any character'. Any combination of alphanumeric characters, asterisks and plus signs is acceptable. For example:

```
TERMINAL T* PROF PROFILEA
TERMINAL TM001 PROF PROFILEB
TERMINAL 0* PROF PROFILEC
TERMINAL 0048 PROF PROFILED
```

associates profile 'PROFILEA' with all terminals which have an id beginning with a 'T', except for the terminal TM001, which uses the profile 'PROFILEB'; 'PROFILEC' with all terminals which have an id beginning with '0', except terminal '0048' which uses the profile 'PROFILED'. A second example:

```
TERMINAL ++M* PROF PROFILEE
TERMINAL ++9* PROF PROFILEE
```

causes all terminals which have an 'M' in the third position to use profile 'PROFILEE' and causes all terminals which have a '9' in the third position to use profile 'PROFILEE'.

If a terminal signing on to Session Manager matches both a generic and a specifically defined termid on a TERMINAL statement, the specific definition takes precedence. When a terminal signs on to Session Manager and it matches more than a single TERMINAL generic name specification, it cannot be determined which definition will be used.

Parameters

In addition to the parameters described below, the parameters described in 'Common end-user parameters' on page 30 are applicable; also, for TERMINAL session definitions, the parameters described in 'PROFILE statement' on page 121 are applicable.

PASSTry *passtry-limit*

Default: PROFILE statement value, SYSTEM statement value, or product default: 0

Specifies a limit to the number of times any user can attempt to sign on to Session Manager. *passtry-limit* may be any value in the range 0-999. When the limit is exceeded, the terminal task for the user is suspended, the terminal is locked, and message 503I is sent to the console.

For the terminal to be used again, a PASSFREE command must be issued.

If PASSTRY 0 is specified, there is no limit to the number of times the user can attempt to sign on.

PROF *profilename*
[PROF *profilename*]

Default: SYSTEM statement value, or product default: PROFILE

Specifies the name(s) of the profile(s) to be associated with this terminal.

For Session Manager 1.1.05 and higher, a user can be associated with multiple profiles (the maximum was increased from nine to eighteen at Session Manager 1.3.05) and their sequence is important. If a particular attribute value (common enduser or common session) has not been defined by the user, the profiles will be searched in the following order and the first value defined explicitly for that attribute will be used:

First PROFILE in the list

Second PROFILE in the list

…

Last PROFILE in the list

See also 'Rules for profile selection' on page 25.

RETAIN [Yes|No|ON|OFF]

Default: RETAIN No

Any 'Profile Options and Limits' parameters and any session default parameters may be placed between a RETAIN and a RETAIN OFF pair. Parameters enclosed in this way are kept in force after a signon, unless specifically overridden by defining the same parameters on the USER statement. For example:

```
TERMINAL ++2*
RETAIN
MISER ALARM OFF
RETAIN OFF
```

would cause all terminals with a name conforming to the pattern to have MISER set on and the terminal alarm set off, unless specifically overridden by a USER statement.

The end of the TERMINAL statement, or the presence of a SESSION or KEY definition, implies RETAIN OFF.

Note (Applicable to existing users of a Classic configuration – that is, all configuration definitions are stored in members of PDS(s) allocated to the DDNAME of CONFIG.) If you upgrade to Session Manager 1.1.10 or higher, and then run the OLA Enabler to implement the new format configuration, then any RETAIN parameters will be *removed* from any TERMINAL or PROFILE statements found.

`SIGNon Yes|No|ON|OFF`

Default: `SYSTEM` statement value, or product default: `Yes`

Specifies the way in which the user at this terminal accesses the Session Manager system and the applications which are therefore available.

A specification of `SIGNON=N` prevents the signon panel being displayed. The Menu Panel is then, by implication, associated with the terminal (through the specification of the `PROF` parameter) rather than the user.

If `SIGNON=N` is specified, the Menu Panel implied by the associated profile (defined using the `PROF` parameter) is displayed as soon as a user logs on to Session Manager from this terminal. If, however, the associated profile has `AUTOSELECT` specified, the terminal will immediately be logged onto the application associated with the defined session id, bypassing the Menu screen completely.

A specification of `SIGNON=Y` causes the appropriate signon panel to be displayed at the terminal. The implication in this case is that despite there being a `TERMINAL` statement for this terminal, the Menu panel is to be associated with the user signing on, rather than the terminal. Therefore, Session Manager would expect to find a matching `USER` statement for that user; `TERMINAL` options are lost unless `RETAIN` is in effect.

The Session Manager user associated variable `t_signed_on` may be used to refer to the `SIGNON` value; for details, see the product's *Panels, Scripts and Variables* manual.

`SIGNONPANEL` *`signon-panel`*

Default: `SYSTEM` statement value, or product default: `SIGNON`

Specifies the name of the signon panel which is to be displayed at this terminal when a user signs on to the Session Manager application. This panel is only displayed when `SIGNON=Y` has been specified or defaulted.

This *`signon-panel`* should be defined using the `PANEL` statement which is documented in the *Panels, Scripts and Variables* manual.

`TRACE INTernal|DATa|MISER|Vtam|NETDATA`

This optional parameter would normally only be specified at the request of your local support representative.

`INTERNAL` activates the Session Manager internal storage tracing routines for the terminal task defined by this statement.

`DATA` activates the tracing routines for the data flowing between the terminal and the application.

`MISER` activates the tracing routines for the operation of MISER between the terminal and Session Manager.

`VTAM` activates the tracing routines for the VTAM data tracing. This shows all VTAM events and the data flowing between the terminal and the application.

`NETDATA` activates the tracing routines for the operation of link traffic for network sessions. `DATA` tracing is implied. Tracing is performed for all active tasks.

The internal trace output is written to the SYSOUT job control statement associated with the DDNAME 'DUMP'.

The `MISER`, `VTAM` and `NETDATA` data trace output is written to the route designated by the `TRACEROUTE` statement.

LU syntax and parameters

The `LU` statement defines an LU-type terminal.

`LU` *`luname|luname_pattern`*

Parameters

The `LU` statement parameters are the same as for the `TERMINAL` statement.

CHAPTER 11

APPL statement

The APPL statement defines the options to be used for an application or system. It is not essential that the options are defined on this statement, but it provides a convenient method of ensuring that specific options are used for a particular application or system.

When an application is unable to cope with parallel sessions, it cannot share a single ACB between terminals that are logged on, but requires a unique ACB for each terminal. The most notorious application in this category is CICS, and another example is IMS™. A convenient method of providing ACBs for such applications is to define free pools of ACBs from which an ACB may be allocated for each terminal.

In combination with the RANGE statement, a set of ACBs can be defined, one of which IBM Session Manager for z/OS can allocate when a session is established with such an application.

Syntax and parameters

The APPL statement takes a range of parameters:

```
APPL applname
    APPLID
    INDRAnge
    INQUIRE
    RECOVERYLevel
    TERMLOGMODE T
```

applname specifies the name by which the APPL statement may be referenced by the APPLID subparameter, with a REFAPPL=Y option, of a session definition on a PROFILE, USER, TERMINAL or LU statement. The *applname* can be any combination of up to eight alphanumeric characters.

Parameters

In addition to the parameters described below, the parameters described in 'Common session parameters' on page 51 are applicable.

APPLID *VTAM-applid*

Default: APPL statement *applname*

The *VTAM-applid* option can be used to specify the name of a VTAM application to which the options, defined by this APPL statement, are to apply.

The *applid* can be any combination of up to eight alphanumeric characters but should be an application which is defined to VTAM on a VTAM APPL definition statement. If omitted, the default is the APPL statement name.

INDRAnge *appl_name*

The Indirect Range parameter allows the user to define a number of APPL statements, which will allocate a unique ACB from a single set of ranges defined on an Indirect APPL.

If the Indirect APPL does not exist, a warning message ISZ0520W is issued at configuration load time. For further information, see the *Messages and Codes* guide.

Note The ranges referred to in the appl_name will only be used if all the following conditions are true:

- There is no range specified in the local APPL.
- SHAREAPPL or User level session ACBs are not in use.
- An explicit ACB has not previously been set, for example in an INITSCRIPT.

For example, consider sessions using the following APPL definitions:

```
APPL S01CICS        INDRAnge IRANGE
APPL S02CICS        INDRAnge IRANGE
APPL S03CICS        INDRAnge IRANGE
APPL S04CICS        INDRAnge IRANGE
APPL S05CICS        INDRAnge IRANGE
```

Each of the sessions would get their unique ACB allocated from the following `IRANGE APPL` definition:

```
APPL IRANGE
     TERMLOGMODE * ACBRANGE R1
     TERMLOGMODE D4A32792 ACBRANGE R2
     TERMLOGMODE D4A32795 ACBRANGE R3
```

`INQUIRE [Yes|No|ON|OFF]`

Default: `INQUIRE Yes`

Prevents or allows a `VTAM INQUIRE` on the application defined by the `APPL` statement.

`INQUIRE=N` can be useful in the situation where a dummy application is displayed unprotected on a menu. This allows the user to key in any VTAM application they wish. If `INQUIRE=Y` is in force the dummy application will be inquired upon, which can cause unnecessary overhead. This can be prevented by specifying `INQUIRE=N`. For example:

```
SESSION 4 APPLID USER REFAPPL=YES
    DESC 'an application of your choice'
APPL USER INQUIRE=NO
```

The frequency of the inquires is governed by the `INQINTERVAL` parameter of the `SYSTEM` statement.

`RECOVERYLevel High | Intermediate | None`

Default: None

Specifies the Sysplex recovery level. Defines the recoverability of forward sessions to the application when the ACB used is being opened. The parameter only takes effect when an ACB is opened. If the ACB is already opened when the session is started then any change to this parameter will have no effect. This parameter can be defined as a Common End User parameter on the USER, TERMINAL, PROFILE or SYSTEM statements or on the APPL statement. If defined as a Common End User parameter and set to `HIGH` or `INTERMEDIATE` then this will override the APPL statement. If `NONE` is specified then the APPL setting applies.

`HIGH` = Recover the application session if the owning (target) Session Manager node fails or the Session Manager Controller fails. Can only be used with `SYSPLEXTYPE I` instances.

`INTERMEDIATE` = Recover the application session if the owning (target) Session Manager node fails. Can only be used with `SYSPLEXTYPE I` instances.

`NONE` = No recovery will be attempted for this application. Can be used with `SYSPLEXTYPE I` or `SYSPLEXTYPE N` instances.

See the 'Parallel Sysplex support' chapter of the *Installation and Customization* manual for further information.

`TERMLogmode` *`logmode-entryname`*`|* ACBRange` *`rangename`*

Specifies the logmode table entry name, and the ACB range to be used when a user logs on to this application.

logmode-entryname | *

Defines the logmode table entry from which the characteristics are to be used when allocating this ACB range for this application. The names which can be used are the same as those defined for the LOGM*nnx* parameter of the SYSTEM statement. This gives the user the ability to force Session Manager to use specific attributes for terminals logging onto this application, irrespective of the bind image received when the user originally logged on.

Alternatively, an asterisk ('*') may be specified, in which case the entry name is that implied by the bind image received from VTAM at logon time.

ACBRange *rangename*

Defines the *rangename* from which an ACB is to be selected when a user logs on to the application defined by this APPL statement. The RANGE statement is used to generate ranges of ACBs: this is described in the following chapter.

Note in particular that several pairs of TERMLOGMODE and *rangename* parameter combinations may appear for one applname. This enables the user to control the range (and therefore the session characteristics) from which ACBs are selected. When all the ACBs from the 'first choice' range have been allocated, Session Manager selects the 'best fit' ACB from another range. This technique is described in the section 'Best Fit ACB Allocation for RANGE Statements' in the 'Setting up applications' chapter of the '*Installation and Customization*' manual.

The TERMLOGMODE parameter is mutually exclusive with the ACB parameter.

CHAPTER 12

RANGE statement

The RANGE statement is used to define ranges of ACBs from which IBM Session Manager for z/OS can select when establishing a session between the terminal and an application, such as CICS, which is unable to cope with parallel sessions.

Ranges of ACB names may overlap, and once an ACB is allocated in one range, it is made unavailable in all ranges for the same APPL statement, until the session with the application is terminated, at which point it becomes available again in all ranges. Since each range defines a free pool of ACBs, any ACB names that are to be reserved for particular applications must not be defined in a range.

Syntax and parameters

The RANGE statement is associated directly with the APPL statement, and takes a range of parameters:

```
RANGE acbname_range
    FROM start_acbname [TO stop_acbnumber]
    [HEX|DEC]
    [RRA Yes|No|ON|OFF]
```

*acbname_range s*pecifies the name to be associated with the following set of ACBs. This is the name which is used on an APPL statement to link a set of ACBs with a particular application. This name may be referenced on more than one APPL statement if required. It can be any combination of alphanumeric characters up to eight bytes long.

Parameters

FROM *start_acbname* [TO *stop_acbnumber*]

Specify the name of the ACB from which this range of ACBs is to begin. The name **must** begin with at least 1 alphabetic character followed by a number of numeric digits. The maximum number of ACBs able to be defined in a range is determined by the number of trailing numerics specified on the FROM parameter *start_acbname.* Therefore, in order to generate a range which is to contain at least one hundred ACBs, three trailing digits would be required and so on. The maximum number of trailing digits is five. If a *start_acbname* is specified without the optional TO parameter, a 'range' is generated with only a single ACB in it. No more than 65535 ACBs can be defined in a range.

The TO parameter specifies the number of the ACB, *stop-acbnumber*, at which the range of ACBs ends. Session Manager only accepts an upper limit numeric value for the *stop_acbnumber*, without the preceding alphabetic character(s). The value of the trailing numerics on the TO parameter must be greater than that specified in the numeric portion of the *start_acbname*.You can specify a maximum of five digits in the TO parameter.

For example, to generate a range of ACBs from ACB20 to ACB50, simply code 'FROM ACB20 TO 50'. Similarly, to generate a single ACB with the name SYSACB1, specify 'FROM SYSACB1', with no associated TO parameter.

Note If all the ACBs available in a range have been allocated, and a user attempts to logon to an application which specifies this range, Session Manager will attempt to allocate an ACB with the same model characteristics but with a different protocol. That is, if the logon is from a non SNA terminal it attempts to allocate an ACB of the same model terminal but SNA and vice versa. This process is described in the section 'Best fit ACB allocation for RANGE statements' in the 'Setting up applications' chapter of the *Installation and Customization* manual.

HEX|DEC

By default, ACB range numbers are specified in decimal. If hexadecimal numbers are required, then the HEX option must be specified.

RRA

If RRA is set to No (the default) then the ACBs are allocated using an algorithm that selects the lowest available ACB in the specified range.

If RRA is set to Yes, or RRA is specified on its own, then the ACBs are allocated using a Round-Robin Algorithm that allocates the next available ACB in ascending sequence, wrapping to the lowest available ACB in the specified range if an ACB higher than the previously allocated ACB is not available.

One advantage of specifying RRA is to avoid users perpetually selecting an ACB that, for whatever reason, is not compatible with the application. In this case, specifying RRA causes the next ACB in the range to be allocated instead of the same ACB being selected repeatedly for subsequent session starts.

See also 'STOPACB administrator command' on page 265.

CHAPTER 13

LINK statement

The `LINK` statement defines the characteristics of a link to another IBM Session Manager for z/OS network node. Each Network node can have as many links defined to other network nodes as necessary.

Just one type of link may be defined – VTAM LU0, which can be used to connect nodes when there is an VTAM route between them.

`LINK` statements must be defined for Session Manager networking to function. Several `LINK` statements can be defined in the configuration file for the local node. Many of the link attributes may be left to default. However, nodes connected using Session Manager Sysplex networking (with `SYSTEM` statement parameter `SYSPLEXGROUP` specified) should not define `LINK` statements for nodes within the same Sysplex group as these nodes are connected automatically.

Syntax and parameters

The LINK statement takes a range of parameters:

```
LINK linkname
    VTAM
       TO
       LOGMODE
       RECVANY
       OPENRETRYLIM
       OPENRETRYINT
    RECONINTVL
    BUFSIZE
    STARTLINK
    TRACE
    ISZ
    ACTIVATE
```

linkname specifies the name to be associated with this link in the network. This name is used in commands issued to the link and associated messages from the link. The *linkname* associated with each LINK statement should be unique in the network.

Parameters

Vtam *local-acb*

Specifies that a VTAM LU0 link is to be defined. *local-acb* is the name of the ACB to be used for this end (the local end) of the link. This can be an ACB used for other links (that is, shared ACB) but it is recommended that a dedicated ACB be used for each link defined, as this provides the greatest operational flexibility.

To *remote-acb*

Identifies the remote end of this link. *remote-acb* is the name of the ACB at the remote end of the link. (This name should correspond to a *local-acb* value on the configuration LINK statement at the remote node.)

LOGMode *logmode*

Default: LOGMODE S3270

Specifies the logmode to be used for the link session. The default is S3270.

RECVany *n*

Default: RECVANY 2

Specifies the number of RECEIVE ANY RPLs to be defined. *n* may be any value in the range 1-8. If the *local-acb* specified for the link is a shared ACB, the number *n* is added to the RECVANY specification on any other links using that ACB. The default should be adequate and there is currently no benefit in increasing it.

OPENRETRYLIM *nnn*

Default: OPENRETRYLIM 30

Specifies the number of times that OPEN ACB should be attempted. *nnn* may be any value in the range 1-999. The default is 30. This parameter is only relevant when Session Manager is likely to be initialized before VTAM, and therefore the ACB cannot initially be opened.

`OPENRETRYINT` *seconds*

Default: `OPENRETRYINT 60`

Specifies the number of seconds between each OPEN ACB retry attempt. *seconds* can be any value in the range 1-300 seconds. This parameter is only relevant when Session Manager is likely to be initialized before VTAM. The interval between attempts to contact the other end of the link is controlled by the `RECONINTVL` parameter, described later in this section.

`RECONIntvl` *seconds*

Default: `RECONINTVL 60`

Specifies the interval between attempts to connect or reconnect the link. *seconds* may be any value in the range 1-300 seconds.

`BUFsize` *buffersize*

Default: `BUFSIZE 8`

Specifies the receive buffersize in Kbytes. *buffersize* may be any value in the range 2-64K.

When a link connection is established, both ends of the link inform each other of their `BUFSIZE` value. The resulting receive buffersize used by the link session is the smaller of the local or remote specification. For example, if the local end of the link has the default value of 8K, and the remote end has a `BUFSIZE` value set to 4K, then the effective receive buffersize used is 4K.

`STARTLink Yes|No|ON|OFF`

Default: `STARTLINK No`

Specifies whether the link is to be activated as soon as the configuration process has completed. If set to `Yes`, the link is activated without operator intervention as soon as the configuration process completes. If set to No, then the Operator must issue the `STARTLINK` *linkname* command to activate the link. See 'Privileged user commands' on page 245 for details of the `STARTLINK` command.

`TRace Yes|No|ON|OFF [NETCTL]`

Default: `TRACE No`

Specifies whether tracing is to be performed for the link on startup. The default is No.

The optional `NETCTL` operand can only be specified with `TRACE=Yes`. It causes external tracing of all network control data passing through the link from the point at which it is activated. Specifying `TRACE=Yes`, without `NETCTL` performs a trace of **all** data passing through the link.

Tracing should only be specified at the request of your local support representative.

`ISZ`

Default: `ISZ`

Specifies the type of system at the other end of the link.

`ISZ` is currently the only type of system available, which specifies a Session Manager system.

`ACTIVATE Yes|No|ON|OFF`

Default: `ACTIVATE Yes`

Specifies whether or not configuration modifications should be propagated to the remote Session Manager instance represented by the link.

If you are using a Sysplex

When operating with the `SYSPLEXGROUP` keyword specified on the `SYSTEM` statement, all Session Manager nodes using the same group name and in the same Sysplex will automatically be connected to each other, and no Session Manager `LINK` statements are required.

If Session Manager `LINK` statements are activated that connect one Session Manager node to another Session Manager node in the same Session Manager group and Sysplex, then these explicit `LINK`s will automatically be deactivated. In the same way, if a `LINK` statement has the same name as a Sysplex node then it will also be automatically deactivated.

Session Manager nodes that are operating in a `SYSPLEXGROUP` can connect to Session Manager nodes that are not part of its group via Session Manager VTAM LU0 networking, by specifying appropriate `LINK` statements.

When a non `SYSPLEXGROUP` Session Manager node connects to another non `SYSPLEXGROUP` node, via Session Manager VTAM LU0 `LINK`s, each node can access other non `SYSPLEXGROUP` nodes connected to the other's node without connecting directly to them using `LINK` statements. However, when a connection is made between a `SYSPLEXGROUP` node and a non `SYSPLEXGROUP` node then other nodes connected to the remote node are not accessible via this link.

To connect pre 1.3.15 Session Manager nodes to 1.3.15 or higher nodes, a PTF must be applied to the pre 1.3.15 nodes. Contact your local representative for details.

CHAPTER 14

RUSER statement

The RUSER statement defines users who can issue commands, using the SEND command, to a remote IBM Session Manager for z/OS system.

RUSER statements are not required for nodes within a SYSPLEXGROUP because the user's AUTH and OLACLASS, as established on the user's originating node, will be used by the remote node. For nodes in a SYSPLEXGROUP, any matching RUSER statements will override the user's AUTH and OLACLASS, therefore any redundant RUSER statements should be deleted.

Syntax and parameters

The RUSER statement takes a range of parameters:

```
RUSER userid_pattern
    NODE
    AUTH
    OLACLASS
```

userid_pattern specifies the name or generic name that is to be authorized to issue commands to a remote Session Manager system, using the Session Manager network. An asterisk (*) may be used to represent any number of any character, a plus (+) may be used to represent one of any character.

Note If OLA is being used then, to define a user who has access to several Session Manager instances that share a configuration, more than one RUSER statement is required. For details, see 'If you are using OLA' on page 153.

Parameters

NOde *nodename*

Specifies the remote Session Manager node name, as defined by the LOCALNODE parameter on the remote Session Manager SYSTEM statement. It may be up to eight characters in length.

AUth *auth-level*

Specifies the authorization level of the Session Manager commands that the user specified on the RUSER statement may issue. *auth-level* may be any value in the range 0-9. Specifying 0 prevents the user from issuing commands to the specified remote node. See the following examples.

OLACLASS *ola-class*

Specifies the OLA security class for the user specified on the RUSER statement.

If OLACLASS is not specified on the RUSER statement, the default value is the same as on the USER statement.

For an overview of OLA security classes, and the allowed values for *ola-class*, refer to the description of the OLACLASS parameter in 'Common end-user parameters' on page 30.

Examples

```
RUSER E* NODE node01 AUTH 9
```

The statement above enables any user whose name starts with E to issue commands up to authority 9 from the remote node node01.

```
RUSER EPG NODE node01 AUTH 0
```

The statement above sets the authorization level to 0 so that no commands can be issued by the user EPG from the remote node, node01. Once an RUSER is defined, it cannot be deleted in the current run. Setting the authority to 0 is the recommended method for disabling an RUSER. Any user who issues a command to a remote node and is either not defined, or does not have sufficient authority level for the command, receives the message:

```
ISZ0232E YOU ARE NOT AUTHORIZED
```

If you are using OLA

OLA can be used to maintain multiple Session Manager instances sharing a configuration.

If OLA has been set up in this way then, to define a user who has access to more than one remote Session Manager instance, more than one RUSER statement is required.

To facilitate this, specify a name other than 'RUSER' for each RUSER definition (that is, for each member of the PDS(E) allocated to the DDNAME of RUSER). For example, if user JOE requires remote access to Session Manager instances NODE01 and NODE02 then you could specify RUSER definitions like these:

Definition name	Statement
JOE_N1	`RUSER joe NODE node01 AUTH 5 OLACLASS c3`
JOE_N2	`RUSER joe NODE node02 AUTH 3 OLACLASS b9`

Note Definition (member) names are restricted to eight characters.

If you are using a Sysplex

When operating with the SYSPLEXGROUP keyword specified on the SYSTEM statement, RUSER statements are not required. RUSER statements are used to specify a remote user's AUTH and OLACLASS values. If no RUSER statement is specified and the SYSPLEXGROUP is also specified then remote users will inherit their original AUTH and OLACLASS values. However, if an RUSER or generic RUSER is found then these RUSER AUTH and OLACLASS values will be used.

For example:

A Sysplex contains two Session Manager nodes, ISM1 and ISM2, which are connected in a SYSPLEXGROUP. A user signs on to ISM1 and his AUTH and OLACLASS are either defined or provided via RACF as "8" and "US" respectively. He then issues the SEND command, either explicitly or via a system function such as the System Management Menu, specifying a node of ISM2 and a command of DLOG. When node ISM2 executes the command and there is no applicable RUSER or generic RUSER statement, the command will run with the user's original values of AUTH = 8 and OLACLASS = US.

CHAPTER 15

GROUP statement

The GROUP statement allocates a name to a group which may optionally consist of users, profiles, and terminals. Each group defines a list of ids which can be used by the BROADCAST command, the QUERY command, and by the SPYGROUP parameter.

Syntax and parameters

The GROUP statement takes a range of parameters:

```
GROUP groupname
    LUNAMES
    TERMINALS
    PROFILES
    USERS
```

groupname specifies the name of the group with which the optional list of userids, profilenames, and terminal ids are associated. The *groupname* may be any combination of up to eight alphanumeric characters. Each groupname must be defined only once.

Parameters

LUNAMES *list-of-lunames*

Each *luname* must be defined by an LU statement. Generic lunames may be specified.

TERMINALS *list-of-terminals*

Each *terminal* in the list must be defined by a TERMINAL statement. Generic terminal names may be specified.

PROFILES *list-of-profilenames*

Each *profilename* in the list should either be defined by a PROFILE statement, or may be the default profile name. Generic profilenames may be specified.

USERS *list-of-userids*

Each *userid* in the list must be defined by a USER statement. There is no limit to the number of names and the list may continue over several logical records. Separators may be spaces, equal signs or commas. Generic userids may be specified; that is, USERS M*

Note If an entry in any of the lists matches a Session Manager statement or parameter name, the entry should be enclosed in quotes, or parentheses.

CHAPTER 16

COMMAND statement

The COMMAND statement is used to allocate a security code to each of the commands and parameters available in IBM Session Manager for z/OS. By selecting an appropriate code, non privileged users can be prevented from issuing sensitive commands such as CLOSEDOWN, and from using commands such as TRACE which should only normally be issued at the request of your local support representative. Users can even be prevented from issuing a command with a specific parameter while still being able to issue the same command but with a different parameter.

For example, the EXIT parameter of the LOGOFF command may be restricted, yet any other parameter may be freely used. The SYSTEM statement may define a default authorization level. In addition, users may have a command authorization level set in the PROFILE, USER or TERMINAL statement and are only permitted to use commands and parameters with a security code numerically equal to or lower than their authorization level. When a command is issued that has a security code equal to or greater than that set by the LOGCMDAUTH parameter of the SYSTEM statement, a message is sent to the system log.

Each command and its parameters have a security code allocated by default, even if no COMMAND statement is supplied. Command parameters automatically take the same security code as the command unless a different security code is specified using the KEYWORD and KAUTH parameters. The list of these defaults appears later.

The COMMAND statement can also be used to specify scripts to run in place of a Session Manager defined command of the same name, or to create a user-defined command. See the CMDSCRIPT parameter below for further details.

Syntax and parameters

The COMMAND statement takes a range of parameters:

```
COMMAND command [AUTH auth-level]
    ACTKEY
    KEYWORD
    CMDSCRIPT
```

This specifies the name of the command *command* and the security code *auth-level* that is to be allocated to it. *auth-level* may have any value in the range 1-9. For example:

```
COMMAND STOP AUTH 2
```

Parameters

ACTKEY Yes|No

When ACTKEY=Y is specified on a command definition then if the command is invoked anywhere other than in the menu command line, the user's CMDACTIONKEY and COMMANDPRFXVAL must also be specified. If the command is invoked on the menu command line, then the user's CMDACTIONKEY and COMMANDPRFXVAL may be omitted.

When either ACTKEY is not specified or has been specified as ACTKEY=N, then CMDACTIONKEY and COMMANDPRFXVAL may be omitted.

KEYWORD *cmd-parm-1* KAUTH *auth-level-1*
[KEYWORD *cmd-parm-N* KAUTH *auth-level-N*]

If certain command parameters are to have their own unique security code, specify the parameter with *cmd-parm* and the security code using *auth-level*. KEYWORD and KAUTH are required when allocating security codes in this way, and provided that they are supplied in pairs, the maximum which may be specified on a COMMAND statement is the number of parameters pertaining to that command. For example:

```
COMMAND  LOGOFF  KEYWORD  EXI   KAUTH 4
      KEYWORD  LOG KAUTH 2
```

Any command parameters not specified, default to the security code of the command itself. As many COMMAND statements should be specified as there are commands for which the security codes are to be altered. The commands and their default security codes are shown in the security code list given at the end of this chapter.

CMDSCRIPT Yes|No|ON|OFF [SNAME *name*]

Default: CMDSCRIPT No

Specifies whether a script is to be run in place of a Session Manager defined command of the same name. If CMDSCRIPT=Y is specified, Session Manager attempts to run a script which has the same name as the command. If SNAME is provided, the specified script *name* is run.

For example:

```
COMMAND logoff CMDSCRIPT Yes SNAME fred
```

causes Session Manager to run a script 'fred' instead of actioning the logoff command.

If `CMDSCRIPT=Y` is specified, but `SNAME` is not specified, then Session Manager attempts to run a script with the same name as the command. For example:

```
COMMAND logoff CMDSCRIPT Yes
```

causes Session Manager to run a script called 'logoff' instead of actioning the command.

If the script does not exist during configuration load/update, then an error message is issued and the command is actioned in the normal way.

If `AUTH` is specified for the command, as well as `CMDSCRIPT=Y`, then the auth setting is honoured. Any `KEYWORD` settings are ignored if a script is specified.

Scripts are not restricted to Session Manager commands. Using the `COMMAND` statement it is possible to define extra user-defined commands. For example,

```
COMMAND bill CMDSCRIPT Yes
```

defines a command called Bill, which causes a script 'Bill' to run if the command is entered.

To reset a command back from actioning a script, specify `CMDSCRIPT=N`. The script is disabled and the command is actioned in the normal way.

Command scripts may contain TPSL (the Panel and Script Language) statements, `AUDITMSG`, `CALLEXIT`, `ISZCMD` and `CALL` verbs. All other script verbs are ignored. See the *Panels, Scripts and Variables* manual for further details of script verbs and TPSL statements.

Note If a `CALL` is specified which is recursive (that is, it re-enters itself) message 580 is issued to the user and to the Audit file.

Default authority list

Command	Authority	Command	Authority
ADDSESS	1	PLAYDS	5
BACKWARD	1	PLAYHEX	5
BLOCK	5	PLAYIMAGE	5
BRECEIVE	1	PULL	1
BROADCAST	9	PUPDATE	9 (see Note)
BWD	1	QACTUSER	1
CLOSEDOWN	9	QQUIT	1
CONCEAL	1	QTASK	9

Command	Authority	Command	Authority
CONFIRM	1	QUERY	1
CUTEND	1	QUIT	1
CUTSTART	1	QUSER	1
DELETE	9	RECORD	5
DELSESS	1	REMOVEUSER	9
DEMO	5	REPLAY	5
DISCONNECT	1	RESET	1
DLOG	5	RETRIEVE	1
DOWN	1	RETURN	1
DSTORE	9	REVEAL	1
DTERM	9	RIGHT	1
DUMP	9	SE	1
END	1	SECFRESH	9
FILTER	1	SEND	1
FIND	1	SME	1
FLASH	5	SPIN	9
FORCE	9	SPY	5
FORWARD	1	SPYOFF	1
FWD	1	STARTCP	9
GFS	9	STARTLINK	9
HALTSCRIPT	1	STARTSC	1
HARDCOPY	1	STOP	9
HCOPTION	1	STOPACB	9
HELP	1	STOPLINK	9
INITSC	1	STOPTCP	9
INQUIRE	9	SWITCHPLX	9
ISZECLP	9	TERMINATE	9
ISZTEST	9	TOP	1
LEFT	1	TRACE	9
LOCK	1	TRANSFER	1
LOCKTERM	1	TTPSL	1
LOGOFF		UP	1
MSG	1	UPDATE	9

Command	Authority	Command	Authority
MSGID	1	VIEW	1
NLOG	1	WINDOWS	1
NW	1		
OK	1		
PANELID	1		
PASSFREE	1		
PASTESTART	9		
PCTRANSFER	1		

Note For `PUPDATE`, the user must also have the appropriate OLA security class to update the particular definition(s).

CHAPTER 17

MESSAGE statement

The MESSAGE statement controls the destination of individual messages and can also completely respecify the text of any message. For normal use, IBM Session Manager for z/OS messages are supplied in mixed case English. The TEXT parameter, in conjunction with the LANGuage subparameter, enables messages to be defined in any language.

The TEXT parameter can be coded several times, so that the same message can be defined repeatedly for different languages. Alternatively, the MESSAGE statement can be coded several times for the same message number. This enables message text for multiple languages to be specified on separate statements; this means that when a change is made to one language, the others do not also have to be respecified.

When duplicate message definitions are found for the same language, it is the last definition that is used.

Other parameters allow the message action code suffix to be altered, and particular messages to be sent to specific users defined in an inform list. Message destination data (AUDIT, LOG, INFORM and ROUTE parameters) can be specified on separate statements, enabling destination data to be updated without having to respecify the message text.

Syntax and parameters

The MESSAGE command takes a range of parameters:

```
MESSAGE messageid
    SUFFIX
    AUDIT
    INFORM
    LOG
    DESCRIPTOR
    ROUTE
    TEXT
```

The identity of the message is provided by its four digit code, *messageid*.

Parameters

SUFfix I|E|W

The message code suffix may be modified using this parameter. Each suffix code has a predefined meaning. These are described at the beginning of 'Session Manager Messages' in the *Messages and Codes* manual.

If defining new messages (for the User Exits) the default is E. If altering an existing message, the default is the existing suffix.

AUDIT [Yes|No|ON|OFF]

The AUDIT parameter with the Yes or ON option causes the message to be directed to the Session Manager Audit file. No or OFF prevents the message appearing in the Audit file.

INFORM *users-inform-list*

Specifies a list of userids to whom the message is to be sent. The message is sent in the preferred language, which may be defined on the SYSTEM statement. No attempt is made to convert the message text to a user specified language.

LOG [Yes|No|ON|OFF]

Directs this message to the system console by specifying the Yes or ON option. The message appears in its original form in the preferred language.

The No and OFF options prevent the message appearing on the system console.

DESCriptor *descriptor-code-list*

Each *descriptor-code* has a value from 1 to 16 and corresponds to one of the z/OS action descriptor codes. Some of the codes are mutually exclusive, but Session Manager does not perform any checking. The IBM manual describing z/OS supervisor services and macro instructions gives a list of codes and their associated actions. Each code in the list may be separated by a comma, a blank, or an equals sign.

ROUTE *route-code-list*

Each *route-code* may be any value between 1-16, and equates to an z/OS message routing code. Session Manager does not check the mutual exclusivity of the codes. Reference should be made to the appropriate IBM manual describing z/OS supervisor services and macro instructions for a list of routing codes.

`TEXT` *`message-text`* [`LANGuage` *`languageid`*]

Redefines the text of any message. Variable portions may be retained provided that the original parameter variables are used.

Language Packs, which are supplied with Session Manager in library .SISZCONF, contain all the Messages (and Panels, Scripts, and Text) that have been translated.

If the text is to be defined in several different languages then the `TEXT` parameter may be specified as many times as is needed, using the `LANGuage` subparameter to denote the language. *`languageid`* may be up to two alphanumeric characters. Any `PROFILE`, `USER` or `TERMINAL`, which has a matching language id specified, will receive the message defined by the appropriate `TEXT` and `LANGuage` set.

When the `LANGuage` subparameter is omitted, the message is assumed to be defined in the preferred language. For example, if the `SYSTEM` statement specifies `PREFLANGuage=FR`, any messages which are to appear in French should be specified using the `TEXT` parameter and the `LANGuage` subparameter may be omitted. The following example illustrates:

```
SYSTEM                   PREFLANGuage=FR

MESSAGE nn1
    TEXT text-in-French
    TEXT text-in-English LANGuage=EN
    TEXT text-in-German  LANGuage=DE

MESSAGE nn2
    TEXT text-in-French
    TEXT text-in-English LANGuage=EN
    TEXT text-in-German  LANGuage=DE

PROFILE APROF            LANGuage=EN
USER OTTO                LANGuage=DE
```

For system console messages and terminal messages, where no override language is in effect, Session Manager searches for a message with `LANGuage=FR`.

For message *`nn1`* there is no `TEXT` with `LANGuage=FR`, so the text without a `LANGuage` subparameter is used. Message *`nn1`* will appear as *`text-in-French`* on the system console, and on any terminals where the `PREFLANGuage` has not been overridden. On terminals where `APROF` is used, the message will appear as *`text-in-English`*, unless the user is `OTTO`, when the message will appear as *`text-in-German`*.

For message *`nn2`*, the message appears as *`text-in-French`* on the console and on any terminal where neither `PROFILE APROF` or `USER OTTO` are in force.

Specifying substitution variables in a message

You can specify substitution variables in the text of a message by using the characters *`@nn`* or *`%nn`*.

Note In this document, the 0x7C (that is, x'7C') character is always presented as the @ sign. It may be displayed as a different character in some non-English code pages. You should enter the appropriate 0x7C character symbol for the code page you are using.

CHAPTER 18

HCPROFILE statement

The HCPROFILE statement can be used to define a hardcopy profile, containing selectable options for screen hardcopy routing, and heading and trailing line formatting. A hardcopy profile may be referred to by name from a number of other IBM Session Manager for z/OS control statements; that is, the SYSTEM, PROFILE, TERMINAL and USER statements.

Syntax and parameters

The HCPROFILE statement takes a range of parameters:

```
HCPROFILE hc_profilename
    HCOPTION
    FORMAT
    ROUTE
```

hc_profilename specifies a 1 to 8 alphanumeric character name by which this hardcopy profile is to be known. This is the name that is used on the SYSTEM and TERMINAL statements to cause a specific hardcopy profile to be available when a user first establishes contact with Session Manager. The name may also be specified on the USER and PROFILE statements to associate a particular hardcopy profile with a specific user or group of applications.

Parameters

HCOPTION *description*

Specifies a description for each set of hardcopy formatting and routing options in a hardcopy profile. Any number of hardcopy options may be specified.

The maximum possible length is 256 bytes, but it is unlikely that such a long description would need to appear on the screen. Hence, the actual maximum length is determined by the way in which the description variable, t_hcop, is defined on the panel.

The description should be enclosed in quotes if any Session Manager keywords, or spaces, appear in the string.

FORMAT *hc_formatname*

Specifies the name of the set of heading and trailing lines to be printed before and after the screen image. The lines are defined with the HCFORMAT statement and therefore the name supplied for the *hc_formatname* **must** match a name given on an HCFORMAT statement.

When this subparameter is omitted, Session Manager generates a set of default headers and trailers, as described under the HCFORMAT statement.

ROUTE *hc_routename*

Specifies the name of a route definition. Print routes are defined using the HCROUTE statement, and therefore the route name **must** match a name provided on an HCROUTE statement.

When this subparameter is omitted, the default print route is dependent on the operating system:

A DD name of the form DD*nn* is generated by z/OS for the SYSOUT file. This file will have a class equivalent to the JOB MSGCLASS, and a DEST parameter of LOCAL.

CHAPTER 19

HCFORMAT statement

The HCFORMAT statement defines a set of heading and trailing lines for screen hardcopies taken with the IBM Session Manager for z/OS Hardcopy facility. The lines so defined are associated with one or more hardcopy profiles by means of the FORMAT subparameter of the HCPROFILE statement.

A more detailed description of the Hardcopy facility may be found in 'Using the Hardcopy Facility' in the *User and Administrator* manual.

HCFORMAT parameters

The HCFORMAT statement takes a range of parameters:

```
HCFORMAT hc_formatname
    HEADER
    TRAILER
```

hc_formatname specifies a name to identify this particular set of screen hardcopy heading and trailing lines. From one to eight alphanumeric characters are acceptable. The FORMAT subparameter of an HCOPTION parameter of an HCPROFILE statement specifies a *hc_formatname*. Screen hardcopies taken when that HCOPTION is selected have the heading and trailing lines automatically printed before and after the screen copy.

Any number of heading and trailing lines may be specified for each HCFORMAT statement.

Parameters

HEADER *header-text*

Specifies a single heading line. The number of HEADER parameters determines the number of heading lines that are printed. If no HEADER parameters are specified, Session Manager generates a set of default heading lines as follows:

```
HEADER '-'
HEADER 'HARDCOPY AT &t_time ON &t_date'
HEADER 'USER &t_user ON &t_termtyp &t_termid '-
'LOGGED ON TO &s_appl ACB &s_acb'
HEADER '-'
```

Each line may contain literal text and optionally substitution variables. A list of the acceptable variables may be found in 'Session Manager Variables' in the *Panels, Scripts and Variables* manual. The chapter 'Using the Hardcopy Facility' in the *User and Administrator* manual contains examples of their use. When the length of the actual substituted variable is less than the maximum allowed for that field, the trailing spaces are not included in the output line.

When the heading line is to contain an identical character in all print positions, this may be specified using a single character enclosed in quotes. For example, a blank line is defined by specifying ' ', and a line of dashes by '-'. If just one character is to comprise the whole line, the character should have a space appended to it, for example, '* ' causes an asterisk to be placed in the middle of the line.

The resultant header string is centered in the line, and if the string exceeds the width of the screen from which the hardcopy was taken, truncation occurs at the end of the line.

TRAILER *trailer-text*

Specifies a single line that is to be printed following the hardcopy screen image. Multiple lines are printed when more than one TRAILER parameter is specified.

When this parameter is omitted, Session Manager generates a single line of dashes. The HEADER and TRAILER parameters are independent of each other. Either one or both may be specified as required.

The definition of a trailing line, and the rules for substitution and truncation, are identical to that of the heading line which is described previously.

CHAPTER 20

HCROUTE statement

The HCROUTE statement is used to specify a print route for screen hardcopies taken using the IBM Session Manager for z/OS Hardcopy facility.

The various routes are selectable by hardcopy option, since each HCOPTION parameter of an HCPROFILE statement refers to a particular HCROUTE statement.

Variable substitution may be used in some of the HCROUTE parameters and are identified in this chapter by the 'Substitution' heading in each parameter description. The valid variables are described in 'Session Manager Variables' in the *Panels, Scripts and Variables* manual.

An overall description of the way in which the Hardcopy facility may be used can be found in 'The Hardcopy Facility' contained in the *User and Administrator* manual.

Syntax and parameters

The HCROUTE statement takes a range of parameters:

```
HCROUTE hc_routename
    CLASS
    COPIES
    DESTINATION
    EXTWTR
    FCB
    FLASH
    FORM
    HIGHLIGHT
    HOLD
    NAME
    NODE
    OUTPUT
```

hc_routename specifies a name by which this print route is known. Between one and eight alphanumeric characters may be specified. This name may be supplied on the ROUTE subparameter of an HCOPTION parameter of an HCPROFILE statement, to cause screen hardcopies to be routed along a particular, and optionally selectable, path.

Parameters

CLASS *class-code*

Substitution: YES (only the first character of the result is taken)

The one character class code, which is normally specified on the DD 'SYSOUT=' operand, may be overridden using the CLASS parameter. When this is not specified, the JOB MSGCLASS code is used in its place.

Variable substitution is restricted to one of the user definable variables and only the first character of the result is taken.

COPIES *copies*

Substitution: NO

Specifies the number of copies to be produced. *copies* may be any value in the range 1-255. The default is 1.

DESTination *destination-code*

Substitution: YES

Specifies the destination code for the hardcopy output. For JES2 and JES3, the routing specification, usually given by the DD 'DEST=' operand, is generated from the information supplied by this optional parameter. When specified, the parameter should conform to the required standards for the 'DEST=' DD statement parameter.

When this parameter is omitted, 'DEST=LOCAL' is used for the appropriate DD statement.

For TSO, this parameter gives the userid to be inserted in the DEST=(*node*,*userid*) parameter. The node can be supplied by the NODE parameter of HCROUTE statement.

`EXTWTR` *`name`*

Substitution: YES

The name of an external writer may be specified using this parameter.

`FCB` *`fcb-name`*

Substitution: NO

Specifies an FCB name. It may be from one to four characters.

`FLASH` *`overlay`* `[`*`count`*`]`

Substitution: NO

The first four characters give the overlay name. These may optionally be followed by a flash count field, *`count`*, which may be any value in the range 0-255.

`FORM` *`forms-name`*

Substitution: YES (the resultant field must be valid both in format and content for the operating system in use.

Specifies the forms name to be associated with this hardcopy route.

`HIGHlight [Yes|No|ON|OFF]`

Substitution: NO

Specifies whether or not overprinting is used to indicate highlit fields. The default is `ON`. `OFF` should be specified when the printer does not have the capability to overprint, otherwise results may be unpredictable.

`HOLD [Yes|No|ON|OFF]`

Substitution: NO

Specifies whether or not list output should be printed when it is placed in a print queue. `HOLD=Y` causes each entry to be held and therefore an appropriate operating system command must be issued in order to release the entry for printing.

`HOLD=N`, which is the default, results in a disposition such that the entry is ready to print.

`NAME` *`override-name`*

Substitution: YES

This parameter may be used in any operating system to modify the relevant fields. The maximum length of the coded operand is 16 characters, but truncation occurs if the resultant string exceeds eight characters.

For each hardcopy, a SYSOUT file is allocated, and the `NAME` parameter may be used to override the DD name that z/OS would otherwise allocate.

Note that the name used should not match any that are specified in the startup jobstream, since Session Manager generates a DD statement.

`NODE` *`nodename`*

Substitution: YES

Specifies a one to eight character node name. For TSO the node name is placed in the DEST=(node,userid) parameter.

`OUTPUT` *`output-JCL-statement-name`*

Substitution: YES (the first eight characters of the result are taken)

Specifies the 1 to 8 character name of a z/OS JCL `OUTPUT` statement which has been included in the start-up job stream for Session Manager. The z/OS `OUTPUT` statement has many different parameters for specifying output options. In this particular context, the most useful ones are, FCB, UCS and FORMS. FCB and FORMS may also be specified with the appropriate parameters of the `HCROUTE` statement.

CHAPTER 21

AUDITROUTE statement

The AUDITROUTE statement is used to specify a print route for the IBM Session Manager for z/OS Audit file. Only one AUDITROUTE may be specified in the configuration. The print parameters for AUDITROUTE are similar to those for the HCROUTE statement and reference should therefore be made to the section describing the HCROUTE statement on page 172. The differences are described below:

- There is no equivalent to the hardcopy *routename* for the Audit file since the auditroute is unique.
- There is no HIGHLIGHT parameter on this statement.
- Variable substitution may not give the desired results since few variables are available for AUDITROUTE parameters. It is recommended that variable substitution is avoided, or used with caution.

CHAPTER 22

TRACEROUTE statement

The `TRACEROUTE` statement is used to specify a print route for IBM Session Manager for z/OS traces. Only one `TRACEROUTE` may be specified in the configuration. The parameters for `TRACEROUTE` are similar to those for the `HCROUTE` statement and reference should therefore be made to the section describing the `HCROUTE` statement on page 172. The differences are described below:

- There is no equivalent to the hardcopy *`routename`* for the traces since traceroute is unique.
- There is no `HIGHLIGHT` parameter on this statement.
- Variable substitution may not give the desired results since few variables are available for `TRACEROUTE` parameters. It is recommended that variable substitution is avoided, or used with caution.

CHAPTER 23

TRANSTABLE statement

The TRANSTABLE statement modifies the translation tables used by IBM Session Manager for z/OS when reading configuration statements, processing commands, or displaying Session Manager panels. Any character in the data, but which is not specified on the TRANSTABLE statement, is processed using the default definition. A translation table definition can be referenced by the TRANSTAB parameter of the SYSTEM, PROFILE, USER and TERMINAL statements.

Syntax and parameters

The TRANSTABLE statement takes a range of parameters:

```
TRANSTABLE transtable
    INPUT|OUTPUT
    SET
```

transtable specifies the table name of a unique one to eight character name assigned to the translation table. It should be the same as the table name used on the TRANSTAB parameter on the PROFILE, USER or TERMINAL statement.

If a table has been created and is no longer required, its settings can be changed back to the defaults by specifying its name followed by either INPUT or OUTPUT, and omitting the SET parameter.

Product default: DEFAULT

INPUT|OUTPUT

Default: INPUT

Causes all Session Manager commands and configuration statements to be parsed using the named table. The purpose of the translate table when reading the configuration statements is to establish valid non-blank characters in the statements so that only these are processed. Other characters may be processed in quotes. INPUT is the default option.

The OUTPUT option translates Session Manager panels using the named table. Its purpose is to prevent invalid characters being sent to terminals.

SET *offset list_of_hex_chars* [SET *offset list_of_hex_chars*] ...

Specifies a list of hexadecimal characters which are used to modify the translation table. When SET is not specified, the translation table is reset to its default values. Multiple SETs can be specified on the same TRANSTABLE statement.

The *offset* is the position in the table of the first hexadecimal character; it must consist of two hexadecimal digits.

The *list-of-hex-chars* specifies the characters which replace the default characters. The list must be an even number of hexadecimal digits.

To suppress a character for INPUT, specify X'00'. To suppress a character for OUTPUT specify X'4B', to make the invalid character appear as a '.'

The default INPUT and OUTPUT translation tables are reproduced here for reference.

Default INPUT table

For Session Manager 1.2.10 and higher, there are DBCS characters in this table.

```
        0 1 2 3 4 5 6 7 8 9 A B C D E F
SET 00 00000000000000000000000000000E0F

        0 1 2 3 4 5 6 7 8 9 A B C D E F
SET 10 00000000000000000000000000000000

        0 1 2 3 4 5 6 7 8 9 A B C D E F
SET 20 00000000000000000000000000000000

        0 1 2 3 4 5 6 7 8 9 A B C D E F
SET 30 00000000000000000000000000000000

        0 1 2 3 4 5 6 7 8 9 A B C D E F
SET 40 00414243444546474849 4A4B4C4D4E4F

        0 1 2 3 4 5 6 7 8 9 A B C D E F
SET 50 50515253545556575859 5A5B5C5D5E5F

        0 1 2 3 4 5 6 7 8 9 A B C D E F
SET 60 60616263646566676869 6A6B6C6D6E6F

        0 1 2 3 4 5 6 7 8 9 A B C D E F
SET 70 70717273747576777879 7A7B7C7D7E7F

        0 1 2 3 4 5 6 7 8 9 A B C D E F
SET 80 80818283848586878889 8A8B8C8D8E8F

        0 1 2 3 4 5 6 7 8 9 A B C D E F
SET 90 90919293949596979899 9A9B9C9D9E9F

        0 1 2 3 4 5 6 7 8 9 A B C D E F
SET A0 A0A1A2A3A0O5A6A7A8A9AAABACADAEAF

        0 1 2 3 4 5 6 7 8 9 A B C D E F
SET B0 B0B1B2B3B4B5B6B7B8B9BABBBCBDBEBF

        0 1 2 3 4 5 6 7 8 9 A B C D E F
SET C0 C0C1C2C3C4C5C6C7C8C9CACBCCCDCECF

        0 1 2 3 4 5 6 7 8 9 A B C D E F
SET D0 D0D1D2D3D4D5D6D7D8D9DADBDCDDDEDF

        0 1 2 3 4 5 6 7 8 9 A B C D E F
SET E0 E0E1E2E3E4E5E6E7E8E9EAEBECEDEEEF

        0 1 2 3 4 5 6 7 8 9 A B C D E F
SET F0 F0F1F2F3F4F5F6F7F8F9000000000000
```

Default OUTPUT table

```
         0 1 2 3 4 5 6 7 8 9 A B C D E F
SET 00 004B4B4B4B4B4B4B4B4B4B4B4B4B4B4B

        0 1 2 3 4 5 6 7 8 9 A B C D E F
SET 10 4B4B4B134B4B4B4B4B4B4B4B4B4B4B4B

        0 1 2 3 4 5 6 7 8 9 A B C D E F
SET 20 4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B

        0 1 2 3 4 5 6 7 8 9 A B C D E F
SET 30 4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B

        0 1 2 3 4 5 6 7 8 9 A B C D E F
SET 40 404B4B4B4B4B4B4B4B4B4A4B4C4D4E4F

        0 1 2 3 4 5 6 7 8 9 A B C D E F
SET 50 504B4B4B4B4B4B4B4B4B5A5B5C5D5E5F

        0 1 2 3 4 5 6 7 8 9 A B C D E F
SET 60 60614B4B4B4B4B4B4B4B6A6B6C6D6E6F

        0 1 2 3 4 5 6 7 8 9 A B C D E F
SET 70 4B4B4B4B4B4B4B4B4B797A7B7C7D7E7F

        0 1 2 3 4 5 6 7 8 9 A B C D E F
SET 80 4B818283848586878889 4B4B4B4B4B4B

        0 1 2 3 4 5 6 7 8 9 A B C D E F
SET 90 4B919293949596979899 9A9B9C9D9E9F

        0 1 2 3 4 5 6 7 8 9 A B C D E F
SET A0 4BA1A2A3A4B5A6A7A8A94B4B4B4B4B4B

        0 1 2 3 4 5 6 7 8 9 A B C D E F
SET B0 4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B

        0 1 2 3 4 5 6 7 8 9 A B C D E F
SET C0 C0C1C2C3C4C5C6C7C8C94B4B4B4B4B4B

        0 1 2 3 4 5 6 7 8 9 A B C D E F
SET D0 4BD1D2D3D4D5D6D7D8D94B4B4B4B4B4B

        0 1 2 3 4 5 6 7 8 9 A B C D E F
SET E0 E04BE2E3E4E5E6E7E8E94B4B4B4B4B4B

        0 1 2 3 4 5 6 7 8 9 A B C D E F
SET F0 F0F1F2F3F4F5F6F7F8F94B4B4B4B4B4B
```

CHAPTER 24

PATCH, PATCHSU, APPLYSU, REMOVESU statements

The `PATCH`, `PATCHSU`, `APPLYSU` and `REMOVESU` statements are all special configuration control statements. These should only be used at the request of your local support representative.

PATCH statement

The PATCH statement causes the loaded Session Manager code to be altered. No change is made to library members. This means that each change must be reapplied each time Session Manager is initiated, but when all the changes are contained in one source member, it becomes easy to maintain the changes and to see at a glance which changes are currently in effect. It also means that PTFs can be applied while Session Manager is active.

The PATCH statement syntax is:

```
PATCH patch-num
    CHECK check-code
    NAME  name1
        VER   hex-offset hex-data
        REP   hex-offset hex-data
        ...
        VER   hex-offset hex-data
        REP   hex-offset hex-data
    NAME  name2
        VER   hex-offset hex-data
        REP   hex-offset hex-data
        ...
        VER   hex-offset hex-data
        REP   hex-offset hex-data
    ...
```

All variable information is supplied by your local support representative. The *patch_num* is the patch number. The *name* is usually a CSECT name or an entry-point name, and the *hex-offset* is the offset from the location of that name.

The subcommands ISZMAP and ZEDMAP of the DSTORE command show lists of names and their addresses in the Session Manager system.

The *hex-data* can be specified as a continuous string, or can be supplied in groups of four hex-digits separated by a space or a comma, to a maximum of 32 hex-digits. For example, these variants are all acceptable:

```
VER 03AD 47F0,4326
VER 03AD 47F0 4326
VER 03AD 47F04326
```

Multiple PATCH statements can be specified, each may contain a number of NAME parameters and VER/REP pairs. However, only one CHECK parameter may be specified for a PATCH statement. This parameter is used to ensure that no keying errors have occurred in specifying the PTF. The check-code is calculated from the values of the hex-offsets and hex-data in all the VER and REP parameters, and treated as one continuous string. For this reason it is important that the order of the parameters in any PATCH statement is **not** changed.

It may be prudent to maintain two source members for PTFs. One might contain the changes which need to be applied each time Session Manager is initiated, and the other might be used to contain a new PTF to be applied while Session Manager is active. The first member would require a COPY statement in the configuration file to cause the PATCH statements to be applied at initiation. The second would need an UPDATE command to be issued, for example:

```
UPD MEM NEWPATCH
```

where NEWPATCH would be the source member name. The patch could then be copied into the first member so that it is applied at each Session Manager initiation.

Backing-out a PTF

It is very simple to back-out at Session Manager initiation since the appropriate PATCH statement can be removed, or commented out, from the source member so that the change is not applied.

If a change needs to be removed while Session Manager is active, the VER parameters should be changed to REP, and REP to VER in the appropriate PATCH statement. The order of the parameters must **not** be changed, otherwise the check processing will fail. It is quite acceptable for REP to appear before VER. Once the statement has been altered, an UPDATE command can be issued to back-out the PTF.

Applying a PTF when using OLA

If OLA is used to maintain the Session Manager configuration then a different procedure is used to apply/back-out PTF(s).

Applying a PTF

To apply PTF(s), complete these steps:

1. Edit the member ISZCINIT in the System PDS(E) allocated to the DDNAME of CONFIG using the ISPF editor (or its equivalent) and copy and paste the contents of the PTF(s) into the bottom of this member.

   Note Ensure that the PTF(s) appear in *ascending* sequence in this member.

2. If you do not want to activate the PTF(s) immediately then skip the rest of the steps and schedule a shutdown and restart at your convenience.

3. If you want to activate the PTF(s) immediately online then issue this command:

   ```
   PUPDATE CONFIG ISZCINIT
   ```

   Note An OLA security 'class default' setting of at least MODIFY is required. Shipped classes that have a 'class default' of ALL include:

   AD – Administrator
   BT – Batch Administrator
   IM – Implementor
   LA – Local Administrator

4. Alternatively, if you want to activate the PTF(s) immediately by running a batch job then complete the following steps.

Note The batch commands referred to below could be wrapped into an authorized script (see the *Panels, Scripts and Variables* manual).

a Edit member ACTCINIT in the System PDS(E) allocated to the DDNAME of CONFIG using the ISPF editor (or equivalent) and insert this line:

```
ACTIVATE DDN(CONFIG) MBR(ISZCINIT)
```

b Edit sample job ISZBAJOB which is supplied in library SISZCONF, and change the supplied command from:

```
ISZSRCH STATEMENT(USER)
```

to

```
* RUN THE ACTIVATE
ISZACTV
 INPUT(DDN(CONFIG ACTCINIT)
```

c Tailor the rest of the JCL for the job to your Installation's requirements.

d Submit the sample job and check that it completes successfully (check the output carefully for errors and correct as necessary). If any problems are encountered, please contact your local Support Representative.

For more information on the batch commands referred to above, see the *Online and Batch Administration* manual.

Backing-out a PTF

The procedure for backing-out a PTF depends on whether or not you activated the PTF(s) when you followed the steps detailed in 'Applying a PTF' on page 185.

If you did not active the PTF(s)

If you did not activate the PTF(s) then to back-out the changes simply remove the PTF(s) from the bottom of member ISZCINIT in the System PDS(E) allocated to the DDNAME of CONFIG using the ISPF editor (or equivalent).

If you activated the PTF(s)

If you activated the PTF(s) then to back-out the changes complete these steps:

1. Edit member ISZCINIT in the System PDS(E) allocated to the DDNAME of CONFIG using the ISPF editor (or equivalent) and reverse the action of each pair of VER and REP entries in the PTF(s) at the bottom of this member.

   For example, if the PTF entries at the bottom of member ISZCINIT are:

```
PATCH 1887
    CHECK 7E6DA3EE
    NAME ISZCVD
        VER 05D8 12FF,58D0,D004,50F0,D010,98EC,D00C,07FE
        REP 05D8 182F,58F0,BA14,47F0,F17C,0000,0000,0000
    NAME ISZZZA
        VER 417C F17C,F17E
        REP 417C 0700,BF1F
        VER 4180 F180,F182,F184,F186,F188,F18A,F18C,F18E
        REP 4180 8044,4780,F194,58F0,B9FC,58F0,FA38,41F0
        VER 4190 F190,F192,F194,F196,F198,F19A,F19C,F19E
        REP 4190 F2E4,05EF,18F2,12FF,58D0,D004,50F0,D010
        VER 41A0 F1A0,F1A2,F1A4
        REP 41A0 98EC,D00C,07FE
```

then change them so that they appear as follows (note that the order of the `VER` and `REP` entries in each pair is not significant):

```
PATCH 1887
    CHECK 7E6DA3EE
    NAME ISZCVD
        REP 05D8 12FF,58D0,D004,50F0,D010,98EC,D00C,07FE
        VER 05D8 182F,58F0,BA14,47F0,F17C,0000,0000,0000
    NAME ISZZZA
        REP 417C F17C,F17E
        VER 417C 0700,BF1F
        REP 4180 F180,F182,F184,F186,F188,F18A,F18C,F18E
        VER 4180 8044,4780,F194,58F0,B9FC,58F0,FA38,41F0
        REP 4190 F190,F192,F194,F196,F198,F19A,F19C,F19E
        VER 4190 F2E4,05EF,18F2,12FF,58D0,D004,50F0,D010
        REP 41A0 F1A0,F1A2,F1A4
        VER 41A0 98EC,D00C,07FE
```

To back out the PTF(s):

2 If you do not want to back-out the PTF(s) immediately then skip the rest of the steps and schedule a shutdown and restart at your convenience.

3 If you want to back-out the PTF(s) immediately online then issue this command:

```
PUPDATE CONFIG ISZCINIT
```

Note An OLA security 'class default' setting of at least `MODIFY` is required. Shipped classes that have a 'class default' of `ALL` include:

`AD` – Administrator
`BT` – Batch Administrator
`IM` – Implementor
`LA` – Local Administrator

4 Alternatively, if you want to back-out the PTF(s) immediately by running a batch job then complete the following steps.

Note The batch commands referred to below could be wrapped into an authorized script (see the *Panels, Scripts and Variables* manual).

a Edit member ACTCINIT in the System PDS(E) allocated to the `DDNAME` of `CONFIG` using the ISPF editor (or equivalent) and ensure this line is present:

```
ACTIVATE DDN(CONFIG) MBR(ISZCINIT)
```

b Edit sample job ISZBAJOB which is supplied in library SISZCONF, and ensure that the supplied command has been changed from:

```
ISZSRCH STATEMENT(USER)
```

to

```
* RUN THE ACTIVATE
ISZACTV
 INPUT(DDN(CONFIG ACTCINIT)
```

c Tailor the rest of the JCL for the job to your Installation's requirements.

d Submit the sample job and check that it completes successfully (check the output carefully for errors and correct as necessary). If any problems are encountered, please contact your local Support Representative.

For more information on the batch commands referred to above, see the *Online and Batch Administration* manual.

PATCHSU statement

The PATCHSU statement enables changes to be applied to Selectable Units. The syntax of this statement is:

```
PATCHSU su_num patch-num
    CHECK check-code
    NAME  name1
        VER   hex-offset hex-data
        REP   hex-offset hex-data
        ...
        VER   hex-offset hex-data
        REP   hex-offset hex-data
    NAME  name2
        VER   hex-offset hex-data
        REP   hex-offset hex-data
        ...
        VER   hex-offset hex-data
        REP   hex-offset hex-data
    ....
```

All variable information is supplied by your local support representative. The *su_num* is the SU number (range 1-999); all other variables are described in 'PATCH statement' on page 184, which you should also refer to for details of how to use the PATCHSU statement.

See also 'Selectable Units' on page 195, the 'INSTALLSU statement' on page 81, the 'APPLYSU statement' on page 190 and the 'REMOVESU statement' on page 191.

APPLYSU statement

The `APPLYSU` statement is a special configuration control statement, which can be used to alter the executable code by applying one or more selectable units. These contain either a programming enhancement or Session Manager maintenance.

The `APPLYSU` statement syntax is:

`APPLYSU` *`su_num`*

where:

`su_num`

Is the SU number (range 1-999) to be applied.

See also 'Selectable Units' on page 195, the 'INSTALLSU statement' on page 81, the 'PATCHSU statement' on page 189 and the 'REMOVESU statement' on page 191.

The `APPLYSU` statement is a special configuration control statement, which can be used to alter the executable code by applying one or more selectable units. These contain either a programming enhancement or Session Manager maintenance. The statement should only be used at the request of your local support representative.

REMOVESU statement

The `REMOVESU` statement is a special configuration control statement, which can be used to alter the executable code by removing one or more selectable units. These contain either a programming enhancement or Session Manager maintenance.

The `REMOVESU` statement syntax is:

`REMOVESU` *su_num*

where:

su_num

Is the SU number (range 1-999) to be removed.

See also 'Selectable Units' on page 195, the 'INSTALLSU statement' on page 81, the 'PATCHSU statement' on page 189 and the 'APPLYSU statement' on page 190.

CHAPTER 25

DELETE statement

The DELETE statement is used to delete in-storage definitions from a running IBM Session Manager for z/OS system. Note that the UPDATE facility does not totally allow in-storage deletion since it requires that Session Manager be recycled in order to complete the updates.

The DELETE statement is designed to operate with OLA and is generated automatically by OLA scripts. The generated DELETE statements are then activated via a PUPDATE command to clear the in-storage references.

However, users of Classic systems can create configuration members by explicitly coding their own DELETE statements then performing an UPDATE command. Note that this will only clear the item from storage and will not affect the Classic configuration datasets. When the delete is performed from an OLA menu, the member of the corresponding PDSE is removed as well.

Syntax and parameters

The syntax of the DELETE statement is:

```
DELETE statement-type statement-value
    RUNODE ruser-node-name
```

statement-type will be one of the following:

USER

TERMINAL (this also covers the deletion of LUs)

PROFILE

GROUP

APPL

LINK

RUSER

Note These are the only statement-types currently supported.

statement-value specifies the name of the specific statement-type entry to be deleted.

Parameters

RUNODE *ruser-node-name*

The RUNODE parameter only applies to the RUSER statement type where it is necessary to specify the particular NODE the RUSER is associated with.

Examples

```
DELETE USER BILL
```

will cause the in-storage definitions for user BILL to be deleted. Where relevant, generic names can be deleted as well. For example:

```
DELETE USER DEVU*
```

will delete the in-storage definition DEVU*.

```
DELETE RUSER ISZSMGR RUNODE BATCH
```

will delete the in-storage definition of the RUSER ISZSMGR associated with the NODE BATCH.

CHAPTER 26

Selectable Units

Each Selectable Unit (SU) contains either a programming enhancement, or IBM Session Manager for z/OS maintenance code. Issuing the `QUERY ISZSMGR` command displays a description of each SU which has been installed.

Applying and removing SUs

Each SU delivers not only a number of new functions, but also a few fixes. Therefore, when you upgrade Session Manager the SUs shipped with the new release must be installed (see 'Loading and applying SUs' below).

Applying SUs at startup

A set of selectable units are supplied in a predefined SU table and contain code relating to the latest release of Session Manager. This table contains the numbers of the SUs that are to be loaded and applied automatically at Session Manager startup, by the use of the `INSTALLSU` configuration statement. The `INSTALLSU` statement should be the first statement after the initial `OPTION` statement (if any) in the configuration file. For details, see 'INSTALLSU statement' on page 81.

If this statement is omitted from the configuration file, and the table contains at least one SU, then an error message is issued. However, the configuration process is **not** aborted and startup continues.

Each selectable unit consists of a load module or phase and optionally some associated configuration members. The `ISZ£READ` file shipped with the latest version of Session Manager will contain a description of each SU to be installed. This file also contains any special processing required, such as making the associated supplied configuration members available to Session Manager. Each SU supplied has a corresponding `ISZCS`*nnn* file (where *nnn* is the SU number).

Loading and applying SUs

To load and apply the SUs automatically at Session Manager startup, use the `INSTALLSU` configuration statement (which is included in a standard configuration). For more information, see 'INSTALLSU statement' on page 81.

Applying individual SUs

There may be instances where your local support representative requests you to add another SU or remove an SU after the initial Session Manager startup. To add an SU, make available to the Session Manager machine, address space, or partition, the SU load module and any supplied configuration members. A member will normally be supplied containing an `APPLYSU` statement, which is used to apply a new selectable unit. Issue an `UPDATE` command for this member.

Various diagnostic messages are issued during the apply process. If the apply fails, the reason for the failure should be identifiable from the messages. The `APPLYSU` statement is not rejected if the SU has already been applied, so for example, if the SU was applied at startup and the `UPDATE ALL` command is issued, the command does not fail just because the SU has already been applied.

Details of this statement can be found on page 190.

Displaying a description of SUs installed

To display a description of each SU which has been installed, use the `QUERY ISZSMGR` command. For more information, see 'QUERY user command' on page 234.

CHAPTER 27

Using the Update Facility

The IBM Session Manager for z/OS Update Facility provides Installations with the option of amending the Session Manager environment while it is active.

This facility enables most of the configuration control statements to be altered. It is possible to change current definitions for terminals, users, profiles, scripts and panels. Any definition with a name which matches an existing definition, totally replaces the original definition. Completely new definitions may also be included. A new User exit may be loaded, or an existing one removed.

Amendments are made to source members using the normal editing facilities available in the Installation's environment. The changes are then put into effect by means of the Session Manager `UPDATE` command.

As performing updates affects Session Manager performance, it is advisable to update only those parts of the configuration that have changed, rather than the whole configuration.

Note If your configuration is OLA-enabled then do not use the `UPDATE` command. Instead, to update the Session Manager configuration, use the OLA menus, lists and attribute display panels. Also, the `PUPDATE` command (see page 259) can be used if necessary.

However, the `UPDATE` command *is* still required to update the user exit and exit scripts, and to perform an `UPDATE ALL`.

Amendments may be made to the original source members used at start up, or may be included in entirely new members, provided that the new members are available to the currently active Session Manager. They must be in a partitioned dataset associated with the DDNAME 'CONFIG'.

Note `FREE=CLOSE` should not be specified on the CONFIG partitioned dataset DD statement, as this will cause the configuration file to be unavailable after initialization. The Update facility will therefore be unable to read members from the partitioned dataset.

Details of the `UPDATE` command syntax are given on page 271.

Temporary changes may be made to the Session Manager environment using new source members. To make these alterations permanent, either the `OPTION` statement must be amended to identify the new names, or the changes must be made to the original source members. This provides a useful method of testing changes and detecting potential errors, before a change is put into production.

Update options and parameters

Parameters on the UPDATE command give several options. Update processing may be specified for just the configuration statements member, a copied member, or the User exit. These options are specified by the CONFIG, the MEMBER, and the EXIT parameters respectively. The section 'Privileged user commands' on page 245 gives a complete description of the UPDATE command and its parameters.

No other processing takes place during the update, however, the overhead can be minimized by selecting only those parts which need to be changed or added. The configuration source may be divided between several members and included in the main configuration file using the COPY statement. For example, the COPY statements might be:

```
Copy system
Copy users
Copy terms
Copy profs
Copy screens
```

If the member named 'USERS' is amended, the changes may be incorporated in the current environment by issuing the command UPD MEM USERS therefore avoiding the need to process the whole configuration file.

A new User exit may be loaded by specifying its name on the EXIT parameter, and an existing one can be removed by specifying UPD EXIT 0, where 0 is zero, not the letter O.

z/OS System Symbols designated by the variable names GC_MVS_*symbolname* can now be modified dynamically as required using the GCMVS parameter.

Update implications

When new source members are specified, either as a new suffix for the configuration control statements source, or as an entirely new member, the original definitions continue to exist in the Session Manager environment.

When using the Update facility, the only definitions that are removed are those which are replaced by new definitions with a duplicate name. Statement definitions for which there is no matching name are included as entirely new statements. Any errors cause the new or replacement statement to be ignored and, if it is a duplicate, the original definition is retained. Statements in error are printed with an appropriate error message.

The original definitions are not deleted from the environment while they are currently in use, but new definitions become current the next time they are used subsequent to the `UPDATE` command being issued. This means that two versions of a particular definition may exist for a time. If there are multiple versions of a definition, then the old version is only deleted when there are no users using it. If there is only one version of a definition, unless the `DELETE` function is used, this will not be deleted from program storage until Session Manager is recycled.

For example, a new profile comes into effect when a user signs on, but any users signed on with active details at the time of the update continue to use the old copy of the profile. For these users, the new profile is put into effect as soon as they have no active sessions. User modified screen variables are merged with the new profile where appropriate.

For details of how to delete definitions from an active system, see 'DELETE statement' on page 193.

When amendments become effective

Changes to the various configuration control statements and parameters become effective at different times, depending on the nature of the alteration. The following tables identify the different points at which each change takes effect.

Control Statement	Parameter	When Changes Become Effective
SYSTEM		
	`ACB`	When system is recycled
	`ACTIVESESSIONS`	At next session start
	`ADDSID`	At next session start
	`AFFINITY`	At next signon
	`ALARM`	At next session start
	`ALLOWESCAPE`	At next session start
	`ALLUSERS`	At next signon
	`APPLSEL`	At next session start
	`ATTR`	Next update of script/panel

Control Statement	Parameter	When Changes Become Effective
	AUDITOGDG	When system is recycled
	AUTH	Immediately
	AUTOSCRIPT	At next session start
	AUTOSELECT	At next signon or reconnect
	AUTOSEQ	At next signon
	AUTOSEQ A|I|E	Immediately
	AUTOSTART	At next session start or reconnect
	BACKWARD	At next signon
	BINDTIMEOUT	Immediately
	BLANKSCRIPT	At next session start
	BRDVAR	Inactive session: effective immediately. Active session: at next session start.
	BRECEIVE	At next signon
	CLOSEACBINACT	Immediately
	CLOSEDISC	At next session start
	(Note: In OLA it is displayed as TYPE)	
	CLOSELOGOFF	At next session start
	(Note: In OLA it is displayed as TYPE)	
	CMDACTIONKEY	At next signon
	COMMANDPRFXVAL	At next signon
	COMPRESS	At next session start
	CONCEAL	Immediately
	CONDLOGOFF	Immediately
	CURESC	At next signon
	CUT	At next signon
	CV64	When ACB is next opened
	DAPPLCHECK	Immediately
	DEFAPPL	Immediately
	DEFMENU	Immediately
	DEFPROFILE	Immediately
	DEMO	At next signon

Control Statement	Parameter	When Changes Become Effective
	DESCRIPTION	Inactive session: effective immediately. Active session: at next session start.
	DISCACTIVE	At next signon
	DLOGLIMIT	Immediately
	DMY|MDY|YMD|YDM	Immediately
	(Note: In OLA they are displayed as DATEFORMAT)	
	DOMAX	Next load of definition
	DOUBLESC	At next signon
	DROP_SESSION	At next signon
	DSESSRANGE	At next signon
	DUMPGDG	Immediately
	ENDSCRIPT	At next session start
	ENVIRONSCRIPT	At next session start
	ERTIMEOUT	At next signon
	ESCAPE	At next signon
	ESCAPE CURSOR	Immediately
	ESMOLAGROUP	At next signon
	EUTIMEOUT	At next signon
	EXITWALEN	When system is recycled
	FORWARD	At next signon
	GENERICACB	When system is recycled
	GENRESNAME	When system is recycled
	HARDENUSER	Immediately
	HCPROF	Immediately
	HCREQUEST	At next signon
	HCREQUEST CURSOR	Immediately
	HIDE	At next session start
	IDLEDISC	At next terminal enter
	IDLELOCK	At next terminal enter
	IDLELOGOFF	At next terminal enter
	ILU	At next session start
	IMSCONVERT	Immediately

Control Statement	Parameter	When Changes Become Effective
	IMSCONVERTC	Immediately
	INITSCRIPT	At next session start
	INITIAL_CMD	When system is recycled
	INPUTEXIT	When system is recycled
	INQINTERVAL	Immediately
	INTERNALSESS	Immediately
	LANGUAGE	Immediately
	LOCALNODE	When system is recycled
	LOGCMDAUTH	Immediately
	LOGDISC	Immediately
	LOGMnnx	At next session start, or next 'CLOSE PASS', or next exit to the Signon screen.
	LOGOFF	At next session start
	MENU	On next display of the panel
	MISER	At next session start
	MOBILE	At next signon
	MSGID	At next signon
	MSGSUFFIX	Immediately
	MULTUSER	At next signon of a 'shared user'
	NCSESC	Immediately
	NETID	At next session start
	OLACLASS	Immediately
	OLA_DEFER_USERS	When system is recycled
	ONESCAPE	At next session start
	ONREAD	At next session start
	ONWRITE	At next session start
	OPEROLACLASS	Immediately
	OUTPUTEXIT	When system is recycled
	OUTPUTWARN	At next session start
	PANELID	Immediately
	PASSPHRASE	At next signon
	PASSTIMEOUT	At next session start

Control Statement	Parameter	When Changes Become Effective
	PASSTRANSID	At next session start
	PASTE	At next signon
	PCTRANSFER	At next session start
	PASSTRY	Immediately
	PREFLANGUAGE	Immediately
	PREVIOUS	At next signon
	PSTKAPPL	At next session start
	PSTKUSER	At next session start
	PULL	At next signon
	PUSH	At next signon
	PUSHLIMIT	Immediately
	QUITACTIVE	At next session start
	RCMDTIMEOUT	Immediately
	REBIND	At next signon
	RECORDLIMIT	Immediately
	RECOVERYLEVEL	When ACB is next opened
	REJBB	At next session start
	REMOTE	At next session start
	REPLAY	At next signon
	RETRCMDS	When system is recycled
	RMISER	At next session start
	RTMT1	At next session start
	RTMT2	At next session start
	SAUTOSEQ	At next session start
	SCREENMODE	Immediately
	SECURITY	At next signon
	SENDCDONSRD	Immediately
	SEQUENCE	Immediately
	SESACB	When system is recycled
	SESSAUTOSAPPL	Immediately
	SESSAUTOS	Immediately
	SESSDATAx	At next session start
	SESSPRIAPPL	Immediately

Control Statement	Parameter	When Changes Become Effective
	SESSPRI	Immediately
	SESSPROGMSG	At next session start
	SESTYPE	Immediately
	SHARE	At next signon
	SHAREAPPL	If already specified, it cannot be updated. If not originally specified, takes effect immediately.
	SHAREDISC	At next signon
	SHARESESS	At next signon
	SIDLTIME	At next session start
	SIGNON	At next signon
	SIGNONPANEL	At next signon
	SIMRECON	Immediately
	SNABUSY	At next session start
	SPYABLE	At next spy operation
	SPYGROUP	At next spy operation
	SRBUFSIZE	Immediately
	STANDBY	When system is recycled
	STARTSCRIPT	At next session start
	STATS	At next session start
	SYSDUMP	Immediately
	SYSPLEXGROUP	When system is recycled
	SYSPLEXTYPE	When system is recycled
	TCP	Immediately. Specifying NO will not deactivate a currently active TCP system. Specifying YES will activate a TCP system if there is not one active.
	TERMERROR	Immediately
	TERMSCRIPT	At next session start
	TRANSTAB	At next signon
	TRBUFSIZE	Immediately
	TRNUMBER	Immediately
	UNBIND	At next session start

Control Statement	Parameter	When Changes Become Effective
	UNDERISZSMGR	At next session start
	(Note: In OLA it is displayed as TYPE)	
	USERDATAx	Immediately
	VERBOSE	Immediately
	WINDSCRIPT	At next start of windows
	WORKQUE	Immediately
APPL		
	INDRANGE	Immediately
	INQUIRE	Immediately
	RECOVERYLEVEL	When ACB is next opened
	TERMLOGMODE	Immediately
	Other parameters	As PROFILE
AUDITROUTE		
	All parameters	Immediately
COMMAND		
	All parameters	Immediately
COPY		
	Change is not applicable but new COPY statements may be added.	
GROUP		
	All parameters	Immediately
HCFORMAT		
	All parameters	Immediately
HCPROFILE		
	All parameters	Immediately
HCROUTE		
	All parameters	Immediately
LINK		
	All parameters	If STARTLINK=NO is set, effective at next STARTLINK command. If STARTLINK=YES is set, and the link is not already active, then immediately; if link is active, when the link stops.
MESSAGE		

Control Statement	Parameter	When Changes Become Effective
	All parameters	Immediately
OPTION		
	MDPROF	When system is recycled
	PRINT	Immediately
	SECURITY	At next signon
	Other parameters	When system is recycled
PANEL		
	All parameters	On next display of the panel
PATCH		
	All parameters	Immediately
PATCHSU		
	All parameters	Immediately
PHEADER		
	All parameters	On next display of panel
PCONTENT		
	All parameters	On next display of panel
PTRAILER		
	All parameters	On next display of panel
PPROCESS		
	All parameters	On next display of panel
PROFILE		
	ACB	When system is recycled
	ACTIVESESSIONS	Immediately
	AFFINITY	At next signon
	AUTH	Immediately
	AUTOSELECT	At next signon or reconnect
	AUTOSEQ	At next signon
	AUTOSEQ A|I|E	Immediately
	AUTOSTART	At next session start or reconnect
	BACKWARD	At next signon
	BRDVAR	Inactive session: effective immediately. Active session: at next session start.

Control Statement	Parameter	When Changes Become Effective
	BRECEIVE	At next signon
	CMDACTIONKEY	At next signon
	COMMANDPRFXVAL	At next signon
	CONCEAL	Immediately
	CONDLOGOFF	Immediately
	CURESC	Immediately
	CUT	At next signon
	DAPPLCHECK	Immediately
	DAPPLESMAUTH	Immediately
	DEMO	At next signon
	DESCRIPTION	Inactive session: effective immediately. Active session: at next session start.
	DISACTIVE	At next signon
	DOUBLESC	Immediately
	DROP_SESSION	At next signon
	DSESSRANGE	At next signon
	ERTIMEOUT	At next signon
	ESCAPE	At next signon
	ESCAPE CURSOR	Immediately
	ESMLEVEL	At next signon
	ESMOLAGROUP	At next signon
	EUTIMEOUT	At next signon
	FORWARD	At next signon
	HCPROF	Immediately
	HCREQUEST	At next signon
	HCREQUEST CURSOR	Immediately
	IDLEDISC	At next terminal enter
	IDLELOCK	At next terminal enter
	IDLELOGOFF	At next terminal enter
	IMSCONVERT	Immediately
	IMSCONVERTC	Immediately
	INTERNALSESS	Immediately

Control Statement	Parameter	When Changes Become Effective
	LANGUAGE	At next signon
	LOGDISC	Immediately
	MENU	Next time panel is displayed
	MOBILE	At next signon
	MSGID	At next signon
	NCSESC	Immediately
	OLACLASS	Immediately
	PASSTRY	Immediately
	PASTE	At next signon
	PREVIOUS	At next signon
	PULL	At next signon
	PUSH	At next signon
	PUSHLIMIT	Immediately
	REBIND	At next signon
	RECORDLIMIT	Immediately
	RECOVERYLEVEL	When ACB is next opened
	REPLAY	At next signon
	Selection commands. (Note: In OLA they are displayed as SELCMDS)	Immediately
	SENDCDONSRD	Immediately
	SEQUENCE	Immediately
	SESSAUTOSAPPL	Immediately
	SESSAUTOS	Immediately
	SESSPRI	Immediately
	SESSPRIAPPL	Immediately
	SESTYPE	Immediately
	SHARE	At next signon
	SHAREDISC	At next signon
	SHARESESS	At next signon
	SIGNON	At next signon
	SIGNONPANEL	At next signon
	SIMRECON	Immediately

Control Statement	Parameter	When Changes Become Effective
	SPYABLE	At next spy operation
	SPYGROUP	At next spy operation
	TERMERROR	Immediately
	TRANSID	Immediately
	TRANSTAB	At next signon
	USERDATAx	Immediately
	WINDSCRIPT	At next start of windows
	Session Default	At next session start
RANGE		
	All parameters	Immediately
RUSER		
	All parameters	Immediately
SCRIPT		
	All parameters	Immediately
USER		
	PASSWORD	At next signon
	PROF	Immediately
	RENUMDUP	Immediately
	Other parameters	As PROFILE
TERMINAL		
	PASSTRY	Immediately
	PROF	Immediately
	SIGNON	At next signon
	SIGNONPANEL	At next signon
	Other parameters	As PROFILE
TRACEROUTE		
	All Parameters	Immediately
TRANSTABLE		
	All parameters	Immediately

Special notes

Amendments to certain parameters, although valid, are ignored by the update processing.

The PROFILE options and limits, and the session and key definitions are updated at the next session start if they are taken from the SYSTEM statement.

Any PANEL or SCRIPT tracing will stop if an UPDATE is done which affects any Enduser parameter defined on a PROFILE, TERMINAL or USER statement.

The user exit

The User exit is treated in a different way to other variables. When the configuration statements are processed by an `UPDATE CONFIG` *`member`* command and no new exit name is supplied, then the exit is not reloaded. When a new exit name is supplied, a load is attempted and if successful, the old exit is deleted. If it is required to load an exit having the same name as the original, the `UPDATE EXIT` command must be used. A name or a '0' must be specified with this form of the command. `UPD EXIT 0` causes the current exit to be deleted. It should be noted that Session Manager is unable to check the success or failure of a load until after the load is attempted, so when the same name is used, the old exit is deleted and a load failure will result in a system with no User exit. For this reason, specification of a new name is the preferred method.

Before a new exit is loaded, either with the same or a different name, the old exit is called at the User Exit Replacement point (E71). This enables the old exit to close any files or perform any tidying function before the new exit is invoked.

Exit scripts

Exit scripts can be specified to run instead of the exit points of the User exit. All exit points can be specified as scripts except `E01`, `E09` and `E71`, which are only applicable to the User exit.

The Exit scripts can be updated with the `UPDATE` command, using the `E`*`xx`* operands to specify which script is to be updated and 'xx' is the exit point. Whenever an `UPDATE` command is entered for one of these operands (even if a script cannot be found) a previous setting of the 'don't call again' switch is reset for the specified call point. The 'don't call again' switch is set by an exit, or script, setting its return code to 40. It can also be reset by the return code 44, or an `UPD EXIT` command.

CHAPTER 28

Commands

IBM Session Manager for z/OS has a set of commands available to users for controlling the sessions on their Menu screen. This set of commands may be entered at the Menu screen and enable a user, for example, to stop a particular session, to redisplay messages, or to quit from the Session Manager system. Some of the commands may also be entered on other Session Manager supplied panels.

There are some additional commands which would normally only be issued by a privileged user such as the Operations or Network Controller, or the System Administrator. These are commands for starting, stopping or tracing various tasks in the system. Normally, these commands would only be issued on the advice of your local Support Representative. Such commands may be issued from the system console, or from any terminal attached to Session Manager.

Note These commands may also be issued by a general user provided that the authorization code is properly set up.

An installation can define its own commands, or replace most Session Manager commands with its own particular requirements, using command scripts. See the *Panels, Scripts and Variables* manual for details.

The commands in this chapter are grouped into 'Basic' and 'User' (the General user commands - see page 222) and 'Operator' and 'Administrator' (the Privileged user commands - see page 245). A chart outlining the displays from which each command can be issued is provided in 'Table of commands' on page 217.

Most Session Manager command keywords and command parameters may be abbreviated. For example, the `DISCONNECT` command could be entered as `DISCONN`, `DISC` or even `DIS`. This is indicated in the command prototype by the mandatory portion being shown in upper case. Commands or command parameters which are already three characters long, or less, cannot be abbreviated. The substitutable parts of a command are indicated by *`italics`*.

Who can issue commands?

Security codes

Each Session Manager command has a security code allocated to it. The security code is a numeric digit from 1 to 9, where 1 is the least secure and 9 is the most secure. Therefore, code 1 is used for generally available commands, and 9 is for security-sensitive commands, such as `CLOSEDOWN`. There is a default security code associated with each command which may be overridden by an Installation, using the `COMMAND` statement in the Session Manager configuration. The `COMMAND` statement also enables different security codes to be applied to the parameters of any command, enabling a command to be partially restricted.

A list of the commands and the default security code allocated to each one is given in the description of the `COMMAND` statement in 'Default authority list' on page 159.

Privilege (authorization) levels

Each user in the Session Manager system has an associated privilege (or authorization) level. This can be related to the user's name, or the terminal being used, or the profile which is being used. The privilege (or authorization level) for a user is a single digit from 1 to 9. Like command security codes, 1 is the lowest and 9 is the highest level of privilege. If a privilege level is not specifically defined, a level of 1 is assumed. This process is described under the `AUTH` parameter in the Common Enduser Parameters section earlier in this manual.

Controlling access to commands

In summary therefore, the commands which a Session Manager user is permitted to issue are determined by the security code which has been allocated to the command, and the privilege level which has been ascribed to the user. A user may only issue those commands for which the security code is equal to, or less than, their privilege level. If security codes have been set for individual command parameters, some forms of specific commands may be restricted. By using the security codes and privilege levels appropriately, several command 'groups' may be set up, each group accessible by only certain users.

Table of commands

The table below shows 'categories' of commands.

They are divided into 'User' commands (with a default authorization level of 1), 'Operator' commands (with a default authorization level of 5), and 'Administrator' commands (with a default authorization level of 9). In addition, there is a 'Basic' category, which includes the simplest commands used for navigating around Session Manager. All of the 'Basic' commands have a default authorization level of 1.

General user commands		Privileged user commands	
Basic	**User**	**Operator**	**Administrator**
BACKWARD	ADDSESS	BLOCK	BROADCAST
BWD	BRECEIVE	DEMO	CLOSEDOWN
DOWN	CONCEAL	DLOG	DELETE BROADCAST
FORWARD	CONFIRM	FLASH	DELETE MSG
FWD	CUTEND	PLAYDS	DSTORE
HELP	CUTSTART	PLAYHEX	DTERM
QUIT	DELSESS	PLAYIMAGE	DUMP
RETRIEVE	DISCONNECT	RECORD	FORCE
RETURN	END	REPLAY	GFS
TOP	FILTER	SPY	INQUIRE
UP	FIND		ISZECLP
	HALTSCRIPT		ISZTEST
	HARDCOPY		PASSFREE
	HCOPTION		PUPDATE
	INITSC		QTASK
	LEFT		REMOVEUSER
	LOCK		SECFRESH
	LOCKTERM		SPIN
	LOGOFF		STARTCP
	MSG		STARTLINK
	MSGID		STOP
	NLOG		STOPACB
	NW		STOPLINK
	OK		STOPTCP
	PANELID		SWITCHPLX
	PASTESTART		TERMINATE
	PCTRANSFER		TRACE
	PULL		TTPSL
	QACTUSER		UPDATE
	QQUIT		
	QUERY		
	QUSER		
	RESET		
	REVEAL		
	RIGHT		
	SE		
	SEND		
	SME		
	SPYOFF		
	STARTSC		
	TRANSFER		
	VIEW		
	WINDOWS		

Command syntax

A quick syntax reference for all the commands is provided in the form of command prototypes in the *Quick Reference* manual.

Where can commands be issued?

Local and Sysplex commands

Commands can either be issued to the local Session Manager node or to other networked Session Manager nodes.

Commands that are to be actioned only on the local node can be issued on the command line, via the System Management Menu facility, via the Sysplex Summary and Menu facility or via the Help Desk facility. Commands that are to be actioned across a Sysplex network to another node or other nodes can be only be issued via the Sysplex Summary and Menu facility or via the Help Desk facility, since commands issued from the command line or the System Management Menu facility are only actioned on the local node.

System Management menu

As well as being able to issue commands to the local node directly in Session Manager, a list of the commands that the user is authorized to issue to the local node is available on the System Management menu. If so configured, this list appears as a Session Manager session on the main menu and can be accessed in the same way as any other session. Commands can be selected by tabbing to the appropriate command and pressing Enter, or by entering 'S' beside the command to be selected and pressing Enter.

For further information on using the System Management menu in Session Manager, refer to the *User and Administrator* manual.

Sysplex Summary and menu

To issue commands to the other nodes in a Session Manager Sysplex network, commands should be issued by using the Sysplex Summary and Menu facility. If so configured, this list appears as a Session Manager session on the main menu and can be accessed in the same way as any other session.

For further information on using the Sysplex Summary and Menu facility in Session Manager, refer to the *User and Administrator* manual.

Help Desk facility

As well as being able to issue commands to the local node directly in Session Manager, a subset of the commands that the user is authorized to issue is available in the Help Desk facility. If so configured, this list appears as a Session Manager session on the main menu and can be accessed in the same way as any other session.

For further information on using the Help Desk facility in Session Manager, refer to the *User and Administrator* manual.

Panel command line

Several sets of commands are grouped together below and assigned a letter for use in the subsequent table.

Code	Commands
a	CONFIRM, CUTEND, CUTSTART, LEFT, NW, OK, PASTESTART, RIGHT, SE, PLAYDS, PLAYHEX, PLAYIMAGE, PULL, SWITCHPLX
b	ADDSESS, DELSESS, FILTER, FIND, LOCKTERM, NLOG
c	DEMO, DLOG, DSTORE, DTERM, GFS, QTASK, REPLAY, SEND, SPY
d	BACKWARD, BWD, DOWN, FORWARD, FWD, TOP, UP
e	BLOCK, BROADCAST, CLOSEDOWN, DELETE DUMP, FLASH, FORCE, INQUIRE, ISZTEST, PASSFREE, SECFRESH, SPIN AUDIT, STARTCP, STARTLINK, STOP, STOPACB, STOPLINK, STOPTCP, TERMINATE, TRACE, TTPSL, UPDATE

The following table provides information on the Session Manager commands that can be issued in each type of environment. It shows the commands that are valid in the Windows environment, those that are valid in session scripts, those that can be issued remotely, and so on. Basic commands' refers to the Basic commands category from the 'Table of commands' on page 217.

Context	Commands that can be issued
Menu Panel	All commands except **a**
Demo Panel	All commands except **a**, **b**
TPSL Trace Panel	All commands except **a**, **b**
GFS Panel	All commands except **a**, **b**
Broadcast Panel	All commands except **a**, **b**, **c**
Query Panel	All commands except **a**, **b**, **d**
Signon Panel	Basic commands, plus LOGOFF, QQUIT
Spytell Panel	HELP, MSGID, RETRIEVE, RETURN, QUIT, OK
Pull Panel	Basic commands, plus PULL
Replay Panel	Basic commands, plus PLAYDS, PLAYHEX, PLAYIMAGE, PULL, RECORD
Cut Panel	Basic commands, plus CONFIRM, CUTEND, CUTSTART, LEFT, NW, PASTESTART, RIGHT, SE
Paste Panel	Basic commands, plus CONFIRM, CUTSTART, LEFT, NW, PASTESTART, RIGHT
Windows (Panel or Script ISZCMD)	All commands except DEMO, DSTORE, DTERM, GFS, REPLAY

Context	Commands that can be issued
Exit Script ISZCMD	All commands except Basic commands, and **c**
Session Script ISZCMD	All commands except **a**, **b**
Remote Session Script ISZCMD	Basic commands plus **e**, and DLOG, GFS, SWITCHPLX, WINDOWS
SEND from console	**e**, plus SWITCHPLX, QTASK, QUERY
SEND from user	**e**, plus SWITCHPLX, QTASK, QUERY, DLOG, DSTORE, HELP
At remote display	Basic commands plus **e**, DLOG, DSTORE, QTASK, QUERY
Console	**e**, plus SWITCHPLX, QTASK, QUERY, SEND

The commands are described in greater detail in the following sections where they are divided into General user ('Basic' and 'User') and Privileged user ('Operator' and 'Administrator') commands.

General user commands

Most of these commands may be issued by any user from the Menu screen, or any screen with an 'Enter Command' field. A number of the commands, such as BWD and FWD are specific to certain screens, in this instance those screens which are pageable.

All the commands contained in this section have a default security authorization level of '1' unless otherwise stated.

ADDSESS user command

ADDSESS *applname sessionnumber*

The ADDSESS command enables users to add sessions to their Session Manager menu. If the command is accepted then the refreshed menu will contain the added session, which can then be selected as normal. The session will remain on the menu for the duration of the user's Session Manager session or until a DELSESS command is issued.

applname is a mandatory value on the command. If the DAPPLCHECK parameter (see page 35) is set to Yes a check is made to ensure that the *applname* exists in the Session Manager configuration. If it does not then the command will be rejected.

If the E22 exit is installed and the DAPPLESMAUTH parameter is set to Yes (See 'Common end-user parameters' on page 30) then the External Security Manager will be called to ascertain if the user has authority to access the application.

sessionnumber is an optional value on the command. If a *sessionnumber* is provided on the command then a check will be made to ensure that it is not currently assigned and if it is then the command will be rejected.

If no session number is provided then a session number will be allocated as follows:

> The configuration parameter DSESSRANGE (see page 36) can be used to allocate a range of session numbers to be used for dynamically added sessions. This can also be used to limit the number of dynamically added sessions a user can concurrently have. If the command is issued by a user who has exhausted their specified range then the command will be rejected.
>
> If DSESSRANGE has not been specified then the first dynamically added session will be assigned the session number 9999. Additional dynamically added sessions will be assigned 9998, 9997 and so on.

BACKWARD basic command

Backward|Bwd [*nnnn*]

The BACKWARD command applies to panels which are pageable due to the number of content lines exceeding the screen size. The command causes the previous page to be displayed if the number of pages is omitted. *nnnn* may be any value in the range 1-9999 pages.

Note Header and trailer sections of a panel are not pageable, these remain static.

BRECEIVE user command

```
BRECeive [Yes|No|ON|OFF|WAITinp|Queue|BELL]
```

The BRECEIVE command can alter the broadcast/message receiving status. It overrides the status defined by the BRECEIVE parameter of the SYSTEM, PROFILE, USER or TERMINAL configuration statements (see page 32).

WAIT causes messages to be stacked and held until the next time an attention key is pressed.

QUEUE causes messages to be stacked until the next escape to the Menu screen.

If a user wishes to know that a broadcast has been sent, but still requires message stacking, then specifying BELL causes the terminal alarm to sound on receipt of each broadcast.

The broadcast screen is automatically displayed on returning to the Menu screen if any messages are waiting, and the stack is cleared once the broadcasts have been viewed.

Yes causes broadcasts to be received immediately. The current screen is displaced, but may be recalled by pressing PF3.

No suppresses all broadcasts apart from those sent using the URGENT parameter.

Whenever this command is issued, a message confirming the new status is displayed. When no parameters are specified, the current receiving status is displayed.

Note If PCTRANSFER=Yes has been set, the current BRECEIVE setting is ignored and Broadcasts are queued until the PCTRANSFER setting has been set to Off.

CONCEAL user command

```
CONCEAL [PFkey|session_number|this_applid]
```

The CONCEAL command is used to dynamically prevent one or more unwanted sessions from being displayed on the user's menu. The typical use for this would be where the External Security Manager (for example, RACF) is used to control what appears on a user's menu (that is, the menu is based on what a user is permitted access to) and this produces a list containing 'noise' entries that are rarely, if ever, used.

If the parameter starts with 'PF' and the remainder is numeric 1-24, then it is treated as a PFkey number; if the parameter is all numeric 1-9999 it is treated as a session number; otherwise it is treated as an applid. When concealing sessions by their applid and you have more than one session with the same applid then the first session will be concealed. To conceal the other sessions with the same applid, reenter the CONCEAL command specifying the applid.

Sessions are only concealed for the duration of the user being signed on. If the user signs off then when they sign on again all the previously concealed sessions will be visible.

To permanently conceal a session set the common session parameter CONCEAL to YES.

This command is incompatible for users with submenus configured and will be rejected as invalid if issued.

See also 'REVEAL user command' on page 241.

CONFIRM user command

```
CONFIRM
```

The CONFIRM command is used in a paste operation to verify that a paste is to be performed to the selected session. If the CONFIRM command is not issued then the data is not pasted into the session.

CUTEND user command

```
CUTEnd
```

CUTEND is a synonym of the SE command.

CUTSTART user command

```
CUTStart
```

CUTSTART is a synonym of the NW command.

DELSESS user command

```
DELSESS [sessionnumber|ALL]
```

The DELSESS command is used to remove dynamically added sessions from the user's menu. A dynamically added session can also be removed by entering a 'D' alongside the session on the user's menu.

A check is made to ensure that the session exists and if not, the command will be rejected. If the session does exist but has not been dynamically added, or if the session was added dynamically but is active then the command will be rejected.

If the ALL keyword is entered on the command then all inactive dynamically added sessions will be removed. Any active dynamically added sessions will remain.

DISCONNECT user command

```
DISconnect [Exit|Signon|Logon]
```

The DISCONNECT command disconnects the terminal at which the command is issued from Session Manager. The command is rejected if there are any active sessions for which either QUITACTIVE=No or DISCACTIVE=No is in effect.

Any sessions active at the time the command is issued are maintained. If this user reconnects to the system from a different terminal, Session Manager is able to transfer the Menu screen and all the active sessions to the new terminal. It should be noted that reconnection to a different physical terminal will fail if the sessions are of an incompatible type. For example, it is not possible to reconnect to a model

2 terminal if there were model 5 sessions active at the time the DISCONNECT command was issued. Instead, the TRANSFER command with the OVERRIDE parameter can be issued at the Signon panel. This allows reconnection, although some screen displays may be misaligned.

Note The Windows command may be used to view any portion of such displays in the proper alignment, regardless of the terminal model type.

The screen displayed after this command is issued is either the VTAM Logon screen or the Session Manager Signon screen, depending upon the specification of the LOGDISC parameter of the SYSTEM, PROFILE, TERMINAL or USER statement. It is possible to override the defined default by using one of the command operands as described in the following paragraphs.

EXIT causes the VTAM Logon screen to be displayed. To reaccess Session Manager, the user must therefore reestablish contact via a VTAM logon.

SIGNON and LOGON causes the Session Manager Signon screen to be displayed. It is therefore not necessary to perform the VTAM logon again.

DOWN basic command

Down [*nnnn*]

The DOWN command may be used on multi-page panels to move down the pageable portion by the number of lines specified on the command. The default is one line if a number is not specified. *nnnn* may be any value in the range 1-9999 lines.

Note Header and trailer sections of a panel are not pageable, these remain static.

END user command

END *seln-id*|*

The END command causes the script defined by the ENDSCRIPT parameter for the selected session id to be run for the application associated with that session. *seln-id* may be the required session detail number, a PF key number, or a selection command sequence.

To run the script for the relevant applications for all active sessions, 'END *' should be entered.

Any session(s) for which there is no ENDSCRIPT parameter defined is simply reset. It should normally be the case that the script ultimately causes the session to be terminated – as if a RESET command for the session had been issued. If this is not the case, Session Manager issues either a VTAM unconditional logoff, or resets the logical device after the script has completed.

FILTER user command

FILTer [*applid*]

The FILTER command, which can only be used in the Session Manager menu panel, causes only those sessions which exactly match the filter to be displayed. The wildcards '*' (meaning any character or set of characters) and '+' (meaning any single character) can be used within the *applid*. The *applid* can be between one and eight characters.

If the FILTER command is issued without an *applid*, or with just the wildcard '*', the filter will be reset and all sessions will be displayed.

FIND user command

FIND [*applid*]

The FIND command, which can only be used in the Session Manager menu panel, causes the part of the session list containing the next match to the specified *applid* to be found. The match will be the first of the sessions displayed. The wildcards '*' (meaning any character or set of characters) and '+' (meaning any single character) can be used within the *applid*. The *applid* can be between one and eight characters.

If the FIND command is issued without an *applid* then the list will be scrolled to display the next match using the *applid* previously specified. The TOP command (see page 243) can be used to return to the top of the list.

FORWARD basic command

Forward|Fwd [*nnnn*]

The FORWARD command may be used to display the subsequent pages of a multi-page panel. The next page is displayed when the number of pages is omitted. *nnnn* may be any value in the range 1-9999 pages.

Note Header and trailer portions of a panel are not pageable, these remain static.

HALTSCRIPT user command

HALtscript *seln-id*|*

The HALTSCRIPT command can be used to terminate the execution of a currently running script, when it is entered at a Menu screen. If the script is an ENDSCRIPT, the command causes either an immediate unconditional logoff or a reset of the logical device. For other types of script, the session remains active.

seln-id can be the number of the session, a selection command sequence, or a PF key number.

HALT * terminates the execution of all scripts running for active sessions for the Menu screen.

HARDCOPY user command

HARdcopy *seln-id* [Profile *hcprofile* Option *hcoption*]

The HARDCOPY command causes a hardcopy of the specified session's screen to be taken. The specified session must be active for the command to be actioned. If the session specified is inactive, then message 0715E is issued.

seln-id can be the number of the session, a selection command sequence, or a PF key number.

hcprofile specifies the name of a hardcopy profile defined by a HCPROFILE statement.

hcoption identifies which formatting and routing options are to be used. These are defined by the HCOPTION parameters within the HCPROFILE statement. The *hcoption* number identifies the relative position of the HCOPTION parameter within the HCPROFILE statement.

By default, the hardcopy is sent using the routing instructions identified by the HCROUTE parameter of the SYSTEM, PROFILE, USER or TERMINAL statement.

The Profile and Option keywords may be used to override the User's hardcopy profile with an alternate. For example, a hardcopy for a particular application screen may need to be routed to the destination specified by an auditor.

HCOPTION user command

HCoption [*option-number*]

The HCOPTION command is entered at the Menu screen to select a hardcopy option prior to invoking a screen hardcopy. The operand *option-number* indicates the relative position in the list of the required option. If *option-number* is omitted, the next hardcopy option in the list is selected. If the current option is the last, then the first option is selected. At signon time the first option is in effect.

HELP basic command

Help [*help-topic-name*] | [*panelname* PANEL]

Help panels are supplied with Session Manager to provide assistance in the use of the product. Help may be invoked from any Session Manager panel which has a command area.

When the command HELP is entered without any topic name, a help menu panel is displayed from which help for a specific topic may be selected by entering the appropriate number in the command area.

A specific topic may be selected directly by entering the HELP command followed by the topic name. The topic-name is restricted to seven characters because Session Manager searches for a panel with the name H followed by the topic name. For example, if 'H COMMAND' is entered, Session Manager looks for a panel named HCOMMAND. If the panel is not found, the help menu panel is displayed.

Rather than specify a topic name, it is possible to enter a panel name, which must then be followed by the PANEL parameter. This prevents Session Manager prefixing the name with the letter 'H', so the command 'H USERHELP PANEL' requests Session Manager to display a panel called USERHELP.

The RETURN command (usually assigned to PF4) returns immediately from any help screen back to the point of invocation of the HELP command. The QUIT command (usually assigned to PF3) returns to the previously displayed help screen, or to the point of invocation of help if there is no previously displayed screen.

INITSC user command

`INITSC Yes|No|ON|OFF  seln-id|*`

The `INITSC` command can be issued by a suitably authorized user to disable or (re-)enable the running of the session 'initialization' script specified using the common session parameter `INITSCRIPT` (see page 57). Both parameters are mandatory. (Clearly, if a session 'initialization' script has not been specified in the configuration then the `INITSC` command will have no effect.) The command does not work against hidden sessions.

`INITSC=No` or `OFF` disables running of the script; `INITSC=Yes` or `ON` (re-)enables running of the script. The new session variable `s_runinitsc` will be set appropriately – that is, `Y` for `Yes|ON` and `N` for `No|OFF`. For more information, see the *Panels, Scripts and Variables* manual.

`seln-id` can be the number of the session, a selection command sequence (only one such sequence on a session is supported), or a PF key number.

`INITSC *` disables the running of the 'initialization' script for all sessions associated with the Menu screen.

LEFT user command

`LEFT [nn]`

The `LEFT` command moves the display area to the left on a panel which is wider than the display area. The default number of columns to move is 1. *nn* may be any value in the range 1-99.

LOCK user command

`LOCK [Yes|No|ON|OFF]`

The `LOCK` command prevents users that are configured as mobile from transferring their sessions from one terminal to another.

`LOCK=No` may be issued to unlock the sessions and enable transfer. The `LOCK=Yes` command locks the sessions to this terminal.

Unless the lock option has been used to prevent transfer, a mobile user's sessions may be acquired from another terminal at any time.

The default is `ON` when no parameters are specified.

LOCKTERM user command

`LOCKTERM`

The `LOCKTERM` command enables a user to lock their terminal. `LOCKTERM` displays a panel requesting the user's password in order to unlock the terminal. `LOCKTERM` can be issued from the Session Manager menu or from within an active session. In order to allow the command to be issued from within an active session, the following `SAUTOSEQ` parameter must be configured: `SAUTOSEQ LOCKTERM N LOCKTERM`.

LOGOFF user command

```
LOGoff [Exit|Signon|LOGon]
```

The LOGOFF command terminates the user or terminal from Session Manager. Any sessions active at the time this command is issued are terminated. This command may alternatively be keyed as 'QQ'.

The screen displayed after this command is issued is either the VTAM Logon screen or the Session Manager Signon screen. EXIT requests the VTAM Logon screen and SIGNON or LOGON requests the Session Manager Signon screen.

A parameter specified on this command overrides any EXIT, SIGNON or LOGON option specified on the LOGDISC parameter of the SYSTEM, PROFILE, TERMINAL or USER control statement.

Note If an active session is configured as QUITACTIVE=No, the LOGOFF command is rejected.

MSG user command

```
MSG message-text
    Applid applid        |
    Group groupname      |
    Lu luname            |
    Profile profilename  |
    User userid
    [ [Hold hh:mm] | [For hh:mm|days|days hh:mm] ]
    [URGent]
```

The MSG command sends a message to a terminal user, or group of users. The length of the message may be 256 characters *minus* the length of the command and any parameters, so the effective limit is approximately 240 characters. The message text appears at the recipient's terminal on a special panel; the same as that used by the BROADCAST command. The MSG command is a restricted form of the BROADCAST command in that the destination cannot be specified as a pattern, or ALL.

The message panel interrupts the recipient's current session to display the message, unless the broadcast receiving status is set such that messages are stacked, or the recipient is running a Windows script which can receive messages. The explanation for the BRECEIVE command gives a description of the different receiving modes.

The message destination may be specified using one of the following parameters:

Applid *applid*

All Session Manager users logged on to the given VTAM *applid*

Group *groupname*

All users who are members of the stated *groupname*

Lu *luname*

The terminal identified by *luname*

Profile *profilename*

All terminals and users using the nominated *profilename*

`User` *`userid`*

The user with this id

The `MSG` command only accepts specifically named destinations. The `BROADCAST` command should be used when a destination must be specified generically.

`Hold`

causes the retention of the message until the specified time, which may be up to 24 hours from the current time. If the destination is a user or a terminal, once the message has been displayed at that destination during the signon procedure it is not displayed again at subsequent signons. However, if the destination is a profile, a group or an application id, which may be used by a number of different users, then a user signing on several times will be presented with the message at each signon.

`FOR`

enables a message to be held for a relative period of time. The message is sent to the destination when that destination logs on, as long as the time period has not expired. The `FOR` and `HOLD` parameters are mutually exclusive.

`URGENT`

overrides any inhibiting `BRECEIVE` setting in order to display the message immediately at the destination terminals. This parameter should be used with consideration, since if used for trivial messages, a certain amount of irritation may easily result.

If you are using a Sysplex

When operating with the `SYSPLEXGROUP` keyword specified on the `SYSTEM` statement, by default all messages and broadcasts sent with the `MSG` user command and the `BROADCAST` operator command will be distributed across the Sysplex. For example if a `BROADCAST` command was issued in one Session Manager instance the command would be sent to all other Session Manager instances in the same Sysplex group. This global scope of the Session Manager `MSG` and `BROADCAST` commands can be changed in two ways, via global variables `t_global_msg` and `t_global_msgdef`. The latter is a read-only variable reflecting the value set by (`SYSTEM` statement) parameter *`SYSPLEXGroup GLOBALMessages`*.

If a message has been restricted in scope to the issuing Session Manager instance (by use of variable `t_global_msg` or the `GLOBALMESSAGES` sub-parameter of the `SYSPLEXGROUP`) then only the issuing Session Manager instance will process it.

Consequently, regardless of which Session Manager instance a `QUERY Broadcast` is issued on, all held messages will be displayed irrespective of each message's scope.

Each held message is displayed as *`number qualifier-text`*,

where.

`number` is the message number,

`text` is the message text,

qualifier is either blank or displays one or more of the following special characteristics:

<U><G><,U=*xxxx*><,*yyyyy*>

where:

U indicates that the message was designated as urgent.

G indicates that the message was sent to a generic target.

U=*xxxx* (where *xxxx* is the user id) indicates that the message was queued after a user had received it then exited from the message display panel without acknowledging the message (as a result of expiry of the IDLELOGOFF time, for example).

yyyy (where *yyyy* is relevant Session Manager node name) indicates that the message was restricted to a Session Manager instance in a Sysplex environment (SYSPLEXGROUP parameter has been set on SYSTEM statement).

Incidentally, this approach enables a newly started Session Manager instance to have access to all held messages previously generated. Note that if all Session Manager instances in a Sysplex group are closed then the held messages are lost. However, provided at least one Session Manager instance is active, the held messages will be maintained.

The QUERY Broadcast display indicates the scope of messages that have been restricted to one Session Manager instance.

MSGID user command

MSGID [Yes|No|ON|OFF]

Message text may be preceded by a message id. The PROFILE, USER and TERMINAL configuration control statements contain an option to control the display or suppression of this message code. The MSGID command may be used to override the option at the terminal.

Yes causes any messages to be displayed with their codes. No causes the codes to be suppressed. ON is the default when no parameter is supplied.

In each instance, the previous message is redisplayed in the desired format.

NLOG user command

NLOG *VTAM-applid*

The NLOG command allows a user to log on natively to an application from the main menu.

This will bypass any Session Manager activity, including any scripts, and establish a session with the application. The user will be unable to escape from the session and can only return to the main menu by logging off from the application. If the E22 exit is installed then a check will be made with the External Security Manager as to whether the user has authority to invoke the application.

Recovery Level HIGH or INTERMEDIATE sessions can not be started with the NLOG command.

NW user command

```
NW
```

The NW command is used in the Cut and Paste operation to define the north west corner, or start, of the cut or paste area. The north west corner is the top left-hand corner of the area.

CUTSTART and PASTESTART are both synonyms of the NW command.

OK user command

```
OK
```

The OK command applies to the SPYTELL panel only. It enables the user who is the object of a spy operation to accept the spy operation.

PANELID user command

```
PANELID [Yes|No|ON|OFF]
```

Used when displaying a Session Manager panel, the PANELID command is used to switch the display of the panel identifier (ID) on and off. PANELID ON is the default when no parameter is supplied.

Although the PANELID command has a default authorization level of '1', it is intended for use primarily by Systems Programmers who are developing panels and scripts, but could also be used to assist Support during problem resolution.

The panel ID appears in the upper-left corner of the panel, overlaying whatever information is normally displayed there.

Note Using the PANELID command will override any setting specified in the SYSTEM PANELID option.

PASTESTART user command

```
PASTEStart
```

This is a synonym for the NW command.

PCTRANSFER user command

```
PCTRANSFER [Yes|No|ON|OFF] [seln-id]
```

The PCTRANSFER command enables the PCTRANSFER parameter setting of the SYSTEM, PROFILE, TERMINAL or USER statement, to be switched on and off dynamically for active sessions only. It should be used when a file is to be transferred to, or from, a PC connected to Session Manager.

The default is Yes when no parameter is supplied and the setting applies to all active sessions. The *seln-id* can be used to specify a particular session, if required. For example PF2, or just 2 is acceptable. These parameters can be supplied in any order, for example: PCTRANSFER PF2 OFF and PCTRANSFER OFF PF2 have the same effect.

The 'pctransfer' setting is changed for active sessions only. Therefore when PCTRANSFER is specified, MISER is switched off for all active sessions, data stream compression is not used, addname, addnode and addsid are not displayed (if applicable), the alarm is not sounded, broadcasts are queued, and Session Manager does not intercept any WSF RP(Q) since these are known to interfere with PC emulators. A PC transfer should not be attempted when Session Manager Windowing is being used at the terminal. Windows must be ended or suspended prior to the PC transfer.

PULL user command

PULL *index-number* [RETurn Yes|No|ON|OFF]

The PULL command can be used on the PULL index display to select one of the listed screens to be displayed. The screen is displayed as it appeared when it was first PUSHed. This is the only selection command available from the index display. *index-number* is the number assigned to the screen.

Note If the supplied PULL index display panel is used, the PULL keyword is already assigned to a PF key. Using this panel it is possible to just enter the number of the screen to be displayed and press the relevant PF key.

The optional RETURN operand can be used to specify whether the Index is to be displayed when quitting from a saved screen, or not. The default is RETURN=Yes, to display the Index. If RETURN=No is specified, then return is made to the current session.

QACTUSER user command

QACTuser

This command displays the number of active signed on users.

QQUIT user command

QQuit [Exit|Signon|Logon]

QQUIT is a synonym for LOGOFF, described previously on page 229.

QUERY user command

```
Query
    ACb acb_pattern                      |
    ALL                                  |
    Applid appl_pattern                  |
    BRDVAR id_pattern                    |
    Broadcast                            |
    Group groupname                      |
    Ident                                |
    ISZsmgr                              |
    LAstmsg                              |
    LU luname_pattern                    |
         [RTM [RESET] | RTMALL [RESET]] |
    Net ALLSESSIons                      |
        LInk link_pattern                |
        NETMan NOde nodename             |
        NOde node_pattern                |
        NUMbers                          |
        SESSion NOde nodename            |
                LInk linkname            |
                TAsk taskname            |
                CORrelator name_pattern |
    PROFile profile_pattern              |
    REMOTE node_pattern                  |
    SIGNON                               |
    SPY                                  |
    STAts                                |
    STOruse [Pool]                       |
    SUspend                              |
    TERMinal term_pattern
         [RTM [RESET] | RTMALL [RESET]] |
    TN3270 tcp-client_pattern
         [RTM [RESET] | RTMALL [RESET]] |
    User userid_pattern
         [user_qualifier RTM [RESET] | RTMALL [RESET]]
         [NONQual]
```

The QUERY command displays information on various aspects of the system. When input at a terminal, the reply is displayed on the DQUERY or the DATA panel. If the reply is longer than can be displayed on one screen, pressing Enter without other input shows the next screen of the reply. If the last, or only screen is currently displayed, pressing Enter re-issues the original QUERY command.

Paging commands are not effective during a Query display.

The QUIT, or RETURN, command causes the Menu screen to be displayed.

On several of the QUERY commands 'patterns' may be specified. This means that the wildcard characters + and * may be used to build a generic name for the query.

ACb *acb_pattern*

Displays the identity of users who have an active session with a VTAM application using an ACB conforming to the *acb_pattern*.

For example:

```
Q AC *CIC*
Q ACB SA7SW01
```

`ALL`

Displays the identities of all terminals and users signed on to the system.

`Applid` *`appl_pattern|APPL-name`*

Displays the identity of users who have an active session with a VTAM application with an applid conforming to the input pattern given by *`appl_pattern`*. For example:

```
Q APP SA9TSO
Q A ++8+*
```

Alternatively, an `APPL` statement name can be specified, and the active sessions using that definition and their users will be displayed.

`BRDVAR` *`id_pattern`*

Displays the identity of users and their sessions, both inactive and active, which have been assigned an id-string conforming to the *`id_pattern`* specified on the command.

`Broadcast`

Lists all the broadcasts and messages that are held. This enables individual messages to be deleted by issuing the `DELETE BROADCAST` or `DELETE MSG` command.

If you are using a Sysplex

When operating with the `SYSPLEXGROUP` keyword specified on the `SYSTEM` statement, by default all messages and broadcasts sent with the `MSG` user command and the `BROADCAST` operator command will be distributed across the Sysplex. For example if a `BROADCAST` command was issued in one Session Manager instance the command would be sent to all other Session Manager instances in the same Sysplex group. This global scope of the Session Manager `MSG` and `BROADCAST` commands can be changed in two ways, via global variables t_global_msg and t_global_msgdef. The latter is a read-only variable reflecting the value set by (`SYSTEM` statement) parameter *`SYSPLEXGroup GLOBALMessages`*.

If a message has been restricted in scope to the issuing Session Manager instance (by use of variable t_global_msg or the `GLOBALMESSAGES` sub-parameter of the `SYSPLEXGROUP`) then only the issuing Session Manager instance will process it.

Consequently, regardless of which Session Manager instance a `QUERY Broadcast` is issued on, all held messages will be displayed irrespective of each message's scope.

Each held message is displayed as *`number qualifier-text`*,

where.

`number` is the message number,

`text` is the message text,

qualifier is either blank or displays one or more of the following special characteristics:

<U><G><,U=*xxxx*><,*yyyyy*>

where:

U indicates that the message was designated as urgent.

G indicates that the message was sent to a generic target.

U=*xxxx* (where *xxxx* is the user id) indicates that the message was queued after a user had received it then exited from the message display panel without acknowledging the message (as a result of expiry of the IDLELOGOFF time, for example).

yyyy (where *yyyy* is relevant Session Manager node name) indicates that the message was restricted to a Session Manager instance in a Sysplex environment (SYSPLEXGROUP parameter has been set on SYSTEM statement).

Incidentally, this approach enables a newly started Session Manager instance to have access to all held messages previously generated. Note that if all Session Manager instances in a Sysplex group are closed then the held messages are lost. However, provided at least one Session Manager instance is active, the held messages will be maintained.

The QUERY Broadcast display indicates the scope of messages that have been restricted to one Session Manager instance.

Group *groupname*

Lists all active users in the given group.

Ident

Displays the userid, terminal id and profile name for the terminal at which the command is issued. It is displayed in the message area of the current display, rather than on the DATA panel.

ISZsmgr

Displays the time that Session Manager was loaded and lists the features that are currently enabled.

LAstmsg

Redisplays the last message output to the terminal, using the current MSGID setting.

LU *luname_pattern* [RTM [RESET] | RTMALL [RESET]]

Displays the identity of users, profiles, terminals, and the sessions running at terminals with an identity conforming to the luname pattern; for example: Q LU SA2LU0*.

The RTM and RTMALL parameters cause statistics from the Response Time Monitor to be displayed for sessions with STATS=Yes specified. See 'Using the Monitoring Statistics' in the *Installation and Customization* manual for details of the Response Time Monitor routine.

RTM displays the total response time counts per session.

RTMALL displays all counts; that is, network, application and total response times per session.

RESET causes all counts to be reset.

Net

Identifies that this is a network query.

ALLSESSions

Displays details of all sessions running from, to, and through this system.

LInk *link_pattern*

Displays details of a link or links which conform to the specified link_pattern. The link_pattern must correspond to the linknames entered on the LINK configuration statement.

NETMan NOde *nodename*

Displays details of the local and remote Network Manager tasks. Network Manager tasks may only be addressed by node names that they own. The nodename must be the correct one for the Netman being enquired upon and should correspond to a valid nodename as entered on a LOCALNODE parameter of the SYSTEM configuration statement

NOde *node_pattern*

Displays details of the node, or nodes, which conform to the specified node_pattern. The node_pattern must correspond to the nodenames entered on the LOCALNODE parameter of the SYSTEM configuration statement.

NUMbers

Displays network statistics.

```
SESSion NOde nodename |
        Link linkname |
        TAsk taskname |
        CORrelator name_pattern
```

Displays details of a specific session.

If NODE is specified, details are given of the sessions to, and from, the named node. *nodename* must correspond to a valid node name as entered on a LOCALNODE parameter of the SYSTEM configuration statement.

If LINK is specified, details are given of the sessions running through the named link. *linkname* must correspond to a valid link name as entered on a LINK configuration statement.

If TASK is specified, details are given of the sessions owned by the named task, *taskname*.

If CORRELATOR is specified, details are given of the sessions that match the pattern for the 16 byte corelator.

PROFile *profile_pattern*

Displays the identity of users, profiles, terminals, and sessions using profiles with names which conform to the specified *profile_pattern*.

`REMOTE` *`node_pattern`*

Displays the identity of remote nodes using names that conform to the specified *`node_pattern`*.

`SIGNON`

Displays the number of terminals connected to Session Manager.

`SPY`

Displays the identity of each user who is spying on another user.

`STAts`

Displays statistics regarding system activity and throughput which are useful when tuning the Session Manager system.

The statistics show information such as the number of inputs and outputs from terminals and the number of bytes sent. Similar information is also shown for sessions.

Statistics for data stream compression are also given. These show the number of compressed buffers with the number of bytes input and the number of bytes output. If applications are issuing WSFRPQs and Session Manager is simulating them, the number of session output bytes is greater than the compression input bytes. The difference between the two values is the number of bytes saved by simulating WSFRPQs.

Statistics for the MISER feature are shown. MISER statistics are described in detail in the *Installation and Customization* manual.

The number of ACBs currently open and the maximum number that were open are displayed along with the time that the maximum number was observed. Similar information is also displayed for receive RPLs, active terminals, and active sessions. Terminal receives are shown in pre-defined size ranges so that the best size for the `TRBUF` parameter on the `SYSTEM` statement can be calculated. Also shown is the number of times data has been received from sessions.

`STOruse [Pool]`

Displays information on the use of dynamic storage in various categories. The optional `Pool` parameter presents information on the usage of storage pools.

`SUspend`

Displays all terminals that are locked due to the user exceeding the `PASSTRY` limit. The command enables the System Administrator to easily identify a particular terminal that is locked so that it can be unlocked using `PASSFREE` command.

`TERMinal` *`term_pattern`* `[RTM [RESET] | RTMALL [RESET]]`

Displays the identity of users, profiles, terminals, and sessions using terminals with names which conform to the specified terminal pattern. It is a synonym for Query LU. For example: `Q TERM S10*`

The `RTM` and `RTMALL` parameters cause statistics from the Response Time Monitor to be displayed for sessions with `STATS=Yes` specified. See 'Using the Monitoring Statistics' in the *Installation and Customization* manual for details of the Response Time Monitor routine.

RTM displays the total response time counts per session.

RTMALL displays all counts; that is, network, application and total response times per session.

RESET causes all counts to be reset.

TN3270 tcp-client_pattern [RTM [RESET] | RTMALL [RESET]]

Displays any active TCP/IP clients that match the name, or name pattern entered in the command. Enter an asterisk to obtain details of all TCP/IP clients active.

The command returns details of each user, the profile being used, the terminal type and the host node IP address.

The RTM and RTMALL parameters cause statistics from the Response Time Monitor to be displayed for sessions with STATS=Yes specified. See 'Using the Monitoring Statistics' in the *Installation and Customization* manual for details of the Response Time Monitor routine.

RTM displays the total response time counts per session.

RTMALL displays all counts; that is, network, application and total response times per session.

RESET causes all counts to be reset.

User *userid_pattern* [*user_qualifier* RTM [RESET] | RTMALL [RESET]] [NONQual]

Displays information about an active user, or group of users.

Note The QUERY USER command is not suited to being issued from a script or an Assembler exit. See instead 'QUSER user command' on page 240.

The information displayed by the QUERY USER command includes the profile name, the terminal name, the sessions that are active and the session detail numbers. The VTAM application names, ACB names and logmodes are also displayed. Both SIGNON=Yes and SIGNON=No users are reported.

The Shared User facility enables multiple users to sign on with the same userid (see the *Installation and Customization* manual). A unique, or fully qualified, identifier is created internally for each user by appending a 'user qualifier' to the signon userid. Depending on the value of parameter MULTUSER on the SYSTEM statement, either the 'user qualifier' is created from the last four characters of the terminal LU name, or it is an eight-digit number in the range 1-99999999. For Session Manager 1.2.10 and higher, if multiple users are signed on with the same userid then you can include either a four-character or an eight-digit *user_qualifier* as appropriate with the 'QUERY USER' command to distinguish which user is required.

The RTM and RTMALL parameters cause statistics from the Response Time Monitor to be displayed for sessions with STATS=Yes specified. See 'Using the Monitoring Statistics' in the *Installation and Customization* manual for details of the Response Time Monitor routine.

RTM displays the total response time counts per session.

RTMALL displays all counts; that is, network, application and total response times per session.

RESET causes all counts to be reset.

NONQual, if specified, must precede the *userid_pattern* which should be the only other parameter specified. Details of only **non-qualified** users matching the pattern are returned.

QUIT basic command

QUIT [Exit|Signon|Logon]

On help panels, the QUIT command redisplays the previously displayed help panel. On panels displayed as a result of commands such as QUERY or DLOG, the QUIT command returns to the Menu panel. The optional operands have no effect in these instances.

On a Menu panel, or a Signon panel, the QUIT command is equivalent to the LOGOFF command, and one of the optional operands may be entered to display either the VTAM logon screen or the Session Manager Signon screen. The current LOGDISC setting from the configuration file determines which screen is shown when no operands are supplied with QUIT.

QUSER user command

QUSER *userid* [LU *lu_name*]

The QUSER command can be used to establish the status of a particular userid. When this command is issued, the response is a message which is returned in the t_message variable (see messages 341I, 371I, 394I, 396I, 399I and 400I in the *Messages* manual). This may be useful when it is necessary for a script or Assembler exit to establish the status of a particular userid.

The LU parameter is meaningful for multiple users, where SHARE, SHARESESS or SHAREDISC apply. If the parameter is specified for a normal user, it will be ignored.

When the LU parameter applies, the message response depends on the scenario:

1. If there is only one user signed on, then message 399I will be returned.
2. If there is more than one user signed on with the *userid*, and one user is signed on to the specified *lu_name*, then message 399I will be returned.
3. If there is more than one user signed on with the *userid*, and one user has disconnected but was last connected to the specified *lu_name*, then message 396I will be returned.
4. If none of the above scenarios applies, then message 371I will be returned.

RESET user command

RESet *seln-id*|*

The RESET command, when entered with an asterisk, causes any active sessions to be terminated. The terminal will remain connected to Session Manager. The selection id parameter may be the number of a session, a PF key number, or a selection command sequence. This enables a selective session termination to be achieved.

RETRIEVE basic command

`Retrieve`

The `RETRIEVE` command, when entered at a Session Manager panel, causes the previous command that was issued to be recalled and displayed in the command area. The command can then be modified and reissued.

You can use the `RETRCMDS` parameter of the `SYSTEM` statement to specify the number of commands that are to be stored for retrieval, up to a maximum of 30.

RETURN basic command

`RETURN`

The `RETURN` command may be entered at Session Manager display screens such as those invoked by `HELP`, `QUERY` or `DLOG`, and returns immediately to the screen at which the command was issued. It may not be issued at the Menu screen.

REVEAL user command

`REVEAL [PFkey|session_number|this_applid|ALL]`

The `REVEAL` command is used to dynamically display one or more previously concealed sessions on the user's menu (see 'CONCEAL user command' on page 223 and CONCEAL common session parameter on page 54).

If the parameter starts with 'PF' and the remainder is numeric 1-24, then it is treated as a PFkey number; if the parameter is all numeric 1-9999 it is treated as a session number; otherwise it is treated as an applid. When revealing sessions by their applid and you have concealed more than one session with the same applid then the first matching concealed session will be revealed. To reveal the other sessions with the same applid reenter the `REVEAL` command specifying the applid.

Sessions are only concealed for the duration of the user being signed on. If the user signs off then when they sign on again all the previously concealed sessions will be visible.

To permanently conceal a session set the common session parameter `CONCEAL` to `YES`.

`ALL` will reveal all concealed sessions.

See also 'CONCEAL user command' on page 223.

RIGHT user command

`RIGHT [`*`nn`*`]`

The `RIGHT` command moves the display area to the right on a panel which is wider than the display area. The default number of columns to move is 1. *nn* may be any value in the range 1-99.

SE user command

`SE`

The `SE` command is used in the Cut operation to define the south east corner of the cut area. The south east corner is the bottom right-hand corner of the area, or the end of the cut.

`CUTEND` is a synonym of the `SE` command.

SEND user command

Note Default security authorization level: '5'.

`SEND` *`nodename command`*

The `SEND` command is used to send commands to another remote Session Manager system, which is connected to the local node by the Session Manager network.

The *`nodename`* given is the remote Session Manager node name. This name must match the node name given on the `LOCALNODE` parameter for that remote node. *`nodename`* may be up to eight characters in length.

`command` is the command to be sent. If the command contains spaces, for example, QUERY ISZSMGR, then it must be enclosed in delimiters. For example:

SEND node01 'Query ISZSMGR'

If any of the command parameters contain spaces then these too must be enclosed in a different set of delimiters. The command may be up to a maximum of 256 characters in length. Valid delimiters are:

'this phrase is delimited'

"this phrase is delimited"

(this phrase is delimited)

!this phrase is delimited!

?this phrase is delimited?

If the Session Manager systems are networked using `LINK` statements then the command is executed for the user on the remote node specified, with an authority level of 0, unless an appropriate `RUSER` statement has been defined in the remote Session Manager system. In this case, the authority level is taken from the `RUSER` definition. The `RUSER` must specify the originating nodename and either the remote user name or a matching generic user name.

If the Session Manager systems are networked using Session Manager's Sysplex networking then the command is executed for the user on the remote node specified, with the authority level of the user, unless an appropriate `RUSER` statement has been defined in the remote Session Manager system. In that case, the authority level is taken from the `RUSER` definition. The `RUSER` must specify the originating node name and either the remote user name or a matching generic user name.

The commands that are valid for use with the `SEND` command: are shown in 'Table of commands' on page 217. Command scripts may be invoked. However, if ISZCMD is invoked with more than one full screen command, then only the first is processed; subsequent ones are ignored.

SME user command

SME

This command, which operates only from the Menu, sends the Menu escape key (or sequence) to the last active session. When using this command, the time interval for 'double escape' processing (see 'DOUBLESC' on page 36) must be non-zero, and the last activity at the terminal must be a menu escape from an application session.

SPYOFF user command

SPYOFF

The SPYOFF command can be issued by a user who is the target of a spy operation, and has the SPYABLE TELL option set. It causes the current spy session to end. The command is only accepted when SPYABLE TELL is in effect.

STARTSC user command

STARTSC Yes|No|ON|OFF *seln-id*|*

The STARTSC command can be issued by a suitably authorized user to disable or (re-)enable the running of the session start script specified using the common session parameter STARTSCRIPT (see page 69). Both parameters are mandatory. (Clearly, if a session start script has not been specified in the configuration then the STARTSC command will have no effect.) The command does not work against hidden sessions.

STARTSC=No or OFF disables running of the script; STARTSC=Yes or ON (re-)enables running of the script. The new session variable s_runstartsc will be set appropriately – that is, Y for Yes|ON and N for No|OFF. For more information, see the *Panels, Scripts and Variables* manual.

seln-id can be the number of the session, a selection command sequence (only one such sequence on a session is supported), or a PF key number.

STARTSC * disables the running of the start script for all sessions associated with the Menu screen.

TOP basic command

TOP

The TOP command may be used on multi-page panels to move to the top of the panel.

Note Header and trailer sections of a panel are not pageable, these remain static.

TRANSFER user command

TRANSFER [Override]

The TRANSFER command may be entered at a Session Manager Signon panel in order to acquire the Menu screen, and all active sessions, of a user signed on at another terminal. The user id and password must be entered in the user and password fields of the signon screen of the acquiring terminal.

The Menu screen being acquired must have been defined in the configuration file as MOBILE and must have 'unlocked' status. The lock status may be altered using the LOCK command.

If the acquiring terminal and the terminal that is being acquired are incompatible, a message is issued at the acquiring terminal and the transfer is not performed. The OVERRIDE operand may be used to force the acquisition, but it is then the responsibility of the user to overcome any misaligned displays. Sessions displayed using the Windows feature will not be misaligned.

UP basic command

Up [*nnnn*]

The UP command may be used on multi-page panels to move up the pageable portion by the number of lines specified on the command. The default is one line if a number is not specified. *nnnn* may be any value in the range 1-9999.

Note Header and trailer sections of a panel are not pageable, these remain static.

VIEW user command

VIEW *demonstrator-id* [PASS *password*] [Override]

The VIEW command allows a user to join a demonstration. It is only valid when issued from the Menu screen.

demonstrator-id

This is the userid of the user who is giving the demonstration. That is, the person who entered the DEMO command. This parameter must be entered.

PASS *password*

This is the demonstration password. If a password was set by the demonstrator, then it must be entered with the VIEW command to gain access to the demonstration.

Override

Indicates that the check for compatibility between the demonstrator's terminal and that of the user who entered the VIEW command is not to be performed, and the user should be allowed to view the demonstration regardless. The results, however, may be unpredictable.

Note If this option is omitted, a user entering the VIEW command with a different terminal type to the demonstrator will either be denied access to the demonstration, or will have less terminal capabilities than the demonstrator; for example, no highlighting/colors, and so on.

WINDOWS user command

Windows [*script-name*]

The WINDOWS command starts the window script specified by the *script-name* parameter. If no *script-name* is specified, the *script-name* specified on the WINDSCRIPT parameter in the configuration file is used (see page 50).

Privileged user commands

(These commands should not usually be issued by non-privileged users. The commands can be entered by a user logging on to the system console or from any terminal connected to Session Manager whenever a screen has an 'Enter Command' field.

To enter a command from a z/OS system console, the standard 'modify' command is used; for example, 'F ISZSMGR'. The system console is treated as a user with the highest authorization level of '9'.

BLOCK operator command

Note Default security authorization level: '5'.

`BLock applid [Yes|No|ON|OFF] [EXclude groupname]`

The `BLOCK` command may be used to prevent further logons to the specified application until such time as a `BLOCK applid OFF` or `BLOCK applid No` command is issued.

applid can either be the VTAM applid or it can be the dummy application name specified on the `APPL` statement.

Note You cannot issue multiple `BLOCK` commands against a single VTAM applid. The **last** `BLOCK` command issued for the applid is the only one that is executed. Use the `APPL` name instead.

For example:

```
APPL FRED APPLID S05TSO
APPL HARRY APPLID S05TSO
```

If a `BLOCK FRED` command is issued, users of `APPL HARRY` can continue to start sessions and you can also start sessions with S05TSO directly. If S05TSOVis blocked, **all** users are blocked from accessing any applications that have an `APPL` statement using that applid.

The `EXCLUDE` parameter allows a group of users, profiles or terminals to still have access to the application. The users who are blocked have the Menu display automatically updated to reflect the new status. Users who are in a group that is excluded are able to access the application as normal. Multiple `BLOCK` commands can be issued to exclude multiple groups. Refer to the `GROUP` statement in this manual for details on setting up groups.

A blocked application may be indicated on the Menu panel by either of the Session Manager variables, s_aa or s_a, which have default values of (B) and 20 respectively for a blocked application. These values may be altered by the `MESSAGE` statement in the configuration file.

BROADCAST administrator command

Note The Broadcast command panel, which provides an interface for entering the parameters shown below, can be accessed directly from the Session Manager menu in the same way as any other session. The parameters will be displayed and can be modified by the user before the command is issued. By default, the BROADCAST command can only be issued by users with a security authorization level of 9.

```
BRoadcast message-text
          ALL                     |
          Applid appl_pattern     |
          BRDVAR id_pattern       |
          Group group_name        |
          Lu luname_pattern       |
          Profile profile_pattern |
          User user_pattern
               [ [Hold hh:mm | [FOR hh:mm|days|days hh:mm] ]
          [URGent]
```

The BROADCAST command sends a message to one or more terminal users. The length of the message may be 256 characters *minus* the length of the command and any parameters, so the effective limit is approximately 240 characters. The message text should be enclosed in quotes when the text contains embedded spaces. The message interrupts the recipient's current session, unless a receiving status has been set by the BRECEIVE command to prevent interruption, or the user is running a Windows script that cannot receive messages. If PCTRANSFER has been set then broadcasts are queued, regardless of the BRECEIVE setting.

The destination(s) may be specified using one of the following parameters:

ALL

All terminals logged on to Session Manager.

Applid *appl_pattern*

All Session Manager users logged on to VTAM applids, or APPL statement names, which conform to the specified pattern.

BRDVAR *id_pattern*

All users with a session which has been allocated an id-string which conforms to the specified pattern. The session may be active or inactive, and so the command can be used to send a message to users of an abended application when it is has been restarted.

Group *group_name*

All users who are members of a group with a matching group name.

Lu *luname_pattern*

All terminals whose LU names conform to the specified pattern.

Profile *profile_pattern*

All terminals or users using PROFILEs which conform to the specified pattern.

User *user_pattern*

All users whose ids conform to the specified pattern.

The main difference between this command and MSG is in the capability to define destinations using a pattern rather than a specific name. A pattern is a generic name where a plus symbol ('+') indicates one of any character, and an asterisk ('*') indicates any number of any character. For example:

BR 'Good Morning' U ++AA* HOLD 9:30

This command displays the 'Good Morning' message on the screens of all users currently signed on who have 'AA' as the third and fourth characters of their userid.

The Hold parameter causes the retention of the message until the specified time, which may be up to 24 hours from the current time. If the destination is a user or a terminal, once the message has been displayed at that destination during the signon procedure it is not displayed again at subsequent signons. However, if the destination is a profile, or a group which may be used by a number of different users, then a user signing on several times will be presented with the message at each signon.

FOR enables a message to be held for a relative period of time. The message is sent to the destination when that destination logs on, as long as the time period has not expired. The FOR and HOLD parameters cannot be specified together.

URGENT overrides any inhibiting BRECEIVE setting in order to display the message immediately at the destination terminals. This parameter should be used with consideration, since if used for trivial messages, a certain amount of irritation may easily result.

If you are using a Sysplex

When operating with the SYSPLEXGROUP keyword specified on the SYSTEM statement, by default all messages and broadcasts sent with the MSG user command and the BROADCAST operator command will be distributed across the Sysplex. For example if a BROADCAST command was issued in one Session Manager instance the command would be sent to all other Session Manager instances in the same Sysplex group. This global scope of the Session Manager message/broadcast commands can be changed by setting the variable t_global_msg. A read-only variable, t_global_msgdef, reflects the value set by (SYSTEM statement) parameter *SYSPLEXGroup GLOBALMessages*.

If a message has been restricted in scope to the issuing Session Manager instance (by use of variable t_global_msg or the GLOBALMESSAGES sub-parameter of the SYSPLEXGROUP) then only the issuing Session Manager instance will process it.

Consequently, regardless of which Session Manager instance a QUERY Broadcast is issued on, all held messages will be displayed irrespective of each message's scope.

Each held message is displayed as *number qualifier-text*,

where.

number is the message number,

text is the message text,

qualifier is either blank or displays one or more of the following special characteristics:

<U><G><,U=*xxxx*><,*yyyyy*>

where:

U indicates that the message was designated as urgent.

G indicates that the message was sent to a generic target.

U=*xxxx* (where *xxxx* is the user id) indicates that the message was queued after a user had received it then exited from the message display panel without acknowledging the message (as a result of expiry of the IDLELOGOFF time, for example).

yyyy (where *yyyy* is relevant Session Manager node name) indicates that the message was restricted to a Session Manager instance in a Sysplex environment (SYSPLEXGROUP parameter has been set on SYSTEM statement).

Incidentally, this approach enables a newly started Session Manager instance to have access to all held messages previously generated. Note that if all Session Manager instances in a Sysplex group are closed then the held messages are lost. However, provided at least one Session Manager instance is active, the held messages will be maintained.

The QUERY Broadcast display indicates the scope of messages that have been restricted to one Session Manager instance.

CLOSEDOWN administrator command

Note Default security authorization level: '9'.

CLOsedown [FORce|END]

The CLOSEDOWN command schedules a shutdown of Session Manager by one of two methods.

FORCE – All sessions are terminated immediately. This is the default parameter.

END – All applications are 'blocked' against further logons, ENDSCRIPTS are run on all sessions immediately, and when all have completed, the system is shut down.

Note The END option does not cause the immediate shutdown of the console session or of terminal sessions, so Administrators can continue to issue commands while the shutdown progresses.

The TERMINATE command is a synonym of the CLOSEDOWN command.

DELETE BROADCAST administrator command

Note Default security authorization level: '9'.

DELETE BROadcast *message_num*

A message sent using either the BROADCAST or the MSG command may have a time specified for which the message is held in the system. The authority of a user deleting a message must be equal to or greater than the originator of the message. The message number may be obtained from either the Session Manager Audit file, or by issuing the QUERY BROADCAST command.

If you are using a Sysplex

When operating with the SYSPLEXGROUP keyword specified on the SYSTEM statement, the DELETE BROADCAST and DELETE MSG commands will delete the message from all Session Manager instances.

DELETE MSG administrator command

DELETE MSG *nnn*

This command is a synonym of the DELETE BROADCAST command, which is described above.

If you are using a Sysplex

When operating with the SYSPLEXGROUP keyword specified on the SYSTEM statement, the DELETE BROADCAST and DELETE MSG commands will delete the message from all Session Manager instances.

DEMO operator command

Note Default security authorization level: '5'.

```
DEMO [ [PASS password|NOPASS]
       [AUTocopy [Yes|No|ON|OFF]]
       [INTernal [Yes|No|ON|OFF]]
       [DISplay [Yes|No|ON|OFF]]
       [STOP] ]
```

The DEMO command allows an authorized user to begin a demonstration, change the demonstration parameters for a demonstration, switch between the Demonstration Viewer list and the Menu screen, or end the demonstration.

The DEMO command is only valid when issued from the Menu screen or the Demonstration Viewer list.

PASS *password*|NOPASS

PASS *password* sets the demonstration password which must be entered by viewers wishing to join the demonstration. The password can be up to eight alphanumeric characters in length. NOPASS, which is the default for a new demonstration, indicates that no password is required.

AUTocopy [Yes|No|ON|OFF]

Determines whether outputs to the demonstrator's screen should be automatically sent to the viewers of the demonstration. If AUTOCOPY=Yes is specified, outputs are automatically sent to the viewers. If AUTOCOPY=No is specified, only those output screens chosen by the demonstrator (via the demokey) are sent to the viewers. The default for a new demonstration is AUTOCOPY=Yes.

`INTernal [Yes|No|ON|OFF]`

Determines whether Session Manager internal screens (such as the Demonstration Viewer list) are to be included when automatic screen copying is in effect (see the `AUTOCOPY` parameter). If `INTERNAL=Yes` is specified, or if the `INTERNAL` parameter is specified with no subparameter, Session Manager internal screens are included. If `INTERNAL=No` is specified, internal screens are not included. The default for a new demonstration is `INTERNAL=No`.

`DISplay [Yes|No|ON|OFF]`

Determines whether the Demonstration Viewer list (panel DEMO) is to be displayed following completion of the `DEMO` command. If `DISPLAY=Yes` is specified, or if the `DISPLAY` parameter is specified with no subparameter, the Demonstration Viewer list is displayed. If `DISPLAY=No` is specified, display of this panel is suppressed. If `DISPLAY=No` is specified while the Demonstration Viewer list is being displayed, the Menu screen is re-displayed, without terminating the demonstration.

`STOP`

Terminates the demonstration. When specified, it must be the only parameter on the `DEMO` command.

DLOG operator command

Note Default security authorization level: '5'.

`DLog`

The `DLOG` command displays the messages most recently written to the Session Manager Audit file. Paging commands, `UP`, `DOWN`, `FWD` and `BWD`, may be used to scroll through the messages. Other commands may also be issued from the display.

The `QUIT` and `RETURN` commands cause the Menu panel to be displayed.

DSTORE administrator command

Note Default security authorization level: '9'.

`DStore [sub-command]`

The `DSTORE` command may be used to examine and alter storage in the Session Manager system. It places the terminal in a completely separate display mode, temporarily suspending the Menu panel and any associated sessions currently active at the terminal.

It is advisable to use this command only when requested by your local support representative and to restrict access to only a few users.

When entered without a sub-command, the initial screen shows the current user block. From left to right across the screen are the absolute address; the relative address; 16 bytes of storage in hex; and finally the 16 bytes of storage translated into character format. (Note these storage figures are for model 2, 3, 4, x screens; model 5 screens display 32 bytes of storage). Once storage display mode has been entered, the commands listed below can be used to directly access various parts of Session Manager storage.

The storage display commands can be entered as a sub-command of the `DSTORE` command to access the required part of Session Manager storage directly. Storage display commands should be entered in the command line of the screen display, or can be issued as PF key commands. The available PF key commands are displayed at the bottom of the screen.

Any storage display command can be prefixed by a 'retain' character. This causes the command to remain in the input area after it has been actioned. This can prove useful if a command is to be issued repeatedly. The repeat character is an ampersand '&'.

Storage display commands can be concatenated by separating them by a concatenation character, the hash '#'' symbol. (Hexadecimal 7B). In this way, several storage display commands can be entered as a string, which are processed sequentially. The following commands may **not** be concatenated: Alter, Quit and Help.

The storage display commands, *sub-command*, are:

`+xx|-xx`

Move the display forwards by '*xx*' bytes by specifying '`+xx`'. Alternatively, move the display backwards by '*xx*' bytes by specifying '`-xx`', where '*xx*' is a hexadecimal number.

address

Display storage from the given hexadecimal *address*.

```
ALTER xxx ... xxx
ALTER 'ccc ... ccc
```

Replace the storage at the top of the display with the hexadecimal or character bytes given as the operand. A maximum of 32 bytes may be changed, but the command area may be shorter than this, dependent on the terminal model being used. (For example, the input area cannot contain 64 hex characters on any 80 column display).

For a storage alteration in character format, the starting apostrophe must be specified, and all significant characters including any keyed trailing spaces are placed into storage. Mixed case characters are stored in mixed case; if upper case characters are required, upper case characters must be keyed.

The closing apostrophe is optional and denotes the end of the keyed character string: any characters keyed after an apostrophe are ignored. Note, however, that the rightmost apostrophe is the delimiting character when more than a single apostrophe is entered in a character string.

For example, ALTER 'ABC and ALTER 'ABC' place ABC into storage, as does ALTER 'ABC'ABC (since in this case the trailing ABC is ignored). However, ALTER 'ABC'ABC' causes ABC'ABC to be placed in storage since the rightmost apostrophe is taken as the delimiter.

Using the trailing apostrophe therefore provides a method of placing an apostrophe in storage.

`BACK [nn]`

Move the storage display backwards by a full screen, or by '*nn*' screens.

`DOwn [nn]`

Move the storage display forwards by a single line, or by '*nn*' lines.

`FOrward [nn]`

Move the display forwards by a full screen, or by '*nn*' screens.

`Help`

Display help on the commands available.

`Note notename`

Give the address at the top of the display the note given by '*notename*' which can be from one to eight characters long. If a note 'notename' already exists, it is deleted, and '*notename*' inserted at the new address. The first fullword on the line against which the note is entered is highlighted.

When there is more than one note on the same line of the display, the one associated with the lowest address on the line is shown on the right hand side of the display. All fullwords on the line for which notes exist are displayed in high brightness.

See also the *notename* command below.

`DELete notename`

Delete the previously entered note *notename.*

notename

Display storage from the address associated with the previously entered note *notename.*

`Offset [+|-] [xxxx]`

Display storage from the current origin. A hexadecimal value '*xxxx*' may be added to, or subtracted from the origin prior to the storage being displayed. See also the `ORIGIN` command below.

`ORIGin [*|address|notename]`

Set or reset the origin value. The origin is the base address from which the offset column on the display is calculated.

The Asterisk (*) represents the address the display currently starts from. When the asterisk is entered this address is displayed as the Origin. The offset column is recalculated from this base address. When the command is entered with no operands it has the same effect as specifying `ORIGIN *`.

'*address*' may be any address in the range 0-7FFFFFFF on XA systems, and 0-FFFFFF on non-XA systems. If the storage is outside the bounds of the application, a blank screen is displayed.

'*notename*' may be any note, either predefined or keyed by the user. The origin value is set to the address associated with that note.

`Point`

Display storage from the address contained in the first four bytes of the display.

`POINTH|PH`

Display storage from the address contained in the first two bytes of the display.

`POINTV|PV`

This command means 'Point via Fullword'. The address in the first four bytes of the display is added to the current address, and storage is displayed from there.

`POINTVH|PVH`

This command means 'Point via Halfword'. The address in the first two bytes of the display is added to the current address, and storage is displayed from there.

`Quit|QQuit`

Exit from Storage Display Mode and return to the Session Manager.

`Up [nn]`

Moves the display backwards by a single line, or by '*nn*' lines.

As well as the previous commands, there are special commands which request the display of certain blocks of storage. These are:

`SYSB`

Display the system block.

`CMAP|ISZMAP`

Display the program map.

csect-name

Display the address of a csect name. *csect-name* is the name to be displayed.

If you do not know the csect name you require, use CMAP or ZEDMAP to find the address in the list presented.

`USEB`

Display the current user block.

`TTAB`

Display the current user internal trace block, when internal tracing is on.

`LU` *luname*

Display storage associated with the *luname*.

`ZEDAPIB`

Display the program interface block.

`ZEDMAP`

Display the second program map.

`ZPSMAP`

Display the third program map.

DTERM administrator command

Note Default security authorization level: '9'.

`DTerm ['LU luname']`

Control blocks relating to the physical terminal can be displayed using this command. It would normally only be used under the guidance of your local support representative.

Single quotes must be used. When no *luname* is specified, storage is displayed for the terminal from which the command is issued.

The parts of storage initially displayed are:

- Device Information Block
- Read Partition Query Reply Buffer Block
- VTAM Request Unit Buffer Block

In the display mode, the commands available are:

`Help`

To invoke help for DTerm.

`Quit|QQuit`

To exit from the display mode.

`LU luname`

To display the device information blocks for the given luname.

PF keys are available to display device information for other terminals, by paging forwards or backwards. If the information for a terminal is contained on more than one screen, pressing Enter causes the display to page forwards. PF keys can also be used to display other control blocks. The Query Reply Structured fields may be accessed by PF key, in addition to the session parameters, the logmode name in use, and the primary and secondary lunames.

The functions assigned to each PF key are as follows:

`PF1`

Invokes help for `DTerm`.

`PF3`

Quits out of the display mode to the application.

`PF4`

Shows fields from the Request Unit buffer, these are:

- The session parameters
- The logmode name
- The primary LU name
- The secondary LU name

`PF5`

Shows the Query Reply structured fields that are currently available.

`PF6`

Returns to the device information block display if another type of display is showing. If a device information block is already displayed then PF6 shows the device block for the current terminal.

`PF7`

Shows the device information block for the previous terminal.

`PF8`

Shows the device information block for the next terminal in the chain. The sequence of the chain depends on the sequence that users sign on.

PF10, 11, and 12 are only applicable to the current terminal display. They issue RPQ commands if the appropriate Reply buffers are not available.

`PF10`

Issues a Read Partition Query.

`PF11`

Issues a Read Partition Query List.

`PF12`

Issues a Read Partition Query All.

PA1 and PA2 alter the screen size. PA1 switches to the alternate size if one is available, PA2 switches to the default size. When a display extends for more than a screen, pressing Enter displays the next screen.

DUMP administrator command

Note Default security authorization level: '9'.

`DUMP`

The `DUMP` command writes the whole of Session Manager storage to the standard dump file. No other work is processed by Session Manager while the dump is being taken. This is a diagnostic facility and should only to be used at the request of your local support representative.

FLASH operator command

Note Default security authorization level: '5'.

`FLash` *`message-text`* `[Applid` *`applid`* `| Variable` *`global-variable`*`]`

The `FLASH` command can be used to update the flash area. There are two types of flash area, the system flash area and the session flash area. The two areas are referenced by the variables *`gcflash`* and *`s_flash`* respectively and may be included in PANEL definitions, the Menu and Signon panels being the most useful.

When the `FLASH` command is issued without any optional operands, the system flash area is updated.

To update a flash area associated with a session, the applid for the session must be specified. This can be the VTAM or real applid, or the `APPL` statement name. The `FLASH` command is the only method for updating *`s_flash`* variables, and *`s_flash`* variables throughout the whole system are updated. This is particularly useful for informing users of session status.

Any message text containing embedded spaces must be enclosed in special delimiting characters, which may be single quotes, double quotes, exclamation marks, question marks, or paired parentheses. This also enables special characters to be embedded in the text. See the examples on page 256.

The message remains current until it is altered, or cleared. A flash area is cleared by specifying the message text as a blank, for example:

FL ' ' A HARRY

clears the flash message area for all sessions using the APPL statement name HARRY.

In addition to the system and session flash areas, the FLASH command may be used to update GC prefix variables. These are user defined global character variables. The GC prefix must be supplied.

Examples of use are:

FL ‘System will be available from 7 am tomorrow’

FL (Not available until 2 o’clock) A CICSGEN

FL ‘Red Lion this evening’ v gcvenue

The first example puts the message in the system flash area, this is the variable *`gcflash`*. The second example places a message announcing the availability of the CICSGEN application in the session flash area for CICSGEN. The third example updates the user defined global variable *`gcvenue`*.

The flash variable is not displayed on the OLA (Online Administration) and the VTM (Virtual Terminal Masking) internal sessions.

FORCE administrator command

Note Default security authorization level: ‘9’.

`FORCE` *`taskname`* `|` `LINK` *`linkname`*

The `FORCE` command purges a Session Manager task from the system and is useful when a task appears to have stopped.

The *`taskname`* is the first word of the task id, for example, SAA01347.

The `STOP USER ALL` command should be used before issuing this command. Issuing the `FORCE` command without the advice of your local support representative, may lead to unexpected results.

Under no circumstances should a task which does not appear in a QT U display be forced.

If a `STOPLINK` command has been issued for a link and it fails to respond, the `FORCE LINK` command can be used to terminate the link task. *`linkname`* is the name of the link to be affected.

GFS administrator command

Note Default security authorization level: '9'.

```
GFS STATS|USAGE|STOR
```

The GFS command should only be used at the request of your local support representative. It is used to obtain statistics regarding the operation of GFS in the Installation's operating environment.

STATS displays a panel that provides statistics about the number of GET-storage requests handled by GFS and their relative success.

USAGE displays a panel that indicates the amount of storage managed by GFS and the percentage of free storage available for various sizes of GET requests.

STOR displays a panel giving the layout of storage managed by GFS, that is, the address and size of the storage and bitmaps used by GFS.

Note The GFS command can only be used on a system that has GFS active. If issued on a system that does not have GFS active, then message 559I GFS IS NOT ACTIVE - COMMAND REJECTED is issued.

Information and samples of the GFS displays is provided in 'Problem diagnosis and reporting' on page 273.

INQUIRE administrator command

Note Default security authorization level: '9'.

The INQUIRE command is used for ad hoc inquiries, that is, to perform VTAM inquiries immediately, and has no operands. (VTAM inquiries are normally performed under the control of the INQINTERVAL parameter on the SYSTEM statement – see page 96).

ISZECLP administrator command

Note Default security authorization level: '9'.

The ISZECLP command is only supplied to set an AUTH level for use of the Eclipse interface. When a user attempts to sign on via the Eclipse interface, the user's authority is compared to the authority level of the ISZECLP command, and if equal to or higher, the user is allowed to log on (assuming any subsequent security checks are passed). The authority level on the ISZECLP command may need to be changed depending on your installation's requirement. To change the default level see the 'COMMAND statement' on page 157. The user will receive message ISZ4297E if their authority is lower than the level specified for the ISZECLP command.

ISZTEST administrator command

Note Default security authorization level: '9'.

```
ISZtest [Yes|No|ON|OFF]
```

The ISZTEST command is used to switch on and off certain diagnostic processes. The configuration OPTION TEST parameter has the same effect. ON is the default when no parameters are supplied.

ISZTEST should only be used at the request of your local support representative.

PASSFREE administrator command

Note Default security authorization level: '9'.

`PASSFREE LU luname`

The `PASSFREE` command unlocks a terminal that has been locked by Session Manager due to a signon password being entered incorrectly more times than is allowed by the `PASSTRY` parameter setting on the `TERMINAL` statement.

Message `5Ø3I` gives the terminal type and name which should be specified when this command is issued.

PLAYDS operator command

Note Default security authorization level: '5'.

`PLAYDs index-number`

The `PLAYDS` command can be used on the `REPLAY` index display to select one of the listed screens for display. The screen is formatted as 3270 data stream orders. *`index-number`* is the number assigned to the screen in the `REPLAY` index.

See also the `PLAYIMAGE` and `PLAYHEX` commands.

Note If the supplied `REPLAY` display panel is used, this command is already assigned to a PF key.

PLAYHEX operator command

Note Default security authorization level: '5'.

`PLAYHex index-number`

The `PLAYHEX` command can be used on the REPLAY index display to select one of the listed screens for display. The screen is formatted as a hex dump of the 3270 data stream. *`index-number`* is the number assigned to the screen in the REPLAY index.

See also the `PLAYIMAGE` and `PLAYDS` commands.

Note If the supplied REPLAY display panel is used, this command is already assigned to a PF key.

PLAYIMAGE operator command

Note Default security authorization level: '5'.

`PLAYImage index-number`

The `PLAYIMAGE` command can be used on the REPLAY index display to select one of the listed screens for display. The screen is displayed as it appeared when it was first RECORDed. *`index-number`* is the number assigned to the screen in the REPLAY index.

See also the `PLAYHEX` and `PLAYDS` commands.

Note If the supplied REPLAY display panel is used, this command is already assigned to a PF key.

PUPDATE administrator command

The PUPDATE command enables the Session Manager environment to be changed without the need to close and restart Session Manager.

Although members and statements can be deleted from the configuration files using Online Administration, it is not possible to delete statements from the active Session Manager environment using the PUPDATE facility; the only definitions that are removed are those which are replaced by new definitions with a duplicate name. Statement definitions for which there is no matching name are included as entirely new statements. Any errors cause the new or replacement statement to be ignored and, if it is a duplicate, the original definition is retained. Statements in error are printed with an appropriate error message.

The original definitions are not deleted from the environment while they are currently in use, but new definitions become current the next time they are used subsequent to the PUPDATE command being issued. This means that two versions of a particular definition may exist for a time. If there are multiple versions of a definition, then only when there are no users using the old version, is it deleted but if there is only one version of a definition this will not be deleted from program storage until Session Manager is recycled.

For example, a new profile comes into effect when a user signs on, but any users signed on with active details at the time of the update continue to use the old copy of the profile. For these users, the new profile is put into effect as soon as they have no active sessions. User modified screen variables are merged with the new profile where appropriate.

Examples of how to disable statements on an active system:

1. Effectively deleting a USER statement on an active IBM Session Manager.

   Redefine the USER statement with no options other than a PROF parameter that points to a new dummy profile. The dummy profile has no options in it, for example:

   ```
   USER joe PROF nullprof
   PROFILE nullprof
   ```

   You could also set up a dummy menu that simply displays an appropriate message.

   Note Normally, there is no requirement to delete a user, since the user would be disabled in the external security system and therefore have no access to Session Manager.

2. Effectively deleting a GROUP statement on an active Session Manager.

   Since a valid GROUP statement must contain at least one object in the group, you could redefine the GROUP with an LUNAMES referring to a nonexistent terminal name, for example:

   ```
   GROUP grpa LUNAMES xxxxxxxx
   ```

3. Effectively deleting an APPL statement on an active Session Manager.

   Redefine an APPL by adding an INITSCRIPT that issues an ENDSESS to terminate any selected session associated with the APPL, for example:

```
APPL appl1 INITSCRIPT notavail DESC “This application is no
longer available”
SCRIPT notavail
Let t_message = s_appl “ is no longer available”
Endsess
```

Notes

1. Default security authorization level: ‘9’.
2. This command is relevant only if you are using Online and/or Batch Administration to tailor Session Manager. If you are *not* using Online and/or Batch Administration, see instead ‘UPDATE administrator command’ on page 270.

`PUPdate` *`ddname member-name`*

If you are using Online and/or Batch Administration to tailor the product, configuration data is stored in several PDS(E)s. In this configuration, each PDS(E) is allocated to a particular `DDNAME` and *must be maintained exclusively by Online and/or Batch Administration*. The `PUPDATE` command enables components of the configuration to be changed without the need to close and restart Session Manager.

Note To update the particular definition(s) using `PUPDATE`, the user must have the appropriate OLA security class.

`ddname`

Specifies a `DDNAME` to be processed. For a list of possible `DDNAME`s, see the *Installation and Customization* manual.

`member-name`

Supplies the name of the member to be updated and may be any valid combination of alphanumeric characters up to a maximum of eight.

QTASK administrator command

Note Default security authorization level: ‘9’.

```
QTask [Task taskname       |
       User userid_pattern |
       LU luname_pattern   |
       [Selection seln-id] ]
```

The `QTASK` command is a special form of query command which provides information about Session Manager tasks.

When entered without operands at a terminal, the QTASK command reports on all tasks associated with that terminal.

Alternatively, a specific task, or group of tasks may be identified using the operands as follows:

`Task` *`taskname`*

Where *`taskname`* is the first word of the task id for example, UAA00042.

`User` *`userid_pattern`*

Identifies the group of tasks associated with a particular user, or the groups of tasks associated with a set of users when the userid pattern is specified generically.

For example:

QT U +L*

All tasks associated with any users with 'L' as the second character are displayed.

`LU` *`luname_pattern`*

Identifies the group of tasks associated with a particular terminal, or the groups of tasks associated with a set of terminals when specified generically.

`User` or `LU` may be qualified with the additional parameter, 'Selection', as follows:

`Selection` *`seln-id`*

where *'`seln-id`'* may be the number of a session, or a selection command sequence.

RECORD operator command

Note Default security authorization level: '5'.

`RECOrd [Yes|No|ON|OFF|TERM|APPL|CLEAR|0]`

The `RECORD` command is used to start and stop recording terminal or application input and output for the user. Each input and output is saved in a record in Session Manager storage and can be viewed using the `REPLAY` command described in the following section. All records for a user are deleted when `RECORD 0` or `RECORD CLEAR` is issued and when the user logs off from Session Manager

When recording an application, both input and output data streams are recorded, but if the terminal is put into spy or view mode only outputs are recorded since the recording terminal 'sees' no inputs.

Detailed instructions for recording terminal input and output are given in 'The Record-Replay Facility' of the *User and Administrator* manual.

REMOVEUSER administrator command

Note Default security authorization level: '9'.

`REMOVEUSER` *`userid`*

The `REMOVEUSER` command logically removes the user specified by *`userid`* from the Session Manager system.

This command cannot be used when the shared userid facility is active. For details of this facility, see the *Installation and Customization* manual.

CAUTION `REMOVEUSER` should only be used at the request of your local support representative.

This command has a default security authorization level of '9'.

REPLAY operator command

Note Default security authorization level: '5'.

`REPLay`

The `REPLAY` command displays the Replay Index screen. The index lists all the terminal input and output records available to the menu, and identifies each by a number and a code to indicate whether it is an input (I), output (O), setup (S), partial setup (P), or escape (E) record. Replay may be started from any record, simply by entering the starting record number. The records may then be paged forwards and backwards using the PA2 and PA1 keys.

There are three formats in which records may be displayed, 'As Recorded', Dump, and Data Stream Analysis, although non-technical users would probably only require the 'As Recorded' format.

Both the `RECORD` and `REPLAY` commands may be assigned to PF keys so that a single keystroke may invoke either command.

Support for the Record-Replay Facility is generated by the `RECORDLIMIT` and `REPLAY` parameters of the `SYSTEM`, `PROFILE`, `USER` or `TERMINAL` statements in the Session Manager configuration file.

SECFRESH administrator command

Note Default security authorization level: '9'.

`SECFRESH`

This command deletes and then recreates the in-storage security profiles. The effect is to update the in-storage profiles with the latest version of the security database.

This command relates to the Dynamic Menus facility (see the *Installation and Customization* manual).

SPIN administrator command

Note Default security authorization level: '9'.

`SPIN AUdit|Trace|AUDITGDG`

The `SPIN AUDIT` and `SPIN TRACE` commands close and re-open the audit file, or the trace print file for DATA, MISER, VTAM or NETDATA traces, so enabling previous output to be printed. The `AUDITROUTE` or `TRACEROUTE` configuration statements determine the route for the output.

If the system has been configured to write the Audit to a GDG dataset then the `SPIN AUDITGDG` command closes the current GDG dataset and allocates, catalogues and opens the next (+1) generation without having to re-start the system.

SPY operator command

Note Default security authorization level: '5'.

```
SPY User userid |
    LU luname
    [NOAUTO] [Override]
```

The SPY command enables a user to simultaneously view the screen contents currently displayed at the terminal of another Session Manager user. By default, the spied screen image is automatically updated when the screen display of the user being spied upon is updated. Every I/O to the terminal of the user being spied upon is automatically reflected on the terminal of the 'spier'. If, however, the optional NOAUTO operand is used, the automatic screen refreshing does not take place, and the screen has to be updated manually using the Enter key.

The spy operation is terminated when the user who is spying enters one of the escape commands or when the user who is the target of the spy operation enters the SPYOFF command.

User *userid*

Identifies the user to be spied upon.

The User option is mutually exclusive to the various shared *userid* common enduser parameters SHARE, SHAREDISC and SHARESESS. If sharing userids, specify the user's LU instead.

LU *luname*

Identifies the terminal to be spied upon.

If the target of the spy operation is running at an incompatible terminal type, for example when a spy is attempted from a model 2 to a model 5, the spy operation is rejected. This can be overcome by issuing the SPY command with the OVERRIDE parameter, however the resulting display may be mis-aligned. The SPY command may be issued from within the Windows environment to avoid any mis-alignment.

There are several other situations in which this command will not be processed successfully as follows:

- If the target menu is inactive.
- If the target menu is already spying on another menu If the target menu has 'SPYABLE OFF' specified.
- If the target menu does not appear in the SPYGROUP for the user issuing the command.
- If the target menu is viewing a demonstration.
- If the user issuing the command is currently demonstrating.
- When the target menu has SPYABLE TELL specified and rejects the spy request.
- If the User option is specified when a shared *userid* option is in effect.

Use of the Spy facility is controlled by the configuration options SPYABLE, SPYGROUP, and the COMMAND statement. It may be made available only to those individuals who need it in the course of their work.

STARTCP administrator command

Note Default security authorization level: '9'.

```
STARTCp
```

The STARTCP command restarts the Session Manager TCP/IP manager after a STOPTCP command has been issued. Session Manager TCP/IP manager can be started automatically at the end of Session Manager startup processing, or by using an UPDATE command.

STARTLINK administrator command

Note Default security authorization level: '9'.

```
STARTLink linkname
```

The STARTLINK command activates a link which has either not been set up to start automatically when the Session Manager system initializes, or has been deactivated using the STOPLINK command. The linkname specified must correspond to a linkname as defined by the LINK configuration statement.

STOP administrator command

Note Default security authorization level: '9'.

```
STOP User userid|ALLDISC [user_qualifier] |
     LU luname   |
     [Sel seln-id] [ALL]
```

The STOP command is used to terminate sessions associated with another user or terminal. The SEL parameter may be used to stop a particular session, which may be specified as a session detail number, a PF key number, or a selection command.

If 'STOP USER *userid*' is specified for a SIGNON NO user, the user is stopped, provided that there is only one active user with the specified name. If there is more than one active, none are terminated. In this instance, STOP LU could be specified instead.

If 'STOP USER ALLDISC' is specified then all disconnected users will be logged off. (This means that ALLDISC is a reserved userid.)

The Shared User facility enables multiple users to sign on with the same userid (see the *Installation and Customization* manual). A unique, or fully qualified, identifier is created internally for each user by appending a 'user qualifier' to the signon userid. Depending on the value of parameter MULTUSER on the SYSTEM statement, either the 'user qualifier' is created from the last four characters of the terminal LU name, or it is an eight-digit number in the range 1-99999999.

For Session Manager 1.2.10 and higher, if multiple users are signed on with the same userid then you can include either a four-character or an eight-digit *user_qualifier* as appropriate with the 'STOP USER *userid*' command to distinguish which user is required. (The 'QUERY USER' command can be used to display the qualifier for Shared users.)

When the ALL parameter is used, all sessions are terminated and the user is logged off Session Manager (as for LOGOFF command).

When the command is issued without any optional operands, the user or terminal is disconnected from the system but any active sessions are maintained in an active state. The relevant Menu screen can be picked up at the same, or any other compatible terminal, and the active sessions picked up exactly where they were left. The effect therefore, is exactly as if the `DISCONNECT` command described earlier were issued at the target terminal.

STOPACB administrator command

Note Default security authorization level: '9'.

`STOPAcb` *`appl_name acb_name`*

The `STOPACB` command permanently allocates the specified ACB in the specified `APPL` statements's associated ACB range (defined using the `RANGE` statement), making the specified ACB unavailable for subsequent allocation. Both parameters are mandatory. This command is useful when, for whatever reason, the ACB is not compatible with the application.

For example, if an `APPL` has a problem with an ACB such that any sessions started with it are rejected, and the `RRA` parameter is *not* specified on the `APPL`'s associated `RANGE` statement (see '`RRA`' on page 145), then many users may be allocated this problem ACB. To take the ACB out of service, first you may have to prevent user access to the `APPL` temporarily (using a `BLOCK` *`applid`* `Yes` command), then issue the `STOPACB` command, and finally restore user access to the `APPL` (using a `BLOCK` *`applid`* `No` command).

STOPLINK administrator command

Note Default security authorization level: '9'.

`STOPLink` *`linkname`*

The `STOPLINK` command may be used to de-activate a link. The *`linkname`* specified must correspond to a link name entered on a `LINK` configuration statement.

STOPTCP administrator command

Note Default security authorization level: '9'.

`STOPTcp`

The `STOPTCP` command stops the Session Manager TCP/IP manager; Session Manager remains active. You can start the Session Manager TCP/IP manager again using the `STARTCP` command.

SWITCHPLX administrator command

Note Default security authorization level: '9'.

The `SWITCHplx` command is used in a Sysplex environment to transfer all users from one Session Manager instance to another Session Manager instance. This facility forms the planned element of the VTAM session recovery mechanism in Session Manager (the other element being recovery after a catastrophic failure). See the 'Parallel Sysplex support' chapter in the *Installation and Customization* manual for a full description of session recovery.

Note The `SWITCHplx` command is issued against the Standby instance or the Standby Controller, and not against an active Session Manager instance or Controller. This is because the active instance or Controller may not be in a state where it can accept and/or run the command. It is recommended that the command be used only via the 'Sysplex node status summary' panel (for details see the 'The Sysplex Summary and menu facilities' chapter of the *User and Administrator* manual).

TERMINATE administrator command

```
TERMINATE [FORce|END]
```

The `TERMINATE` command is a synonym of the `CLOSEDOWN` command. Refer to the `CLOSEDOWN` command on page 248.

TRACE administrator command

Note Default security authorization level: '9'.

```
TRace Data|MISER|Vtam|NETDATA [Yes|No|ON|OFF|SPIN]
      [User userid |
      LU luname    |
      [Selection seln-id] ]
```

There are six different types of trace available, `DATA`, `MISER`, `VTAM`, `NETDATA`, `INTERNAL` and `LINK` and they are controlled by the configuration `USER` or `TERMINAL` `TRACE` parameter, or `LINK` `TRACE` parameter, and also through the `TRACE` command. The `TRACE INTERNAL` and the `TRACE LINK` commands are described later.

`Data|MISER|Vtam|NETDATA`

When specified without any other parameters, the trace is started for the user issuing the command.

`Yes|No|ON|OFF|SPIN`

`Yes` and `ON` are the default parameters and request that a trace should start. `No` and `OFF` stop the tracing and also stop any internal tracing associated with the task. The `SPIN` parameter closes the current trace file, but the trace continues and subsequent trace output is written to a new file. An alternative method of closing the current trace file is to issue the command `SPIN TRACE`.

`User` *`userid`*
`LU` *`luname`*

Tracing may be restricted to a particular user or terminal as specified by the *`userid`* or *`luname`*. When no qualification is made, all associated tasks are traced for the user or terminal, that is the physical terminal, the user and all the session tasks. This should be allowed to default when tracing, unless a local support representative has requested otherwise.

Tracing of tasks may be limited to any one or any combination of the tasks associated with the user or terminal, by using the Selection parameter, as follows:

`Selection seln-id`

where '*seln-id*' may be a session detail number, a PF key number, or a selection command sequence.

TRACE LINK administrator command

```
TRace LINK linkname [NETCTL] [Yes|No|ON|OFF|SPIN]
```

The *linkname* specified on the command should match that specified on the `LINK` statement. `TRACE LINK ON` traces **all** the data sent to and received from the link and the associated internal postings for all sessions. The formatted output is written out to the same trace file as the DATA trace.

The optional operand `NETCTL` causes the external tracing of all network control data passing through the specified link. It is possible to have two types of LINK tracing in effect, for example, `TRACE LINK ON` and `TRACE LINK NETCTL OFF`, so that a separate trace of the network data is performed, in addition to normal link tracing.

Link tracing is performed for all links both active and inactive. The trace status remains in effect until changed by a new `TRACE LINK` command, or an `UPDATE` command is issued which replaces the current `LINK` statement definition.

Like conventional explicit Session Manager VTAM links, Sysplex links would only be traced if an IBM representative requires diagnostic data to help resolve a problem. Sysplex link tracing can be started either at start up by either specifying `LINKTRACE` or `LINKNETCTL` on the `SYSTEM` statement parameter `SYSPLEXGROUP` where it applies to all Sysplex links, or using the `TRACE LINK` command. `LINKTRACE` traces all data flowing on all Sysplex links. `LINKNETCTL` traces just link control data, not data buffers, on all Sysplex links. Each `TRACE LINK` command will trace just one link.

Link tracing should only be used at the request of your local support representative.

CAUTION Link tracing can generate a large amount of data, which may be generated internally at a faster rate than the output can be written and as a consequence may cause storage shortages.

TRACE INTERNAL administrator command

Note Default security authorization level: '9'.

```
TRace Internal [Yes|No|ON|OFF] [DUMP|NODUMP]
      [Task taskname  |
       User userid    |
       LU luname      |
       [Selection seln-id] ]
```

The `INTERNAL` trace is primarily intended for tracing Session Manager program flow, and would not normally be used unless a local support representative requests this type of trace.

The trace is stored in a wrap-around storage table and can be viewed using the `DSTORE` command, or in printed form after the `TRACE OFF` command has been issued. When no parameters are supplied, tracing starts for all current tasks associated with the user that issued the command.

`Yes|No|ON|OFF`

`ON` and Yes are the defaults and request that a trace should start. `OFF` and No terminate the tracing activity and also stop any Data or VTAM tracing for associated tasks.

`DUMP|NODUMP`

`DUMP` is used to request a dump of the internal trace table. `NODUMP` suppresses the output of the internal trace table when the trace is stopped. Both of these parameters allow the trace to continue.

`Task` *`taskname`*
`User` *`userid`*
`LU` *`luname`*
`Selection` *`seln-id`*

Tracing may be activated for a particular task by specifying the task name. The `USER` and `LU` parameters may be used to restrict tracing to a particular user or terminal. When these parameters are not qualified, all associated tasks are traced. If it is required to restrict tracing to selected tasks, the Selection parameters may be specified in the same way as they are for DATA, VTAM and NETDATA traces.

For further explanation of tracing, refer to 'Problem diagnosis and reporting' on page 273.

TTPSL administrator command

The `TTPSL` command should only be used under supervision from your local support representative. For panel and script auditing and tracing we recommend that you use the `FORMATMSG` and `AUDITMSG` facilities, as described in the *Panels, Scripts and Variables* manual.

Note Default security authorization level: '9'.

```
TTpsl [[Yes | No | ON | OFF | Display]
      Task task-name | [Selection seln-id
        [User userid | LU luname]]
      [PAnel | SCript]
      [PRint | NOPrint [DUmp | NODump]]
      [Internal | NOInternal]
      [All | TRaceverb]
      [SIze nn]]
```

The `TTPSL` command provides the facility to trace Session Manager panel and scripts defined by the Panel and Script Language (TPSL). The trace records can either be printed as each statement is processed or stored in a wrap-around internal table.

`Yes|No|ON|OFF|Display`

`ON` and Yes are the defaults when tracing is not already running, and cause tracing to start.

When the `OFF` or No parameters are specified, tracing is stopped and the internal trace table is cleared of any trace records.

The DISPLAY option causes the internal trace table to be browsed. This is the default when tracing is already started, and any other parameters previously specified will be used. If a Selection parameter is also specified, but tracing has not been started, then tracing is started. DISPLAY may not be entered from the console.

Task *taskname*
Selection *seln-id*
User *userid*
LU *luname*

These parameters determine what is to be traced.

The Task *taskname* option identifies either a running task, or a task which will be running when the script or panel is traced. The taskname can be found by issuing the QTASK command; refer to page 260 for details of this command.

The Selection *seln-id* option specifies the user session to be traced. The session can be specified as either a session number or a session selection sequence.

USER causes the user task for *userid* to be traced.

LU is the *luname* of the terminal to be traced.

The USER and LU options can only be specified on the SELECTION parameter, they identify which USER or LU is to be traced. The default is USER with the userid of the user who invoked the command. When TTPSL is invoked from the console, and Selection is specified, either USER or LU must also be specified.

PAnel|SCript

If PANEL is specified then the panel logic is traced. Alternatively, if SCRIPT is specified, the script logic is traced. SCRIPT is the default.

PRint|NOPrint
DUmp|NODUmp

These options specify the destination of the trace records.

PRINT causes the trace record to be sent to the Session Manager Audit file, or the console, or both, depending on the routing that has been specified for the TTPSL messages.

Note To avoid excessive amounts of output being sent to the console log, the following MESSAGE statement should be present in the configuration file:

```
MESSAGE 476 LOG NO AUDIT YES
```

NOPRINT indicates that the trace records are not to be sent to the Session Manager Audit file. NOPRINT is the default.

DUMP causes the trace records, stored in the internal wrap-around table, to be sent to the Session Manager Audit file when either the trace ends or when the TTPSL OFF command is issued.

NODUMP indicates that the trace records are not to be sent to the Session Manager Audit file when tracing ends. This is the default.

Internal|NOInternal

This option indicates whether the internal trace table should be used.

INTERNAL specifies that the internal trace table should be used to store the trace records. The internal trace table is a wrap-around table and its size is determined by the SIZE parameter below. The trace table can be browsed by using the TTPSL DISPLAY command.

NOINTERNAL specifies that no internal trace table is to be used. This is the default.

All|TRaceverb

This option indicates what level of tracing is to be performed.

ALL causes all SCRIPT and PANEL statements to be traced. This is the default.

TRACEVERB causes only the SCRIPT and PANEL statements between the TRACEON and TRACEOFF statements to be traced.

SIZE *nn*

This is the size of the internal trace table, and is only valid if the trace table is being used.

The size is specified in kilobytes. Any value between 1 and 31 can be used, the default is 12K.

When the trace records are displayed in the internal trace table, the latest record is at the bottom of the screen. The display is pageable. However, if a large number of statements have been traced then the initial trace records may have been overwritten as the table wraps around.

UPDATE administrator command

Notes

1. Default security authorization level: '9'.
2. This command is relevant only if you are *not* using Online and/or Batch Administration to tailor Session Manager. If you are using Online and/or Batch Administration, see instead 'PUPDATE administrator command' on page 259.

   Note The UPDATE command *is* still required to update the user exit and exit scripts, and to perform an UPDATE ALL.

```
UPDate Exit [exit-name|0]    |
       E05 S|E               |
       E06 S|E               |
       E08 S|E               |
       E11 S|E               |
       E21 S|E               |
       E22 S|E               |
       E26 S|
       E29 S|E               |
       E31 S|E               |
       E33 S|E               |
       E36 S|
       E39 S|E               |
       E79 S|E               |
       E99 S|E
```

```
UPDate Print [Yes|No|ON|OFF]
       Config xx             |
       Member member-name    |
       ALL
```

```
UPDate GCMVS
```

The UPDATE command has three formats. The first relates to updates made to the Exit facility; the second relates to updates to other components of the configuration and the last relates to updates to z/OS System Symbols designated by the variable names GC_MVS_*symbolname*. Previously these were only created at system startup. They can now be modified dynamically as required.

The UPDATE command enables the Session Manager environment to be changed without the need to close and restart Session Manager.

It is not possible to delete statements using the Update facility; the only definitions that are removed are those which are replaced by new definitions with a duplicate name. The chapter 'Using the Update Facility' on page 199 describes the Update facility in more detail.

Exit [*exit-name*|0]

EXIT requests a load of a new copy of the User exit as defined by the configuration OPTION statement EXIT parameter. The existing exit is deleted, so if the reload fails for any reason, there may be no exit in effect. An alternative method of loading a new exit is to specify an exit name on the EXIT parameter for the UPDATE command. In this case the existing exit is not deleted unless the load is successful. No other parameter should be entered on this command when EXIT is used.

To ensure that no exit is loaded, 0 may be specified. This deletes any existing exit from the Session Manager environment.

E*nn* S|E

When an UPDATE command is entered with one of the exit point parameters, the setting of the 'do not call again' switch is reset for the specified call number. This occurs even if the script cannot be found. The 'do not call again' switch is set by an exit, or script, by setting its return code to 40. It can also be reset by the return code 44, or an UPDATE EXIT command.

The exit points E01, E09 and E71 are only applicable to the User exit and cannot be specified as scripts. The exit points E26 and E36 can only be run as exit scripts.

Print [Yes|No|ON|OFF]

Printing for update may be specified using the PRINT parameter. If specified, it must be specified **before** any other parameter. It is valid with the CONFIG, MEMBER and ALL parameters. If omitted then the print setting specified on the OPTION statement at Session Manager start up is used. The PRINT parameter has no effect when specified with EXIT since only messages applicable to the load are printed.

The subparameters of the PRINT parameter are similar to those of the OPTION statement PRINT parameter. YES requests that all the control statements in the configuration source are printed, and NO suppresses print apart from statements in error and the appropriate error messages.

`Config` *`xx`*

`CONFIG` may be used to change, or make additions to, the current configuration. A suffix must be supplied and may be any two alphanumeric characters which may or may not match the start up configuration source member suffix.

The Update facility processes all statements in the specified file, including `COPY` statements.

The point at which each updated configuration statement or parameter comes into effect is dependent on its type. Details are provided in 'Using the Update Facility' on page 199.

`Member` *`member-name`*

To process statements in an additional source member, the `MEMBER` parameter may be used. A member name must be supplied and may be any valid combination of alphanumeric characters up to a maximum of eight.

All partitioned datasets in the search chain for the 'CONFIG DD' statement, are searched for the member name.

`ALL`

Causes Session Manager to use the `OPTION` statement specified at start up to determine which files are to be processed by the update.

`GCMVS`

The `UPDate GCMVS` command allows you to update the z/OS System Symbols designated by the variable names `GC_MVS_`*`symbolname`*. Previously these were only created at system startup. They can now be modified dynamically as required.

The chapter 'Using the Update Facility' on page 199 describes the Update facility in more detail.

CHAPTER 29

Problem diagnosis and reporting

In the unlikely event of problems being experienced with IBM Session Manager for z/OS it is essential to collect the correct information to enable rapid resolution. Session Manager supplies many diagnostic tools to assist in problem resolution. In addition, if errors do occur, then helpful messages are usually output to either the Menu display or the System Console. If the error message explanation does not enable the problem to be solved, please ensure the following basic information is made available when you report the problem to your local support representative.

- The version and level of the Session Manager system and the version of the operating system on which it is running. This is printed on the Session Manager Audit log at startup or can be displayed by issuing the `QUERY ISZSMGR` command. If the level of the operating system has recently been changed, then the old and new levels should also be supplied. The version and level of the VTAM system, if applicable, are also required.
- The type of terminal (if applicable) at which the error occurred.
- A complete list of the configuration source member.
- Any dumps that were output and error messages that were issued immediately prior to the problem occurring.

Other information may be required, depending on the type of problem encountered. Additional information required, together with advice on collecting it, is provided in the following problem descriptions.

These sections discuss the types of problems that may be encountered:

Incorrect results

This type of error can take many forms, from an unexpected message, to an incorrectly formatted panel. It is anticipated that the majority of errors will be with `PANEL` presentation and `SCRIPT` processing. Due to the nature of the error it cannot be replicated here but possible diagnosis methods are outlined below.

PANEL Processing

Check the correct panel has been invoked. As there are many ways to select a panel it is possible that recent changes may have invoked the wrong path. For example, issue the command `Value t_panel` to check that the correct menu panel is being displayed or issue the `PANELID` command to display the panel name in the top left hand corner.

AUDITMSG and FORMATMSG

The `AUDITMSG` and `FORMATMSG` facilities can be used to enhance understanding of script and panel processing.

- The `AUDITMSG` script parameter can be used to write messages to the audit file during script processing.
- The `FORMATMSG` function can be used in panels definitions to issue Session Manager messages.

Details of the `AUDITMSG` and `FORMATMSG` facilities are provided in the *Panels, Scripts and Variables* manual.

TTPSL

This can be used to trace the execution of a panel, in a manner similar to REXX tracing. Details on the `TTPSL` command are given on page 268 and in 'Tracing Panel and Script Processing' in the *Panels, Scripts and Variables* manual.

Messages

The content, format, and routing of Session Manager messages can be tailored, but the reason for issue cannot be altered. When an error message is issued either the *Messages and Codes* manual or the online Help displays should be consulted. A description and possible cause is given for each message.

Note The `MSGID` command swaps from showing, or hiding the message number, in all places in the system where messages are displayed. Alternatively you can code `MSGID ON` at the system level to always display message numbers.

3270 data stream errors

This is likely to be the most common type of problem encountered. This type of error does not necessarily (in itself at least) cause a dump to be taken, nor is one required. Session Manager does not in the main change user data between the terminal and application. However, when compression, and MISER, is active on the system the way in which the data stream is carried across the system is modified to increase efficiency. This can potentially lead to problems with emulated devices.

The most common occurrence of such problems is with file transfer. The `PCTRANSFER` parameter of Session Manager can be used to switch off data compression and allows the data to flow without modification. The `PCTRANSFER` parameter (and the `PCTRANSFER` command) switches off Compression, MISER, and many extended terminal features, such as the terminal alarm. This parameter is most useful when specified at the session level.

If problems persist, or if the symptoms are in the form of a PROGxxx message at the terminal, then the probable cause is that the data includes a 3270 command or sequence that is not supported by the receiving device, or the data has been corrupted. Whenever such an error is presented, Session Manager automatically traces the offending frame to the output designated for tracing. This type of trace generally indicates the problem but not the part of the process that is responsible. A full trace should be taken to ascertain these details. Information on taking traces is provided later in this chapter.

When traces have been taken that are to be passed to IBM for analysis it is advisable to terse the output and send via ftp, following the instructions at:

www-01.ibm.com/support/docview.wss?uid=swg21204910

Details of how to exchange information with IBM Technical Support can also be found at:

www.ibm.com/software/support/exchangeinfo.html

Further information on IBM's ECuRep system can be found at:

www.ibm.com/de/support/ecurep/index.html

Note The `TRACEROUTE` statement must be correctly specified in the configuration file to ensure the trace is collected correctly.

Inability to LOGON/SIGNON

There are a few error conditions that could occur (for example, storage shortage) where the most obvious symptom is that users can no longer signon to the system; that is, users of current sessions continue to work without problem but new users are unable to signon. In this situation, advance warning messages will have been output by GFS and the reason for the failure should be obvious from a dump of Session Manager. As this situation would not normally cause an automatic dump to be taken, a dump should be taken manually, using the Session Manager `DUMP` command.

In addition, various displays are usually available to check the state of the Session Manager environment from the operating system. The buffer statistics from VTAM can also prove useful in many instances.

A situation can sometimes be caused by a hardware problem. Check the correct logmode is being used and try altering the `REBIND` parameter setting. If you suspect the problem may be hardware related, and you cannot solve the problem yourself, your local support representative will require the make/model of the terminal and possibly that of the control unit, together with any other symptoms and messages output.

Inability to start forward sessions

Apart from a true storage shortage, the most common occurrence of this problem is a shortage of ACBs. This, in turn, may be caused by inappropriate coding; for example, a duplicate ACB name for forward session logon problems or lack of ACBs for access to Session Manager itself. It may also be caused by VTAM running short of storage and being unable to allocate any more ACBs.

When ACB errors do occur, they are usually accompanied by VTAM OPEN return codes. A list of the most common codes are given in Appendix 2 of the *Messages and Codes* manual.

Performance/tuning considerations

Getmain freemain storage routine

This routine improves performance by reducing the number of CPU cycles required to allocate and deallocate dynamic buffer storage. On initialization it obtains storage (default 10M) for Session Manager and then frees a storage cushion for Operating System usage. GFS is started by specifying `GFS=Y` on the `OPTION` statement in the configuration file.

The amount of storage managed by GFS can be changed using the `STORLIM` operand of the `GFS` parameter. The default value is 10M. A value of 72M is suggested as a starting point The `GFS USAGE` command should be entered during a period of high system activity to monitor the storage used and the `STORLIM` value adjusted as necessary.

For those Installations with no storage constraints the `MAXSTOR` operand of the `GFS` parameter may be specified. `MAXSTOR=Y` further reduces the possibility of storage fragmentation. However, it may increase storage used by up to 100 percent. If used, the figure specified for `STORLIM` should be doubled.

GFS displays

There are three displays available that provide further information regarding the way GFS is managing storage. These displays are invoked using the GFS command, which has a default authorization level of 9. The following section provides an overview of each display and a brief description of the fields.

Standard paging commands, that is `UP`, `DOWN`, `BACKWARD`, `BWD`, `FORWARD`, `FWD` and `TOP`, are available on all three displays to show any additional information.

Common header information

Each of the GFS panels has a common header containing the following information:

Map Type Specifies whether GFS is using BIT maps or BYTE maps to manage storage. The field will contain the value BIT or BYTE.

Slot Size Specifies the number of bytes of storage mapped by a Bit or BYTE.

MAXSTOR Y/N Specifies whether MAXSTOR processing is being used.

GFS STATS display

This display is invoked using the GFS STATS command. A sample display is shown below:

```
Session Manager                    GFS Stats                 xx/yy/200z 10:32:49
LU   S05TVV01                                                           MSK

  GET request in  Total Number   GET fitted    All submap  Slot free but
      Range         of GETS    in first slot   slots full    too small
  --------------  ------------ -------------  ------------ -------------
     0 to     56             5              5            0             0
    63 to    120           221            221            0             0
   127 to    152           207            207            0             0
   159 to    184           156            156            0             0
   191 to    312            91             91            0             0
   319 to    440            54             54            0             0
   447 to    568            69             69            0             0
   575 to    696            11             11            0             0
   703 to    856            39             39            0             0
   863 to   1016            49             49            0             0
  1023 to   1528           124            124            0             0
  1535 to   2040           105            105            0             0

Maxstor: No   GETs are rounded up to next 8-byte boundary    Maptype:BIT

===>

PF1:Help  PF3:Quit  PF4:Return  PF7:Bwd PF8:Fwd  PF9:Usage PF10:Stats PF11:Stor
```

Descriptions of the fields can be displayed by pressing the Help key, PF1.

If the GFSCLR command has been used to reset the statistics counters, a message will appear on the panel stating the data and time the command was issued, together with the Session Manager user id of the person who issued the command. For example:

```
Stats last reset at 15:23:50 on 03/10/02 by TXS
```

If no such message is displayed, the statistics shown on the panel are from the last startup of Session Manager.

GFS USAGE display

This display is invoked using the GFS USAGE command. A sample display is shown below:

```
Session Manager                    GFS Usage                 xx/yy/200z 11:22:47
LU   S05TVV01                                                           MSK

Submap  Range of GET  Bytes managed  Bytes used    Bytes free % free
------  ------------  -------------  ----------    ---------- ------
1 >16m   4096 above       3355392        251776       3103616    92%
2 >16m   2048  4088      10167872         77672      10090200    99%
3 >16m   1024  2040       6710784         36720       6674064    99%
4 >16m    704  1016       3355392          9144       3346248    99%
5 >16m    448   696       3355392         11832       3343560    99%
6 >16m    192   440       1677696         18920       1658776    98%
7 >16m    128   184       3355392          2024       3353368    99%
8 >16m      0   120       1576256          2760       1573496    99%
                       ----------------------------------------------
TOTALS >16m              33554176        410848      33143328    98%

Startup definitions : ( Smallest Slot % = remainder )
4096-10,2048-30,1024-20,704-10,448-10,192-5,128-10

Maxstor: No   GETs are rounded up to next 8-byte boundary    Maptype:BIT

===>

PF1:Help  PF3:Quit  PF4:Return  PF7:Bwd PF8:Fwd  PF9:Usage PF10:Stats PF11:Stor
```

Descriptions of the fields can be displayed by pressing the Help key, PF1.

Note In this example, the GET requests have been rounded up to the next 128-byte boundary.

GFS STOR display

This display is invoked using the GFS STOR command. A sample storage layout display is shown below:

```
Session Manager                  GFS Storage                xx/yy/200z  11:25:03
LU   S05TVV01                                                           MSK

   Cushion Address                        :   1261E000
   Cushion Size                           :    1048576 (X'00100000')

   Bitmap table address (above-the-line) :   1261DE48
   Bitmap size                            :     524284 (X'0007FFFC')

   Storage Segment Layout
   ----------------------

   Storage Address         Length
          1271E000       33554432 (X'02000000')





Maxstor: No   GETs are rounded up to next 8-byte boundary    Maptype:BIT

===>

PF1:Help  PF3:Quit  PF4:Return  PF7:Bwd PF8:Fwd  PF9:Usage PF10:Stats PF11:Stor
```

Descriptions of the fields can be displayed by pressing the Help key, PF1.

TPSL performance considerations

The way in which TPSL (the Panel and Script Language) is used in scripts and in panels can impact performance if incorrectly coded. The following list illustrates points that should be taken into account when creating scripts and panels.

- All variable reference should be minimized.
- Variables should be defined only if they are necessary. Defining a variable to hold a literal which is used once or twice is a waste because of the access involved in defining it, the extra access required to use it instead of a literal, and also because of the extra storage required.
- `DO FOR smax` is more efficient than `DO WHILE sub1 <= smax`. Using `FOR`, `smax` is evaluated once. Using `WHILE`, `smax` and `sub1` are evaluated on each iteration. This is also the case for `DO UNTIL`.
- Avoid evaluating unnecessary conditions. For example, if you have multiple `IF` statements, but know that only one of the conditions will be true, code `IF` then `ELSE` type conditions, or, better still, `SELECT WHEN` statements.
- Consider the most common path through the logic and test for those conditions first in `IF` and `SELECT` statements to minimize the instructions executed on the normal path.
- Minimize the number of `OUTSCAN` statements, because each `OUTSCAN` scans the data stream. If you need to code multiple `OUTSCAN`s, order them in the most likely sequence so as to prevent unnecessary scanning of the data stream.

- When you are coding loops on menus to check for sessions which are active and available, consider whether you need to loop from 1 to `smax` or whether you only need to evaluate the sessions which are currently displayed in the content section. If you code to remember which session is the first or last currently displayed, your loop then need only be executed for the number of sessions displayed, rather than the number of sessions available. This can save on the amount of variable access.
- If some of the code is static (that is, the variable values do not change after being evaluated once) it may be worth defining an extra variable. For example, a variable, un_first, could be set the first time through the menu and be subsequently tested each time to prevent the initialization code being executed each time.
- In the process section, if you are coding numerous commands, it may be worthwhile making the first `SELECT WHEN` condition a test for a blank command to do nothing except `RETURN` from the process section.

Diagnostic tools

Traces

There are several different formats of trace provided by Session Manager.

Data trace

This is the simplest form. It can be used at any time to trace the data streams flowing over a terminal-application session. As Session Manager is effectively in the middle of the flow, it captures data flowing from and to the application (`SESSION OUTPUT` and `SESSION INPUT`) and data flowing to and from the terminal (`TERMINAL OUTPUT` and `TERMINAL INPUT`). This type of trace is particularly useful when designing and testing scripts.

MISER trace

If MISER is being used, a data trace may not be enough to analyze a problem, as Session Manager 'massages' the data from the session, depending on the contents of the image buffer; that is, what was on the screen prior to this last output. This process of buffer analysis, edit, and merge is performed in an internal buffer area called an Image Buffer. A `TRACE MISER` option captures these as well as the normal data flow.

VTAM trace

In general, if there is a VTAM-type problem, a VTAM buffer and I/O and a Session Manager VTAM trace should be done simultaneously. The Session Manager trace traces the API (Application Program Interface) and is designed to assist in the diagnosis of interaction between VTAM and Session Manager. For example, it can be used when an application running under Session Manager behaves differently to when it is run without. The trace shows each VTAM event and also includes all data flow. The combination of the two types of trace enables a comparison to be made between what VTAM is sending, and what Session Manager is doing with the information. This type of trace is intended for use by IBM support personnel.

NETDATA trace

Netdata traces can be used to trace a specific network session's link traffic. The output gathered is a subset of the trace produced by Link tracing. DATA tracing is implied and is performed for all active tasks.

LINK trace

A Link trace can be used to trace the data sent to and received from the link and the associated internal postings for all sessions. The formatted output is written to the same trace file as the Data trace.

CAUTION Link tracing can generate a large amount of data which may be generated internally at a faster rate than the output can be written. In consequence, this may cause storage shortages. This type of trace should only be performed at the request of your local support representative.

Starting traces

All traces are started by an authorized user from the command line of a panel or from the Session Manager console. To start a trace enter a `TRACE USER` *xxxxxx* `ON DATA` command. To stop a trace enter a `TRACE USER` *xxxxxx* `OFF DATA` command.

Printing traces

All types of trace may be printed by using the `SPIN TRACE` command. The command closes and reopens the trace print file for DATA, VTAM, or MISER traces. This enables previous output to be printed according to the location specified on the `TRACEROUTE` statement. The command:

- `TRACE D SPIN` prints DATA traces only
- `TRACE V SPIN` prints VTAM traces only
- `TRACE M SPIN` prints MISER traces only.

As with the `SPIN` command, the current trace file is closed and tracing continues in a new file.

TTPSL trace

TTPSL provides the ability to track the execution of TPSL logic in Panels and Scripts. The area traced may be limited with the `TRACEON` and `TRACEOFF` parameters.

Notes

1. To trace a script or panel associated with an application on the Menu, specify 'Panel' or 'Script', together with the userid and selection-id.
2. To trace other TPSL logic, such as the Menu panel itself, or a Windows script, specify the taskid. To find the taskid, issue the `QTASK` command from the terminal that is to be traced and then enter this taskid in the `TTPSL` command.

DUMP command

This command allows a non-destructive dump to be taken of the Session Manager environment. The dump is sent to wherever has been designated as the dump receiver, by the startup process; for example, // SYSUDUMP.

DSTORE command

This command allows Session Manager storage to be displayed and altered. It is recommended that only selected users have authority to use this command. Full details on use of the command are given in the `DSTORE` description on page 250.

DTERM command

This command shows details of the functions that Session Manager believes the physical terminal is capable of. These details can assist in translating the effect of the data seen in a Session Manager Data Trace. It is intended primarily for use by IBM support personnel. Full details on use of the command are given in the `DTERM` description on page 254.

Abends

In the unlikely instance of a Session Manager failure, a dump is always taken. If the failure is recoverable (the majority of cases) then a controlled dump is taken. If a non-recoverable failure should ever occur in Session Manager, a dump of the Session Manager environment is automatically taken. All abend conditions therefore cause a dump to be generated.

In such situations there is little that can be done by the user. Contact your local support representative and provide the following information:

1. The content of message 114E and the associated PSW and registers. (It may be that a similar situation may have arisen at a different site and a resolution is already known.)

   If a resolution is not possible from the information provided here, then provide:

2. The complete dump and the complete Audit file, right from the start-up messages.

Dump notes

Which dump to send

It is possible that even though the abend has been controlled, something has been irrevocably damaged – storage corruption, for example. In this case, there may be subsequent abends, all with the same root cause. For this reason, Session Manager adds a dump number to each dump taken. Only 'DUMP 1' contains the diagnostic information that is required by your local support representative.

What to send

Whenever possible please send the dump via ftp, using the instructions at:

http://www-01.ibm.com/support/docview.wss?uid=swg21204910

Details of how to exchange information with IBM Technical Support can also be found at:

www.ibm.com/software/support/exchangeinfo.html

Further information on IBM's ECuRep system can be found at:

www.ibm.com/de/support/ecurep/index.html

If there is any doubt about the format of the data to be sent please contact your local support representative before dispatch, **or** ensure that the data is extracted from the operating system SPOOL in any way that is not release level dependent.

CHAPTER 30

The ATTN key and locked sessions

In general, there are two methods of attempting to unlock a terminal keyboard when a session with an application 'locks up', dependent upon the terminal type. On non-SNA terminals, the RESET key unlocks the keyboard, and on SNA terminals, the ATTN key is used. On SNA devices, the success of using this key is dependent upon the application handling it correctly, which may not always be the case. A 'lock up' may not always be permanent, it could be caused simply by a very slow response from the application. Nevertheless, even this situation locks the keyboard until the application involved responds.

However, when IBM Session Manager for z/OS is driving the terminal and handling all the communications between it and the various sessions active on the Menu screen, a user can escape from a locked keyboard. This is done on SNA terminals by pressing the ATTN key and then entering the appropriate escape command (which may itself be the ATTN key). On non-SNA terminals it is done by pressing the RESET key and entering the escape command. If the escape command is a character sequence rather than a function or attention key or the lightpen, the Enter key should be depressed after keying the escape sequence so that it is transmitted to Session Manager. Session Manager then acts upon the escape command and, dependent upon the type of escape, either transfers control to another (normally processing) application, or returns to the Menu screen.

If an unrecognized sequence of characters is entered after using the ATTN or RESET key to unlock the keyboard, Session Manager acts differently for SNA and non-SNA sessions. For non-SNA sessions, it passes the keyed characters as normal data to the 'locked' application. For SNA sessions, it ignores the entered data. Since this is what would normally happen anyway in these situations, the results are exactly as they would be without Session Manager, that is, the terminal will probably lock up again. To send 'ATTN' to a 'locked' SNA terminal session, press the ATTN key once to unlock the terminal and then press the ATTN key again to send ATTN to the locked session.

However, when this facility is used as intended, users who normally have more than a single application active can continue working even if one of their applications becomes unavailable.

The following table should help to determine the effect of entering normal data or an escape command when a non-SNA or SNA terminal is locked:

Device	Session	Key to Unlock	Escape Command	Data Keyed
Non SNA	Non SNA	Reset	is processed	Data is sent
Non SNA	SNA	Reset	is processed	ATTN is sent
SNA	Non SNA	Attn	is processed	Data is sent
SNA	SNA	Attn	is processed	ATTN is sent

Finally, if the ATTN key is pressed when the keyboard is not in a locked state (and the ATTN key is not itself the escape command), Session Manager will pass it to the application if it is an SNA session, otherwise it is ignored.

Index

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

Bibliography

IBM Session Manager library

The following publications contain information about IBM Session Manager.

Installation and Customization	GC34-7146-00
Technical Reference	SC34-7147-00
User and Administrator	SC34-7150-00
Panels, Scripts and Variables	SC34-7148-00
Messages and Codes	GC34-7152-00
Quick Reference	SC34-7151-00
Online and Batch Administration	SC34-7149-00
Program Directory	GI13-0564-00

Accessibility

Accessibility for people with disabilities

The following features make it easier for disabled people to use Session Manager:

- Operation by keyboard alone
- Optional font enlargement
- High-contrast display settings
- Can be used with screen readers
- Absence of audio prompts.

Changing font, color and display settings

Session Manager can be controlled using a 3270 emulator such as IBM Personal Communications or Hummingbird HostExplorer. Refer to the emulator documentation for guidance on adjusting font and color settings.

Using Session Manager with a screen reader

Screen readers can be used to provide accessible output for blind users. Session Manager has been tested with the following screen readers:

- Jaws version 4.5, using Hummingbird HostExplorer and the script file for Hummingbird HostExplorer
- WindowEyes 4.2, using Hummingbird HostExplorer and the set file for Hummingbird HostExplorer.

Contact the screen reader manufacturer for information about the availability of set and script files.

Documentation

Softcopy PDF documentation is shipped with Session Manager. The documentation supports optional font enlargement, high-contrast display settings, and may be operated by the keyboard alone. Alternative text is not provided for screen-reader users. Fully accessible softcopy documentation, with alternative text for diagrams, will be made available on request. Contact your IBM service representative for information.

Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 Japan

The following paragraph does not apply in the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore this statement may not apply to you.

This publication could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM United Kingdom Laboratories, MP151, Hursley Park, Winchester, Hampshire, England, SO21 2JN. Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Programming License Agreement, or any equivalent agreement between us.

Trademarks

Sending your comments to IBM

If you especially like or dislike anything about this book, please use one of the methods listed below to send your comments to IBM.

Feel free to comment on what you regard as specific errors or omissions, and on the accuracy, organization, subject matter, or completeness of this book.

Please limit your comments to the information in this book and the way in which the information is presented.

To ask questions, make comments about the functions of IBM products or systems, or to request additional publications, contact your IBM representative or your IBM authorized remarketer.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate, without incurring any obligation to you.

You can send your comments to IBM in any of the following ways:

- By mail, to this address:

  User Technologies Department (MP095)
  IBM United Kingdom Laboratories
  Hursley Park
  WINCHESTER,
  Hampshire
  SO21 2JN
  United Kingdom

- By fax:
  - From outside the U.K., after your international access code use 44–1962–816151
  - From within the U.K., use 01962–816151
- Electronically, use the appropriate network ID:
  - IBM Mail Exchange: GBIBM2Q9 at IBMMAIL
  - IBMLink™ : HURSLEY(IDRCF)
  - Internet: idrcf@hursley.ibm.com

Whichever you use, ensure that you include:

- The publication title and order number
- The topic to which your comment applies
- Your name and address/telephone number/fax number/network ID.

IBM®

SC34-7147-00