



IBM Session Manager for z/OS

# Online and Batch Administration

*Version 3 Release 1*





IBM Session Manager for z/OS

# Online and Batch Administration

*Version 3 Release 1*

**Note**

Before using this information and the product it supports, be sure to read the general information under “Notices” on page 129.

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.

Copyright © 2003-2011 All Rights Reserved. Macro 4 Limited - a division of UNICOM Systems, Inc.

# Contents

<b>About this manual . . . . .</b>	<b>9</b>
Session Manager documentation . . . . .	10
Conventions . . . . .	11
Summary of new features . . . . .	11
<b>Chapter 1 Introduction . . . . .</b>	<b>13</b>
<b>Chapter 2 Configuration . . . . .</b>	<b>19</b>
Hidden application sessions . . . . .	20
Menu sequence numbers for sessions . . . . .	21
Session types for sessions . . . . .	22
Setting priority sessions . . . . .	23
Sample job ISZSTJOB . . . . .	23
Dynamic Session Manager menus . . . . .	25
<b>Chapter 3 Security considerations . . . . .</b>	<b>27</b>
Overview . . . . .	28
OLAClass attribute . . . . .	29
OLA security settings . . . . .	32
Setting up OLA security . . . . .	33
OLA security settings for menu items . . . . .	34
OLA security settings for parameter definitions . . . . .	34
OLA security settings for miscellaneous entries . . . . .	34
<b>Chapter 4 Menus, lists and displays . . . . .</b>	<b>39</b>
Maintaining configuration definitions: overview . . . . .	41
How to use OLA menus . . . . .	41
How to use OLA lists . . . . .	42
How to use OLA 'Attribute Expansion' displays . . . . .	45
Escape processing . . . . .	46
Parameter validation . . . . .	46

Keywords used for display purposes only . . . . .	47
Summary of selection options and commands . . . . .	47
Main Menu . . . . .	50
Display My USER definition menu . . . . .	50
Display list of PROFILE definitions . . . . .	51
Display list of Local USER definitions . . . . .	54
Display list of RUSER definitions . . . . .	55
Display list of APPL definitions . . . . .	55
Display list of TERMINAL definitions . . . . .	56
Display list of GROUP definitions . . . . .	56
Display Hardcopy Configuration Statements menu . . . . .	56
Display System Configuration Statements menu . . . . .	56
Display list of OLA security definitions . . . . .	57
View Activate messages . . . . .	57
USER Definition menu . . . . .	58
Display enduser specific parameters . . . . .	58
Display user's common enduser parameters . . . . .	60
Display user's common session parameters . . . . .	65
Display list of sessions associated with user . . . . .	66
Display User View of session list . . . . .	69
Hardcopy Configuration Statements menu . . . . .	71
Display list of hardcopy profiles . . . . .	71
Display list of hardcopy formats . . . . .	71
Display list of hardcopy routes . . . . .	72
System Configuration Statements menu . . . . .	73
Display list of RANGE definitions . . . . .	73
Display list of LINK definitions . . . . .	73
Display list of COMMAND definitions . . . . .	74
Display list of SYSTEM definitions . . . . .	74
Display list of MESSAGE definitions . . . . .	74
<b>Chapter 5 Batch administration . . . . .</b>	<b>77</b>
Introduction . . . . .	78
Statement syntax . . . . .	78
Applying updates . . . . .	78
Activating the updates . . . . .	79
Updates based on current values . . . . .	79
Combining the commands . . . . .	80
Running Batch Administration . . . . .	81
Executing the batch program . . . . .	81
Specifying update commands . . . . .	82
ISZSRCH – search the configuration . . . . .	85
ISZSRCH output format . . . . .	88
ISZSRCH processing sequence . . . . .	91
ISZSRCH return values . . . . .	92
ISZUPDT – update the configuration . . . . .	93
ISZUPDT input format . . . . .	96

ISZUPDT statement types . . . . .	98
ISZUPDT output format . . . . .	101
ISZUPDT processing sequence . . . . .	102
ISZUPDT return values . . . . .	103
ISZACTV – activate pending updates . . . . .	104
ISZACTV input format . . . . .	106
ISZACTV output format . . . . .	106
ISZACTV return values . . . . .	107
ISZREMC – invoke Session Manager command . . . . .	108
ISZREMC return values . . . . .	108
Examples . . . . .	109
Creating a user definition . . . . .	109
Copying a user definition . . . . .	110
Deleting user definitions . . . . .	111
Modifying a user definition . . . . .	112
Creating user definitions using a script . . . . .	113
Deleting user definitions using a script . . . . .	114
Copying a user definition using a script . . . . .	115
Deleting a specific session from all users . . . . .	115
Batch Administration configuration . . . . .	117
<b>Index . . . . .</b>	<b>121</b>
<b>Bibliography . . . . .</b>	<b>125</b>
IBM Session Manager library . . . . .	125
<b>Accessibility . . . . .</b>	<b>127</b>
Accessibility for people with disabilities . . . . .	127
Changing font, color and display settings . . . . .	127
Using Session Manager with a screen reader . . . . .	127
Documentation . . . . .	127
<b>Notices . . . . .</b>	<b>129</b>
Trademarks . . . . .	130
<b>Sending your comments to IBM . . . . .</b>	<b>131</b>





# About this manual

This is the *Online and Batch Administration* manual for IBM® Session Manager for z/OS®. It is geared towards administrators and end-users of the product.

The manual describes how to use the following Session Manager capabilities to administer product configuration data:

- **Online Administration**

Instead of supplying product configuration statements directly, Online Administration (hereafter “OLA”) enables you to tailor Session Manager 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 you to tailor Session Manager using a batch job.

**Note** To use these capabilities, you must install and run Session Manager in OLA mode (see the *Installation and Customization* manual). In this format, configuration data is stored in several PDS(E)s – each of which is allocated to a particular DDNAME and *must be maintained exclusively by Online and/or Batch Administration*.

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

## Other ways to configure Session Manager

Session Manager can also 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. Full descriptions of each Session Manager configuration statement and each Session Manager command can be found in the *Technical Reference*.

- **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 titled ‘The Session Manager user exit processing’ in the *Installation and Customization* manual.

### External Security Managers

External Security Managers (hereafter “ESMs”), such as RACF<sup>®</sup>, 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
<i>Installation and Customization</i>	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.
<i>User and Administrator</i>	Describes in detail the features and facilities provided by Session Manager.
<i>Online and Batch Administration</i>	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.
<i>Technical Reference</i>	Provides a detailed reference for Session Manager commands and configuration statements, along with problem diagnosis assistance.
<i>Quick Reference</i>	Provides a quick way to find the correct syntax for commands, configuration statements, and variables, without detailed explanations.
<i>Panels, Scripts and Variables</i>	Gives a detailed technical account of defining panels, using scripts and variables, and the product’s Panel and Script Language (TPSL).
<i>Messages and Codes</i>	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:

<b>boldface</b>	Indicates a command or keyword that you should type, exactly as shown.
<i>italics</i>	When using Batch Administration, <i>specify configuration statements and parameters in full</i> – that is, do <i>not</i> use a shortened form documented in the <i>Technical Reference</i> .
<i>italics</i>	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.

## 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

# Introduction

This chapter introduces you to the facilities provided by the OLA capability of IBM Session Manager for z/OS.

**Note** New users of Session Manager should review the *Installation and Customization* manual, which should give a good understanding of the product's concepts and enable administrators to tailor Session Manager to Installation requirements.

**See**

- 'What product configuration data can be maintained by OLA?' on page 14
- 'What changes are required to an existing configuration?' on page 14
- 'How is OLA set up?' on page 14
- 'How is security handled?' on page 14
- 'What are the main OLA menus, lists and displays?' on page 14
- 'Is online Help available for OLA?' on page 16
- 'What if many changes are required to the configuration?' on page 16
- 'How are sessions ordered on Session Manager menus?' on page 16
- 'Can specified Session Manager sessions be hidden?' on page 16
- 'Does Session Manager eliminate duplicate sessions?' on page 16
- 'What are dynamic Session Manager menus?' on page 16
- 'Is a sample Session Manager menu available?' on page 17

## What product configuration data can be maintained by OLA?

OLA can be used to maintain all product configuration data, except for:

- Specialized definitions that are stored in the ISZCONxx members (APPLYSU, AUDITROUTE, COPY, INSTALLSU, OPTION, REMOVESU, TRACEROUTE, PATCH, PATCHSU and TRANSTABLE; and Session Manager panels and scripts.). To update these definitions, you must edit the members directly (see the *Installation and Customization* manual).
- Definitions stored in system datasets can *not* be modified or deleted (see the ‘Customer, samples and system configuration datasets’ section in the ‘Performing a basic configuration’ chapter of the *Installation and Customization* manual).

## What changes are required to an existing configuration?

In the case of new installations, install the latest version of Session Manager in OLA mode (see the *Installation and Customization* manual).

If you are an existing Session Manager Classic user, you must first upgrade to the latest version of Session Manager in Classic mode (see the *Installation and Customization* manual). Then you must run the OLA Enabler to implement the new format configuration. For details, see ‘OLA Enabler’ in the ‘Converting a Classic to an OLA system’ chapter of the *Installation and Customization* manual.

If you are an existing Session Manager OLA user, you must first upgrade to the latest version of Session Manager in OLA mode.

## How is OLA set up?

Usually, OLA will be used to administer a single Session Manager instance, but it can also be used to administer multiple Session Manager instances. For the appropriate set-up instructions refer to ‘Setting up OLA for single and multiple Session Manager instances’ in the ‘Post-installation configuration issues’ chapter of the *Installation and Customization* manual.

## How is security handled?

The facilities available to a particular user depend on that user’s OLA security class – for details, see ‘Security considerations’ on page 27.

## What are the main OLA menus, lists and displays?

A summary list of the main OLA facilities appears below; for a detailed description of each these facilities, see ‘Menus, lists and displays’ on page 39.

### Main Menu

Display My USER definition menu

Display enduser specific parameters

Display user’s common enduser parameters

Display user’s common session parameters

Display list of sessions associated with user

Display list of PROFILE definitions

Display list of Local USER definitions

Display enduser specific parameters

- Display user's common enduser parameters
- Display user's common session parameters
- Display list of sessions associated with user
- Display list of RUSER definitions
- Display list of APPL definitions
- Display list of TERMINAL definitions
- Display list of GROUP definitions
- Display Hardcopy Configuration Statements menu
  - Display list of hardcopy profiles
  - Display list of hardcopy formats
  - Display list of hardcopy routes
- Display System Configuration Statements menu
  - Display list of RANGE definitions
  - Display list of LINK definitions
  - Display list of COMMAND definitions
  - Display list of SYSTEM definitions
  - Display list of MESSAGE definitions
- Display list of OLA security definitions
- View Activate messages

**Notes**

- 1** The facilities available to a particular user depend on that user's OLA security class. For details, see 'Security considerations' on page 27.
- 2** Hidden application sessions (see page 20) will only be shown in OLA menus, lists and displays if the user's OLA security class allows it.
- 3** The External security exit and its Session Manager configuration settings, if installed, will determine if hidden application sessions will be shown in OLA menus, lists and displays.

## Is online Help available for OLA?

Yes. To access the online Help, press the PF Key assigned to the Help function. The top level online Help panel will appear:

```

Session Manager          S/MGR Help - Online Admin Menu          dd/mm/yyyy hh:mm:ss
LU      luname                                     user

This screen gives the main headings for which help is available
for the Online Administration facilities provided in S/MGR.

1 GENERAL   - General information
2 SAVE      - Save and Activation information
3 SELECTION - Selection options
4 PFKEYS    - PFKEYS, Commands and Filtering
5 SECURITY   - Security information
6 SESSION   - Session List information

Enter the required number in the command area, or enter the
highlit part of the help heading, to display the appropriate
help information.

*****      Press PF11 for next topic      *****

====>

PF1:HELP PF3:QUIT PF4:RETURN PF6:TOP PF7:BWD PF8:FWD PF10:PREV PF11:NEXT

```

From here, enter the option appropriate to the point in OLA at which you got stuck and you will be lead to more detailed Help pertaining to that topic.

## What if many changes are required to the configuration?

If many changes are required to a large number of configuration definitions, the Batch Administration capability in Session Manager enables you to tailor Session Manager using a batch job. For details, see 'Batch administration' on page 77.

Batch Administration also enables you to search the Session Manager configuration files for definitions which match specified search criteria (see 'ISZSRCH – search the configuration' on page 85).

## How are sessions ordered on Session Manager menus?

Application sessions are ordered according to their menu sequence numbers. For details, see 'Menu sequence numbers for sessions' on page 21.

## Can specified Session Manager sessions be hidden?

Yes. For details, see 'Hidden application sessions' on page 20.

## Does Session Manager eliminate duplicate sessions?

Yes, Session Manager uses session types to eliminate duplicate application sessions. For details, see 'Session types for sessions' on page 22.

## What are dynamic Session Manager menus?

For a particular user, Session Manager menus can be configured dynamically from access rules specified using the Installation's ESM. For details, see 'Dynamic Session Manager menus' on page 25.



### Is a sample Session Manager menu available?

Sample menus are shipped with Session Manager which:

- Demonstrate the product's menu ordering and sorting ability, and other facilities added in this release. (The sample menus will not display hidden sessions.)
- If your OLA security class allows it (see 'Security considerations' on page 27), enable you to invoke OLA.

The sample menus can be found in member ISZLENPU, which is supplied in library .SISZCONF.

For all Session Manager menus, including the sample menus, the program code for Session Manager will:

- Prevent you from selecting an application session if its s\_hidden value is set to Yes (see page 20).

**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.

- Order application sessions based on the session's s\_sequence value (see page 21).
- Sort application sessions with the same s\_sequence value (see page 21), based on the VTAM<sup>®</sup> applid.

### Notes

- 1 The default menu used by the system is defined by the DEFMENU parameter of the SYSTEM statement (see the product's *Technical Reference*). If the parameter is not specified, a menu called MENU is required.
- 2 The PANEL definition(s) can be amended so that the format of the Menu display conforms to your Installation standards. For details of the PANEL statement and related parameters, see the *Panels, Scripts and Variables* manual.



**CHAPTER 2**

# Configuration

This chapter describes how to configure OLA after IBM Session Manager for z/OS has been successfully installed and set up, as described in the *Installation and Customization* manual.

The subjects covered in this chapter are:

- 'Hidden application sessions' on page 20
- 'Menu sequence numbers for sessions' on page 21
- 'Session types for sessions' on page 22
- 'Dynamic Session Manager menus' on page 25

## Hidden application sessions

Specified Session Manager application sessions can be configured to be hidden.

### Session Manager

In Session Manager itself, hidden application sessions:

- Will not appear on Session Manager menus (and cannot be made visible from the menus).
- Cannot be invoked by users using an escape sequence/program function key.

### OLA menus, lists and displays

In OLA menus, lists and displays, hidden application sessions will only be shown if the user's OLA security class allows it.

### E22 and security

If a site is running the E22 security exit, with external security active, then application sessions may be hidden. In particular, requesting 'My USER definitions' will only display those sessions a user has access to, as permitted by these security constraints.

Additionally, if a site is running the E22 security exit, with external security active, then the AUTH and OLAClass settings will be non-modifiable. When viewed, they will display the settings defined by the security exit.

### How to specify that an application session is hidden

To specify that an application session is hidden, either:

- specify the common session parameter `HIDE`, which is applicable to the `USER`, `TERMINAL`, `PROFILE`, `APPL` and `SYSTEM` configuration control statements (for details, see the *Technical Reference*)
- or
- if your OLA security class allows it (see 'Security considerations' on page 27), use `OLA`

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

## Menu sequence numbers for sessions

For application sessions defined in USER, TERMINAL, PROFILE, APPL and SYSTEM configuration control statements, menu sequence numbers are used by the program code for Session Manager to order sessions:

### How to assign a sequence number to an application session

To assign a menu sequence number to an application session, either:

- specify the common session parameter `SEQUENCE`, which is applicable to the USER, TERMINAL, PROFILE, APPL and SYSTEM configuration control statements (for details, see the *Technical Reference*).
- or
- if your OLA security class allows it (see ‘Security considerations’ on page 27), use OLA. OLA can be used both to set the `SEQUENCE` parameter of the control statements and to give certain sessions ‘priority’, so that they appear above other sessions in the list.

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.

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

### Setting priority sessions

Individual sessions can be set as ‘priority’ sessions on a user-specific basis, enabling users to choose particular sessions that they wish to display at the top of their list. If OLA is being used this can be done through the Session List (see ‘Changing the order of session definitions’ on page 68). For information on setting priority sessions, see page 23.

## Session types for sessions

For application sessions defined in USER, TERMINAL, PROFILE, APPL and SYSTEM configuration control statements, session types are used by Session Manager to eliminate duplicate sessions. A user can be associated with multiple profiles (for details, see the *Technical Reference*), so the ability to eliminate duplicate sessions is especially useful when assigning more than one profile to a USER definition or a TERMINAL definition.

### How to assign a session type to an application session

To assign a session type to an application session, either:

- specify the common session parameter SESTYPE, which is applicable to the USER, TERMINAL, PROFILE, APPL and SYSTEM configuration control statements (for details, see the *Technical Reference*)
- or
- if your OLA security class allows it (see ‘Security considerations’ on page 27), use OLA

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.

**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 and then run the OLA Enabler to implement the new format configuration, session types can optionally be assigned to sessions automatically (see ‘Assignment of session types to sessions’ in the ‘Converting a Classic to an OLA system’ chapter of the *Installation and Customization* manual). This is recommended if user-customizable session ordering is to be used.

## Setting priority sessions

Individual sessions can be set as ‘priority’ sessions on a user-specific basis, enabling users to choose particular sessions that they wish to display at the top of their list. This can be done through the Session List in OLA (see ‘Changing the order of session definitions’ on page 68.)

Users may have their sessions specified on their USER statement and also on one or more PROFILE statements. Each session should ideally have its own definition on the APPL statement.

The user or a System Administrator may wish to change the session number or PF Key associated with any one of the defined sessions. If any of the sessions to be changed has been set as a ‘priority’ session by a user, and the user can set the common enduser parameter `SESPRIAPPL` to NO (the default), then in order for the correct order on the user’s menu to be retained *all* sessions will require a unique identifier, separate from the session number. This is achieved by allocating all the sessions a unique `SESTYPE`. This can be done by:

- using the `addsestype` parameter when running the OLA Enabler (see ‘OLA Enabler’ in the ‘Converting a Classic to an OLA system’ chapter of the *Installation and Customization* manual).
- using the OLA interface, or
- executing the sample ISZSTJOB.

Any new APPL definitions added to Session Manager should also be allocated a unique `SESTYPE`. This can be achieved by configuring and executing the sample ISZSTJOB or by allocating a unique `SESTYPE` to the new definitions by using the OLA interface.

**Note** Installations that manually allocate unique `SESTYPE`s will need to keep an up to date record of all allocated `SESTYPE`s in order to avoid duplication.

## Sample job ISZSTJOB

The sample ISZSTJOB will scan the configuration for all the existing `SESTYPE` settings and then allocate unique numbers, starting with a value of 1000, to all the defined sessions that do not currently have a `SESTYPE` set.

ISZSTJOB provides the same functionality to systems already using an OLA-enabled configuration as the `addsestype` parameter on the OLA Enabler does to systems which are being upgraded from a Classic to an OLA-enabled configuration. See ‘OLA Enabler parameters’ in the ‘Converting a Classic to an OLA system’ chapter of the *Installation and Customization* manual.

Setting a unique `SESTYPE` for each APPL statement restricts all users to just one session for each APPL statement.

A sample ISZSTJOB batch job can be found in member ISZSTJOB, which is supplied in library `.SISZCONF`. To run the ISZSTJOB, complete these steps:

- 1** Edit the sample job ISZSTJOB and make these changes:
  - a** Change the DSNPREF= parameter to match the prefix that you specified when you installed the OLA system and a new set of configuration datasets was allocated.
  - b** Change the CONFDSN= parameter to reference the supplied configuration dataset.
  - c** Change the LOADDSN= parameter to reference your current Session Manager LOADLIB.
  - d** Tailor the remainder of the JCL to your Installation's requirements.
- 2** Submit the sample job ISZSTJOB and check that it completes successfully.
- 3** Either shut down and restart Session Manager, or refresh the APPL definitions within the configuration by issuing the PUPDATE command (see the *Technical Reference*.)

The ISZSTJOB can be run as and when required by your Installation.



## Dynamic Session Manager menus

Session Manager menus for particular users can be configured dynamically from access rules specified using the Installation's ESM.

Dynamic menus are *not* administered using OLA; instead, this facility is provided by a sample E22 (signon completion) user exit. For details, see the product's *Installation and Customization* manual.

### Number of sessions defined for the user

The sample E22 user exit utilizes a pro forma (default) PROFILE definition that contains the maximum number of expected sessions; the actual number of sessions available to the user is likely to be less than this number.

Variable `smax`, the number of sessions defined for the user, is useful in defining the DO loop which processes a user's sessions for Menu display. `smax` can be written as well as read, and its value can be updated to a value less than or equal to the (maximum) number of (expected) sessions defined.



**CHAPTER 3**

# Security considerations

This chapter describes how access to OLA facilities is controlled by IBM Session Manager for z/OS.

**See**

- ‘Overview’ on page 28
- ‘OLAClass attribute’ on page 29
- ‘Setting up OLA security’ on page 33

## Overview

When a user attempts to invoke a particular OLA task (for example, to display a menu, or to list certain attribute values), the OLA security function checks to see if the user has the appropriate authority to perform the required task.

The security mechanism used to control access to OLA facilities is internal to Session Manager; each user will be assigned an OLA security class that defines what the user can or can't do.

The user's OLA security class can either be specified within OLA, or can be derived from an ESM by using the SECURITY parameter on the SYSTEM statement. For more information on using an ESM to set the OLA security class, see the *Technical Reference* manual.

Each user is associated with an OLA security class using the common enduser parameter OLAClass, which is applicable to the USER, RUSER, TERMINAL, PROFILE and SYSTEM configuration control statements. For a description of the possible values, see 'OLAClass attribute' on page 29.

When running OLA, specialized security settings are used to define completely what the user is allowed to do. For details, see 'OLA security settings' on page 32.

## OLAClass attribute

The user's OLA security class is specified using the OLAClass attribute. The Session Manager configuration as shipped comes with these OLA security classes defined:

Class	Category	Description
AD	Administrator	<p>Class default is set to ALL allowing access to all areas in OLA but with the restriction that access is not allowed to OLA security settings. Therefore, users in this class can <i>not</i>:</p> <ul style="list-style-type: none"> <li>▪ Modify the OLAClass or AUTH parameter, which would allow them to override their security.</li> <li>▪ Copy any USER statement with a name prefixed with IM (for example, IMJOE or IMSUE) as these users are assumed to be implementors.</li> </ul> <p><b>CAUTION</b> Although the supplied examples stop the copying of USER or PROFILE statements prefixed with IM, you should also ensure that class AD does not have access to any other USER or PROFILE statements that contain a higher security setting for parameters OLAClass or AUTH.</p>
BT	Batch Administrator	<p>Class default is set to ALL, allowing access to all areas in OLA.</p> <p>This class is required for Batch Administration. Therefore, it should not be renamed or deleted if this capability is required.</p>
IM	Implementor	<p>Class default is set to ALL, allowing access to all areas in OLA.</p>
LA	Local Administrator	<p>Class default is set to ALL allowing access to all areas in OLA but with the restriction that access is not allowed to OLA security settings or to System settings. Therefore, users in this class can <i>not</i>:</p> <ul style="list-style-type: none"> <li>▪ Modify the OLAClass or AUTH parameter, which would allow them to override their security.</li> <li>▪ Copy any USER statement with a name prefixed with either IM (for example, IMJOE or IMSUE) as these users are assumed to be implementors, or AD (for example, ADFRED or ADMARY) as these users are assumed to be administrators.</li> </ul> <p><b>CAUTION</b> Although the supplied examples stop the copying of USER or PROFILE statements prefixed with IM or AD, you should also ensure that class LA does not have access to any other USER or PROFILE statements that contain a higher security setting for parameters OLAClass or AUTH.</p>
SU	Super User	<p>Class default is set to NONE restricting access to all areas in OLA but with these accesses defined:</p> <ul style="list-style-type: none"> <li>▪ Users in this class have access to all of the 'My USER Definition' settings with these exceptions: <ul style="list-style-type: none"> <li>▪ PASSWORD and TRACE are not displayed.</li> </ul> </li> <li>▪ Users can not modify the OLAClass or AUTH parameter, which would allow them to override their security.</li> <li>▪ Display access is defined for the PDS(E) allocated to the DDNAMEs of TERMINAL, PROFILE, APPL and SYSTEM, allowing the default settings/inheritance values to be displayed for the above parameters.</li> <li>▪ Users in this class can make permanent and temporary modifications and have the option to activate changes immediately or defer activation of changes.</li> <li>▪ Users in this class can display Activate messages.</li> </ul>

Class	Category	Description
US	User	<p>Class default is set to NONE restricting access to all areas in OLA but with these accesses defined:</p> <ul style="list-style-type: none"> <li>▪ In the 'My USER Definition' settings, users in this class can modify the common enduser parameters Backward, Cut, Escape, Forward, HCRequest, Paste, Previous, Push and Pull.</li> <li>▪ Display access is defined for the PDS(E) allocated to the DDNAMEs of TERMINAL, PROFILE, APPL and SYSTEM, allowing the default settings/inheritance values to be displayed for the above parameters.</li> <li>▪ Users in this class can make permanent modifications to the above parameters and the changes will be activated immediately when saved.</li> <li>▪ Users in this class can display Activate messages.</li> </ul>
NO	No Access	<p>Class default is set to NONE restricting access to all areas in OLA.</p> <p>This class is used as a default class. Therefore, it should not be renamed or deleted.</p>

### Notes

- 1 All users in the same OLA security class have the same capability.
- 2 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.
- 3 The example OLA security classes and USER/PROFILE naming schema are unlikely to match the requirements of your Installation. Therefore, when you set up your own OLAClass attributes, you must ensure that a given OLA security class does not have access to any USER or PROFILE statements that contain a higher security setting for parameters OLAClass or AUTH.
- 4 When running OLA, specialized security settings are used to define completely what the user is allowed to do. For details, see 'OLA security settings' on page 32.
- 5 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 the SYSTEM statement section in the *Technical Reference* manual.

### How to assign a user to an OLA security class

To assign a user to an OLA security class, either:

- specify the common enduser parameter OLAClass, which is applicable to the USER, RUSER, TERMINAL, PROFILE and SYSTEM configuration control statements (for details, see the *Technical Reference*)
- or
- if your OLA security class allows it, use OLA (see 'Display list of OLA security definitions' on page 57)
- or
- use an ESM to automatically set the security class (see the OLARESNAME subparameter of the SECURITY parameter in the SYSTEM statement chapter of the *Technical Reference* manual)

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.

## OLA security settings

When running OLA, specialized security settings are used to define completely what the user is allowed to do. An OLA security setting may optionally be qualified by a value to limit its scope; the value itself may optionally contain masking characters.

Here is a list of the allowed OLA security settings:

OLA security setting	Description
ALL	No restriction. In this case, the settings DISPLAY, CREATE, MODIFY and DELETE are implied – see below.
DISPLAY	Indicates that a menu or parameter can be displayed. See Note 1 below.
CREATE NOCREATE	Controls whether or not a new entry can be created.
MODIFY NOMODIFY	Controls whether or not an entry or parameter value can be modified.
DELETE NODELETE	Controls whether or not an entry or parameter value can be deleted.
NONE	Full restriction.

### Notes

- 1 DISPLAY is implied by CREATE, MODIFY, DELETE or ALL.
- 2 OLA security settings are not validated in OLA, so care should be taken when specifying their values.

The OLA security settings are processed one at a time to determine whether or not the required task can be performed. Where there is a conflict, an OLA security setting processed later overrides one processed earlier. An OLA security setting of NONE disallows *all* access, even if allowed by an earlier setting.



## Setting up OLA security

OLA security class definitions are stored in the PDS(E) allocated to the DDNAME of CLASS (see the 'Configuration DDNAMES' section in the 'Converting a Classic to an OLA system' chapter and the 'Customer, samples and system configuration datasets' section in the 'Performing a basic configuration' chapter of the *Installation and Customization* manual).

If your OLA security class allows it, to modify these definitions:

- 1 Select the option **Security definitions** from the OLA Main Menu (see page 50). The Security Definition 'list' panel will be displayed.
- 2 This panel, which is similar to the Profile Definition 'list' panel (see page 51), enables you to specify a filter to be used to subset the entries on the List Security Classes 'list' panel, and default values for List operations. From this panel, select the appropriate OLA security class to be edited.
- 3 The Edit Security Class menu will be displayed, which contains options to maintain the various categories of security definition:

```

Session Manager - Admin      Edit Security Class      dd/mm/yyyy hh:mm:ss
LU      luname              user

_ Menu item controls
_ Security definitions for "My USER definition"
_ Security definitions for "PROFILE definitions"
_ Security definitions for "Local USER definitions"
_ Security definitions for "Remote USER definitions"
_ Security definitions for "APPL definitions"
_ Security definitions for "TERMINAL definitions"
_ Security definitions for "GROUP definitions"
_ Security definitions for "Hardcopy definitions"
_ Security definitions for "SYSTEM definitions"
_ Miscellaneous entries

```

- 4 Select an entry from the Edit Security Class menu and further panels will be displayed which show the OLA security settings associated with particular items. For example, select **Access to menu items** and this panel will be displayed:

```

Session Manager - Admin      Edit Security Class      dd/mm/yyyy hh:mm:ss
LU      luname              user

  Menu item                      Setting ( CLASS or inherited )
_ My USER definition            ALL
_ PROFILE definitions           ALL
_ Local USER definitions        ALL
_ Remote USER definitions       ALL
_ APPL definitions              ALL
_ TERMINAL definitions          ALL
_ GROUP definitions             ALL
_ Hardcopy definitions          ALL
_ SYSTEM definitions            ALL
_ Security definitions          NONE
_ View Activate messages        ALL

```

### See

- 'OLA security settings for menu items' on page 34
- 'OLA security settings for parameter definitions' on page 34
- 'OLA security settings for miscellaneous entries' on page 34

## OLA security settings for menu items

To allow an OLA security class to display a menu item, specify `DISPLAY` (without a value) or `ALL` (`DISPLAY` is implied by `ALL`) as the OLA security setting.

To prevent an OLA security class from displaying a menu item, specify `NONE` as the OLA security setting. For example, to prevent the option `Remote USER definitions` from appearing on the OLA Main Menu:

- 1 Select **Access to menu items** from the Edit Security Class menu.
- 2 Change the setting for **Remote USER definitions** to **NONE**:

Menu item	Setting ( CLASS or inherited )
— My USER definition	ALL
— PROFILE definitions	DISPLAY
— Local USER definitions	DISPLAY
— Remote USER definitions	NONE
— APPL definitions	DISPLAY

## OLA security settings for parameter definitions

The OLA security settings that you can use for parameter definitions are listed in ‘OLA security settings’ on page 32. If access to a parameter is not allowed then it will not be shown to the user.

**Note** Only the parameter name is security checked, not its value, so do not specify an additional value in the OLA security setting.

For example, to prevent the parameter `TRACE` appearing in the display of the user’s enduser specific attributes:

- 1 Select **Security definitions for "My USER definition"** from the Edit Security Class menu.
- 2 Select **Parameters for "Display enduser specific parameters"**, and then set **TRACE** to **NONE**:

Parameter	Setting ( CLASS or inherited )
— PROF(s)	ALL
— PASSWORD	ALL
— TRACE	NONE

**Note** Unlike the OLA security settings for menu items, there is a single setting for each common parameter regardless of where it occurs. For example, if you set the OLA security settings for the common user parameter `ESCAPE` to `NONE` then this parameter will not appear in the user’s common parameters, or on `PROFILES`, `SYSTEMS`, and so on.

## OLA security settings for miscellaneous entries

### See

- ‘Parameters for “File I/O”’ on page 35
- ‘Parameters for “Activation of entries”’ on page 38
- ‘Parameters for “Adding, Copying and Deleting sessions”’ on page 38
- ‘Parameters for “Update own USER statement”’ on page 38
- ‘Parameters for “Class default”’ on page 38
- ‘Parameters for “Prioritizing Sessions”’ on page 38

## Parameters for "File I/O"

This option controls access to members in the various OLA PDS(E) datasets (see the 'Configuration DDNAMEs' section in the 'Converting a Classic to an OLA system' chapter and the 'Customer, samples and system configuration datasets' section in the 'Performing a basic configuration' chapter of the *Installation and Customization* manual). Here the member name can be used as an additional 'scoping' value. Standard Session Manager generic values can be used.

For example, to allow access to users whose names start with ABC:

- 1 Select **Miscellaneous entries** from the Edit Security Class menu.
- 2 Select **Parameters for "File I/O"**, and then set **USER PDSE** to **NONE ALL(ABC\*)**:

Parameter	Setting ( CLASS or inherited )
USER PDSE	NONE ALL(ABC*)
PROFILE PDSE	ALL
TERMINAL PDSE	ALL

To allow the inheritance values to be calculated when updating a user statement, the user will require **DISPLAY** access to the PDS(E) allocated to the DDNAMEs of **TERMINAL**, **PROFILE**, **APPL** and **SYSTEM**, as well as the PDS(E) allocated to the DDNAME of **USER**. In order to make temporary updates the user will require at least **MODIFY** access to the PDS(E) allocated to the DDNAME of **TEMPUSER**.

**Note** From Session Manager release 2.2.00 all messages are prefixed with **ISZM** (for example, **ISZM1234**), all classes are prefixed with **ISZCLS** (for example, **ISZCLSIM**) and all commands are prefixed with **ISZ** or **ISZZ** (for example, **ISZUPDT** and **ISZZINIT**).

The conversion table below translates the Session Manager command names into the **ISZ/ISZZ**-prefixed PDSE member name used from release 2.2.00. These names should be used when specifying member names as an additional 'scoping' value.

Pre-2.2.00 Name and 2.2.00 Command Name	2.2.00 PDSE Member Name
ADDS	ISZZADDS
BAT	ISZZBATS
BLOCK	ISZZBLOC
BRECEIVE	ISZZBREC
BROADCAST	ISZZBROA
CLOSEDOWN	ISZZCLOS
DELETE	ISZZDELE
DELSESS	ISZZDELS
DEMO	ISZZDEMO
DISCONNECT	ISZZDISC
DLOG	ISZZDLOG
DRIVEXIT	ISZZDRIV
DSTORE	ISZZDSTO

Pre-2.2.00 Name and 2.2.00 Command Name	2.2.00 PDSE Member Name
DTERM	ISZZDTER
DUMP	ISZZDUMP
FHELP	ISZZFHEL
FILT	ISZZFILT
FILTER	ISZZFLTR
FIND	ISZZFIND
FLASH	ISZZFLAS
FORCE	ISZZFORC
GFS	ISZZGFSS
HALTSCRIPT	ISZZHALT
HARDCOPY	ISZZHARD
HCOPTION	ISZZHCOP
HELP	ISZZHELP
INITSC	ISZZINIT
INQUIRE	ISZZINQU
ISZACTV	ISZACTV
ISZDIAG	ISZDIAG
ISZLACB	ISZLACB
ISZOLAEN	ISZOLAEN
ISZREMC	ISZREMC
ISZRSVRD	ISZRSVRD
ISZSPQCT	ISZSPQCT
ISZSPSWB	ISZSPSWB
ISZSRCH	ISZSRCH
ISZTACBS	ISZTACBS
ISZTRECD	ISZTRECD
ISZUPAPL	ISZUPAPL
ISZUPDT	ISZUPDT
ISZUPDTE	ISZUPDTE
LOCK	ISZZLOCK
LOGOFF	ISZZLOGO
MSG	ISZZMSGG
MSGID	ISZZMSGI
OLASACTV	ISZZOLAS

<b>Pre-2.2.00 Name and 2.2.00 Command Name</b>	<b>2.2.00 PDSE Member Name</b>
PANELID	ISZZPANE
PASSFREE	ISZZPASS
PCTTRANSFER	ISZZPCTR
PUPDATE	ISZZPUPD
QACTUSER	ISZZQACT
QQUIT	ISZZQQUI
QTASK	ISZZQTAS
QUERY	ISZZQUER
QUIT	ISZZQUIT
QUSER	ISZZQUSE
RECORD	ISZZRECO
REMOVEUSER	ISZZREMO
REPLAY	ISZZREPL
RESET	ISZZRESE
RETRIEVE	ISZZRETR
SECFRESH	ISZZSECF
SEND	ISZZSEND
SME	ISZZSMEE
SPIN	ISZZSPIN
SPY	ISZZSPYY
SPYOFF	ISZZSPYO
STARTTCP	ISZZSTCP
STARTLINK	ISZZSLIN
STARTSC	ISZZSSCC
STOP	ISZZSTOP
STOPACB	ISZZSACB
STOPLINK	ISZZSLIN
STOPTCP	ISZZSOCP
SWITCHPLX	ISZZSWIT
TERMINATE	ISZZTERM
TRACE	ISZZTRAC
TRACEACB	ISZZTACB
TTPSL	ISZZTTPS
UPDATE	ISZZUPDA

Pre-2.2.00 Name and 2.2.00 Command Name	2.2.00 PDSE Member Name
VALUE	ISZZVALU
VIEW	ISZZVIEW
VTM	ISZZVTMM
WINDOWS	ISZZWIND

### Parameters for "Activation of entries"

This option controls whether or not a user can activate changes immediately or defer activation of changes.

If the user has access to both functions then a prompt will be displayed on saving an entry.

By defining an additional value the scope can be limited to specified DDNAMEs. For example, to force immediate activation of entries in the PDS(E) allocated to the DDNAME of RUSER, specify:

Parameter	Setting ( CLASS or inherited )
Defer Act	ALL NONE(RUSER)
Immediate Act	ALL

### Parameters for "Adding, Copying and Deleting sessions"

This option controls the ability to add, copy and delete sessions by selecting **My USER definition**, **Local USER definitions**, **PROFILE definitions** or **TERMINAL definitions** from the OLA Main Menu.

Specifying ALL as the OLA security setting allows this ability and specifying NONE restricts it.

### Parameters for "Update own USER statement"

This option controls file access in the specific situation of a user updating their own USER statement. Specifying ALL as the OLA security setting allows this ability and specifying NONE restricts it.

The **Temp save** setting controls whether or not the user is allowed to perform temporary changes. Specifying ALL as the OLA security setting allows this ability and specifying NONE restricts it.

### Parameters for "Class default"

This option allows you to specify a default OLA security setting for entries in this security class.

**Note** This setting is completely ignored for any entry that has its own setting.

### Parameters for "Prioritizing Sessions"

This option controls whether the column that allows users to move sessions to the top of their Session List will appear in OLA. Unless the user is a member of an OLA class with DISPLAY authority, the column, headed **Priority**, will not be displayed.

### Parameters for "Autostarting Sessions"

This option controls whether the column that allows users to set the Autostart setting in their Session List will appear in OLA. Unless the user is a member of an OLA class with DISPLAY authority, the column (headed **AUTOSTART**) will not be displayed.

## CHAPTER 4

# Menus, lists and displays

This chapter describes each of the main OLA menus, lists and displays.

### Notes

- 1 The facilities available to a particular user depend on that user's OLA security class. For details, see 'Security considerations' on page 27.
- 2 Hidden application sessions (see page 20) will only be shown in OLA menus, lists and displays if the user's OLA security class allows it.
- 3 The External security exit and its Session Manager configuration settings, if installed, will determine if hidden application sessions will be shown in OLA menus, lists and displays.
- 4 Escapes are not permitted from some panels (in which the command line prefix is shown as --->). This is to allow for an escape sequence to be entered as *data*. Those panels where escapes *are* allowed have the usual format (==>) for the prefix.

### See

- 'Maintaining configuration definitions: overview' on page 41  
New users of OLA should review this section, which describes:
  - 'How to use OLA menus' on page 41
  - 'How to use OLA lists' on page 42
  - 'How to use OLA 'Attribute Expansion' displays' on page 45
  - 'Keywords used for display purposes only' on page 47
  - 'Summary of selection options and commands' on page 47
- 'Main Menu' on page 50
- 'USER Definition menu' on page 58
- 'Hardcopy Configuration Statements menu' on page 71
- 'System Configuration Statements menu' on page 73

### What product configuration data can be maintained by OLA?

OLA can be used to maintain all product configuration data, except for:

- Specialized definitions that are stored in the ISZCONxx members (APPLYSU, AUDITROUTE, COPY, INSTALLSU, OPTION, REMOVESU, TRACEROUTE, PATCH, PATCHSU and TRANSTABLE; and Session Manager panels and scripts.). To update these definitions, you must edit the members directly (see the *Installation and Customization* manual).
- If a system definition or sample definition is modified then it will be stored in the corresponding customer dataset. Unlike customer definitions, system definitions and sample definitions can not be deleted by OLA. In other words, a definition stored in a system or sample dataset can not be modified or deleted, although it can be replaced by a modified version which is stored in the corresponding customer dataset.

### Notes

If you delete a modified system or sample definition (stored in the corresponding customer dataset) then the original system or sample definition will be reinstated. However, the reinstated system or sample definition will not be used until it is activated manually, or Session Manager is stopped and restarted. For more information, see the 'Customer, samples and system configuration datasets' section in the 'Performing a basic configuration' chapter of the *Installation and Customization* manual.

To delete the supplied samples an external utility like TSO/ISPF should be used.



## Maintaining configuration definitions: overview

This section briefly describes how to use OLA menus, lists and displays to maintain Session Manager configuration data.

### Notes

- 1 Full descriptions of each Session Manager configuration statement can be found in the *Technical Reference*.
- 2 Customer configuration definitions are split from system and sample configuration definitions, any definitions added or modified by OLA will be held in the *customer* datasets. Further, if a system or sample definition is modified then it will be stored in the corresponding customer dataset; unlike customer definitions, system and sample definitions can *not* be deleted. For more information, see the 'Customer, samples and system configuration datasets' section in the 'Performing a basic configuration' chapter of the *Installation and Customization* manual.

### See

- 'How to use OLA menus' below
- 'How to use OLA lists' on page 42
- 'How to use OLA 'Attribute Expansion' displays' on page 45
- 'Keywords used for display purposes only' on page 47
- 'Summary of selection options and commands' on page 47

## How to use OLA menus

OLA menus are used to show the options available to you, based on your OLA security class (see 'Security considerations' on page 27). Some menu options and their selection numbers may not display if your OLA security settings have excluded them.

This example shows the Main Menu:

```

Session Manager - Admin          Main Menu          dd/mm/yyyy hh:mm:ss
LU  luname                      user

1  _  My USER definition          Filter or name:
2  _  PROFILE definitions          _____
3  _  Local USER definitions      _____
4  _  RUSER definitions           _____
5  _  APPL definitions            _____
6  _  TERMINAL definitions        _____
7  _  GROUP definitions           _____
8  _  Hardcopy definitions        _____
9  _  System definitions          _____
10 _  Security definitions        _____

Command ==>
Available options: S - Select (default)
PF1 Help      2      3 Quit      4 Actmsg      5      6
PF7 Backward  8 Forward  9      10      11      12

```

### How to select a menu option

To select an option from an OLA menu, either:

- type **S** (= **Select**) in the input field against the required option and press Enter.  
or
- tab the cursor to the required option and press Enter  
or
- enter a selection number in the command line and press Enter  
or
- enter a filter in the relevant **Filter or name** field. If the specified filter exactly matches a member name then that member is selected and displayed without displaying a member list: otherwise a filtered member list is displayed.

### How to use OLA lists

OLA lists are used to show either:

- a group of related attribute definitions (for example, the user's common enduser attributes, the user's common session attributes, or SYSTEM definition attributes)  
or
- all of the entries, or a subset of the entries (based on a user-specified filter), for one type of configuration definition (for example, TERMINAL, PROFILE or APPL).

#### See

- 'List of a group of related attributes' below
- 'List of entries for one type of configuration definition' on page 44

### Filtering lists

Filtering is available as a function within member and session lists. 'Top' and 'Bottom' commands are added to enable rapid movement within the lists. PF Key 10 is set to 'Top', and PF Key 11 is set to 'Bottom'. In the case of member lists (not session lists) if the specified filter exactly matches a member name then that member is selected and displayed: otherwise a filtered member list is displayed.

### Identifying eligible members for a list

Whenever a user is able to enter the name of a member as a parameter, it is possible to create a list of eligible members in a selection panel.

Do this by entering the '?' character in the field as a request to list eligible members.

Selection lists may be generated in this way for the following statements: APPL, CLASS, HCFORMAT, HCPROF, HCRROUTE, LINK, PROFILE, RANGE, USSCRIPT.

### Parameter-specific help

In addition to the general help available when no parameter is selected, parameter-specific help is available on parameter list displays. Entering 'H' for help against any displayed parameter will produce help about that specific parameter. The user can also move the cursor to a displayed parameter and press PF Key 1 for parameter-specific help.

**FIND function**

Member and Session lists have a FIND function available. This operates in the same way as FIND command on the main menu.

When used on a session list, the FIND function searches the list for the next match of the specified APPLID. When used on a member list, the FIND function searches the list for the next match of the specified member name. The wildcards '\*' (meaning any character or set of characters) and '+' (meaning any single character) can be used within the APPLID or member name. The APPLID or member name can be between 1 and 8 characters.

If the FIND function is issued without an APPLID or member name, the list will be scrolled to display the next match using the APPLID or member name previously specified.

For further information on the FIND command, see the *Technical Reference* manual.

**List of a group of related attributes**

Here is an example of an OLA list that shows the user's common enduser (that is, non-session) attributes:

```

Session Manager - Admin   Common Enduser Parameters   dd/mm/yyyy hh:mm:ss
LU   lname               user

  Parameter              Setting ( USER or inherited )
- ACTIVESESSIONS        9999
- AFFINITY               S09KL010
- AUTH                   9
- AUTOSELECT             3
- AUTOSEQ                +*#.
- BACKWARD               <<
- BRECEIVE               YES
- CURESC                 NO
- CUT                    *c NFILLC YES
- DEMO                   AUTOCOPY YES INTERNAL NO DISPLAY YES
- ESCAPE                 ;;
- FORWARD               >>
- HCPROF                 hcprofb
- HCREQUEST              /H

                                     >>>>

Command ==>
Available options: S - Select (default)  D - Delete  R - Reset  H - Help
PF1 Help      2      3 End      4 Actmsg    5      6
PF7 Backward  8 Forward  9      10      11      12 Cancel

```

If your OLA security class allows it (see 'Security considerations' on page 27), new attribute definitions can be added, or existing attribute definitions can be viewed, modified, reset or deleted. For details, see 'Display user's common enduser parameters' on page 60.

**Highlighting of attribute definitions**

Depending on the characteristics of your display device, either different colors or highlighting are used to distinguish between attribute definitions set by the user, and attribute definitions inherited from other configuration definitions.

**Multi-page attribute definition lists**

For a multi-page attribute definition list, scroll indicators are displayed in the bottom right hand corner of each page:

>>>> Indicates you can page forward to the next page of attributes.

<<<<< Indicates you can page backward to the previous page of attributes.

List of entries  
for one type of  
configuration  
definition

Here is an example of an OLA list that shows a subset of the PROFILE definitions, based on the user-specified filter PROF:

```
Session Manager - Admin      List PROFILE Definitions      dd/mm/yyyy hh:mm:ss
LU      luname              user

_ Default options
  Filter: _____

_ PROFA      _____
_ PROFB      _____
_ PROFC      _____
_ PROFD      _____
_ PROFDEF    _____
_ PROFE      _____
_ PROFUSER   _____

Command --->
Available options: S/E - Edit  C - Copy  D - Delete  R - Rename  A - Activate
PF1 Help          2          3 Quit      4 Actmsg   5 Refresh  6 Add
PF7 Backward      8 Forward  9          10 Top     11 Bottom  12
```

If your OLA security class allows it (see ‘Security considerations’ on page 27), new entries can be added, or existing entries can be viewed, copied, modified, renamed or deleted. For details, see ‘Display list of PROFILE definitions’ on page 51.

**Note** If you select an entry for deletion, you may be prompted to confirm the deletion (see ‘Confirmation of deletion?’ on page 45).

Although members 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 Activate (PUPDATE) facility; the only definitions that are removed are those which are replaced by new definitions with a duplicate name. For further information see the description of the PUPDATE command in the *Technical Reference* manual.

To reread the configuration, press PF Key 5. Session Manager redisplayes the list from the top.

Multi-page configuration definition lists

For a multi-page configuration definition list, scroll indicators are displayed in the bottom right hand corner of each page:

- >>>> Indicates you can page forward to the next page of definitions.
- <<<<< Indicates you can page backward to the previous page of definitions.

Immediate activation of changes?

To specify that subsequent changes to the definitions are to be activated immediately, type **Y** against the **Immediate activate** prompt on the configuration definition ‘Default options’ panel. This is a system-wide setting. For details, see ‘Profile Definition ‘default options’ panel’ on page 53.

### Confirmation of deletion?

To specify that a subsequent deletion of a configuration definition requires confirmation, type **Y** against the **Delete confirmation** prompt on the configuration definition 'Default options' panel. This is a system-wide setting. For details, see 'Profile Definition 'default options' panel' on page 53.

If you subsequently select a configuration definition for deletion then a panel will appear prompting you to confirm the deletion. Here is a typical example of the panel that appears:

```

Session Manager - Admin   Confirm Delete of Entry           dd/mm/yyyy hh:mm:ss
LU  luname               user

You are about to delete

Entry:
PROFA

Type :
PROFILE

Y Continue to receive delete confirmation (y/n)

Press ENTER to confirm that you want to delete
Press QUIT to cancel the delete

Command ==>

PF1 Help      2          3 Quit      4          5          6
PF7           8          9          10         11         12

```

**Note** You can also type **Y** against the **Continue to receive delete confirmation** prompt on this panel to indicate that a subsequent deletion of a configuration definition requires confirmation.

On the Confirm Delete of Entry panel, either press the Enter key to confirm the deletion or issue the QUIT command to cancel the deletion request.

## How to use OLA 'Attribute Expansion' displays

If the user's OLA security class allows it (see 'Security considerations' on page 27), an OLA 'Attribute Expansion' display enables an attribute definition to be modified.

Here is an example of a user 'Attribute Expansion' display:

```

Session Manager - Admin      AUTOSELECT Expansion      dd/mm/yyyy hh:mm:ss
LU      luname              user
Parameter: AUTOSELECT
USER setting: 3

To modify this parameter, enter a new value and optional comment below:

Value: 3
Comment: _____

Activation: Effective at next signon or reconnect

Command --->

PF1 Help      2 Reset      3 End      4 Actmsg      5 Delete      6
PF7 Backward  8 Forward      9          10 Previous  11 Next      12 Cancel

```

From here, the user is prompted to add the attribute definition if it does not already exist, or to modify the user attribute definition if it does exist. For details, see ‘User ‘Attribute Expansion’ display’ on page 62.

### Notes

- 1 Some attribute definitions may have multiple occurrences and some may have sub-attributes which, when selected, will navigate you to additional sub-menus.
- 2 Some attribute definitions allow one or more values to be entered as a list of values. The list may be wrapped using one of the standard delimiters: a single quotation mark (‘), a double quotation mark (“), an exclamation mark (!), or a question mark (?). Alternatively, matched parentheses ( ... ) may be used. If a delimiter is used at the start of the list then a corresponding delimiter must be specified at the end of the list.

### Highlighting of attribute definitions

Depending on the characteristics of your display device, either different colors or highlighting are used to distinguish between attribute definitions set by the user, and attribute definitions inherited from other configuration definitions.

## Escape processing

All escape processing will be disabled when displaying the ‘Attribute expansion’, ‘Session list’ and ‘Member list’ panels. This will allow the user to enter and save escape sequence values as valid attribute and filter data. The user must quit the ‘Attribute expansion’, ‘Session list’ or ‘Member list’ display in order to action any of the escape sequence processing.

## Parameter validation

When entering a value for an existing parameter, it is not possible to enter spaces to blank out the parameter. The value for an existing parameter must contain data. If a parameter is no longer required, it should be deleted.

## Keywords used for display purposes only

The Session Manager configuration allows for certain statements to have mutually exclusive values. In order to present these in a user-friendly manner in OLA, these additional keywords are used solely for display purposes.

Additional keyword	Configuration statement	Descriptions
BASE	RANGE	Added for the HEX or DEC values.
COMMENT	Optionally on all statements	OLA allows for comments only on first level keywords; comments are <i>not</i> allowed on sub-parameters. Therefore, in a sub-parameter display, the COMMENT keyword has been added to allow modification of the comment.
DATEFORMAT	SYSTEM	Added for the MDY, DMY, YMD or YDM values.
IDLETYPE	Common enduser parm IDLELOGOFF	Added for the AFTERDISC value.
INHERIT	Common session sub-parm for ADDSID	Added for the STOP_INHERIT value.
LINKTYPE	LINK	Added for the ISZ value.
SELCMDS	PROFILE	Added for the SESSION selection commands ( <i>selection-commands</i> ) or pseudonyms.
TYPE	Common session parm	Added for the UNDERISZSMGR, CLOSEDISC and CLOSELOGOFF values.

### Notes

- 1 These additional keywords are *not* valid Session Manager keywords and will be rejected if found in the corresponding configuration statements.
- 2 Full descriptions of all Session Manager configuration statements and keywords can be found in the *Technical Reference*.

## Summary of selection options and commands

OLA selection options and commands are summarized below.

### Selection options

Here is the complete list of OLA selection options:

Option	Action
A	Activate changes to the item.
C	Copy the item.
D	Delete the item.
E	Edit the item.

Option	Action
H	Display parameter context help. Alternatively, the cursor can be positioned on a parameter line and PF1 (Help) pressed for parameter context help.
I	Insert a new 'dummy' item after this entry. (If the new entry is not modified it will not be saved.)
P	Create a duplicate of the item after this entry.
R	Depending on the context, either: <ul style="list-style-type: none"> <li>▪ rename the item</li> <li>or</li> <li>▪ reset the item to its original value.</li> </ul>
S	(The default value.) Select the item.  <b>Note</b> If you tab the cursor to the required item and press Enter then the item will be selected by default.

### Commands

The table below shows all the commands (in alphabetical order), most of which (not Find, for example) are represented by PF Key settings. For each OLA panel, PF Key settings pertaining to the panel are displayed at the bottom of the screen.

Setting	Action
ActMsg	Display the messages from the last activate action.
Add	(For list displays.) Add a new entry.
Backward	Page backward in the display to the previous page of definitions.
Bottom	Page to the bottom item within this display. This command can be abbreviated to BOT.
Cancel	Discard changes and return to member list or (if in My USER) to the Main Menu.
Delete	Delete the (user-defined) attribute.
End	Return to next highest level.
Find	Search the definitions for the next match of the specified APPLID.
Forward	Page forward in the display to the next page of definitions.
Help	Enter the OLA online Help.
Next	Display the next attribute.
Previous	Display the previous attribute.
Quit	Return to next highest level (or leave OLA if on Main Menu)
Refresh	Rereads the configuration.
Reset	Reset the attribute to its original value.



Setting	Action
Resetas	Resets any modifications made to the Autostart settings.
Resetpri	Resets any modifications made to the session priority.
Save	Commit changes and return to next highest level.
Source/ Messages	'Toggle' between the messages from the last activate action and the corresponding configuration source.
Top	Page to the top item within this display.
Userview	Display the 'Userview' session list panel.

## Main Menu

Here is the OLA Main Menu:

```

Session Manager - Admin          Main Menu          dd/mm/yyyy hh:mm:ss
LU  luname                      user

1  _  My USER definition        Filter or name:
2  _  PROFILE definitions
3  _  Local USER definitions
4  _  RUSER definitions
5  _  APPL definitions
6  _  TERMINAL definitions
7  _  GROUP definitions
8  _  Hardcopy definitions
9  _  System definitions
10 _  Security definitions

Command ==>
Available options: S - Select (default)
PF1 Help      2          3 Quit      4 Actmsg      5          6
PF7 Backward  8 Forward  9          10         11         12

```

**Note** The options available to a particular user depend on that user's OLA security class (see 'Security considerations' on page 27).

### See

- 'Display My USER definition menu' below
- 'Display list of PROFILE definitions' on page 51
- 'Display list of Local USER definitions' on page 54
- 'Display list of RUSER definitions' on page 55
- 'Display list of APPL definitions' on page 55
- 'Display list of TERMINAL definitions' on page 56
- 'Display list of GROUP definitions' on page 56
- 'Display Hardcopy Configuration Statements menu' on page 56
- 'Display System Configuration Statements menu' on page 56
- 'Display list of OLA security definitions' on page 57
- 'View Activate messages' on page 57

## Display My USER definition menu

Select the option **My USER definition** from the Main Menu (see above) to maintain your own attribute definitions.

The USER Definition menu will appear, which contains a separate option for each group of related user attributes (common enduser attributes, common session attributes, and so on).

For details, see 'USER Definition menu' on page 58.

## Display list of PROFILE definitions

If your OLA security class allows it (see ‘Security considerations’ on page 27), select the option **PROFILE definitions** from the Main Menu (see page 50) to maintain PROFILE definitions.

The maintenance tasks include:

- List existing PROFILE definitions.
- Add a new PROFILE definition.
- Copy an existing PROFILE definition.
- Modify an existing PROFILE definition.
- Rename an existing PROFILE definition.
- Delete an existing PROFILE definition.

If you select a PROFILE definition for deletion, you may be prompted to confirm the deletion (see ‘Delete confirmation?’ on page 54).

### See

- ‘Profile Definition ‘list’ panel’ on page 51
- ‘Saving changes to PROFILE definitions’ on page 54

## Profile Definition ‘list’ panel

The Profile Definition ‘list’ panel will show all PROFILE definitions, or a subset of the PROFILE definitions specified using the **Filter**.

The Profile Definition ‘list’ panel has some default values that determine whether changes to definitions are activated immediately, and whether deletion of a definition requires confirmation. You may enter the default options panel at any time using the option on the list panel (see ‘Modifying default options for the Profile Definition ‘list’ panel’ on page 53).

### Filtering entries

To subset the entries on the Profile Definition ‘list’ panel, type your selection pattern in the **Filter** input field, or leave blank to select all PROFILEs. Wildcard entries may be used:

- \* An asterisk indicates any number of any characters.
- + A plus sign indicates any single character in that position.

### Example Selects

- \*ABC Names ending in ABC.
- \*A\* Names containing a A, possibly surrounded by any other characters.
- ABC\* Names beginning with ABC.
- ++ABC Five-character names ending in ABC.
- +A\* Names at least 2 characters in length with A as the second character.
- ++A\* Names of at least 3 characters in length with A as the third character.

If the specified filter exactly matches a member name then that member is selected and displayed: otherwise a filtered member list is displayed.

Here is an example of an OLA list that shows a subset of the PROFILE definitions, based on the user-specified filter PROF\*:

```
Session Manager - Admin      List PROFILE Definitions      dd/mm/yyyy hh:mm:ss
LU  luname                  user

_ Default options
  Filter:  PROF*

_ PROFA      _____
_ PROFB      _____
_ PROFC      _____
_ PROFD      _____
_ PROFDEF    _____
_ PROFE      _____
_ PROFUSER   _____

Command --->
Available options: S/E - Edit  C - Copy  D - Delete  R - Rename  A - activate
PF1 Help      2      3 Quit      4 Actmsg   5 Refresh  6 Add
PF7 Backward  8 Forward  9      10 Top    11 Bottom  12
```

If your OLA security class allows it (see ‘Security considerations’ on page 27), this list enables you to copy, modify, rename, delete or activate PROFILE definitions.

To reread the configuration, press PF Key 5. Session Manager redisplay the list from the top.

See

- ‘Multi-page configuration definition lists’ on page 52
- ‘Codes for List actions’ on page 52
- ‘Modifying a particular PROFILE definition’ on page 54

Multi-page configuration definition lists

For a multi-page configuration definition list, scroll indicators are displayed in the bottom right hand corner of each page:

- >>>> Indicates you can page forward to the next page of definitions.
- <<<< Indicates you can page backward to the previous page of definitions.

Codes for List actions

Action codes may be entered as line commands for required PROFILE definitions:

- A  
Activate a PROFILE definition. See also ‘Immediate activation of changes?’ on page 53.
  - C  
Copy a PROFILE definition.
  - D  
Delete a PROFILE definition.
- If you select a PROFILE definition for deletion, you may be prompted to confirm the deletion (see ‘Delete confirmation?’ on page 54).

E

Select/edit a PROFILE definition (see ‘Modifying a particular PROFILE definition’ on page 54).

R

Rename a PROFILE definition.

S

Select/edit a PROFILE definition (see ‘Modifying a particular PROFILE definition’ on page 54).

### Modifying default options for the Profile Definition ‘list’ panel

To modify the ‘default options’, select this option within the ‘list’ panel and press Enter, or use the E (= Edit) or S (= Select) action code. The Profile Definition ‘default options’ panel will be displayed.

### Profile Definition ‘default options’ panel

This panel enables you to specify some default values for subsequent List Profile operations.

```

Session Manager - Admin   List PROFILE Definitions   dd/mm/yyyy hh:mm:ss
LU   luname               user

Immediate activate: Y
Delete confirmation: Y

Command ==>

PF1 Help      2          3 Quit      4 Actmsg      5          6
PF7 Backward  8 Forward  9          10         11         12
  
```

### See

- ‘Immediate activation of changes?’ on page 53
- ‘Delete confirmation?’ on page 54

When you have specified suitable values, Quit the ‘default options’ panel and the Profile Definition ‘list’ panel will appear (see page 51).

### Immediate activation of changes?

If your OLA security class allows it (see ‘Security considerations’ on page 27), type **Y** against the **Immediate activate** prompt on the Profile Definition ‘default options’ panel to specify that subsequent changes to PROFILE definitions are to be activated immediately. This is a system-wide setting. See also the A (= Activate) action code in ‘Codes for List actions’ on page 52.

### Delete confirmation?

To specify that a subsequent deletion of a PROFILE definition requires confirmation, type **Y** against the **Delete confirmation** prompt on the Profile Definition ‘default options’ panel. This is a system-wide setting. For more information, see ‘Confirmation of deletion?’ on page 45.

### How to save your changes

For instructions on how to save changes to PROFILE definitions, see ‘Saving changes to PROFILE definitions’ below.

### Modifying a particular PROFILE definition

To modify a particular PROFILE definition select it and press Enter or use the E (= Edit) or S (= Select) action code (see ‘Codes for List actions’ on page 52). The attributes associated with that definition will be shown in an OLA list.

Processing of attributes for a particular PROFILE definition is similar to processing of common enduser attributes, so refer to ‘Display user’s common enduser parameters’ on page 60 for a description of how to proceed.

### Saving changes to PROFILE definitions

At any time you can press PF12 (= **Cancel**) to exit, discarding any pending changes. Otherwise, if changes have been made, you will be prompted to confirm a save of the changes when you exit the Profile Definition menu. From there you can perform one of these actions:

- Issue the END (PF3) command to cancel the save and return.
- Press Enter to save changes and quit.
- Issue the CANCEL (PF12) command to discard all changes and quit.

## Display list of Local USER definitions

If your OLA security class allows it (see ‘Security considerations’ on page 27), select the option **Local USER definitions** from the Main Menu (see page 50) to maintain a particular user’s attribute definitions.

To restrict the list of Local USER definitions to userids which are in the OLA users default ESM group, set the common enduser parameter ESMOLAGROUP to Yes or ON. For more information about this parameter, see the *Technical Reference* manual.

The User Definition menu will appear. Processing of USER definitions is similar to processing of PROFILE definitions, so refer to ‘Display list of PROFILE definitions’ on page 51 for a description of how to proceed.

### Modifying a particular USER definition

If your OLA security class allows it (see ‘Security considerations’ on page 27), to modify a particular USER definition select it and press Enter or use the E (= Edit) or S (= Select) action code (see ‘Codes for List actions’ on page 52).

**Note** When Session Manager has been configured to utilize an External Security Manager (ESM) (for example, RACF) and exit E21 is active then the AUTH and OLACCLASS values will be extracted from the ESM. ESMs can only work with discrete userids, therefore if the userid selected from the Local USER list is generic a valid discrete userid must be entered before the AUTH and OLACCLASS values can be extracted.

The USER Definition menu will appear, which contains a separate option for each group of related user attributes (common enduser attributes, common session attributes, and so on). The USER Definition menu enables you to either:

- view the selected user's attribute definitions
- or
- modify the selected user's attribute definitions.

Processing of attributes for a particular USER definition is similar to My USER definition processing (see page 50); for a description of how to proceed, see 'USER Definition menu' on page 58.

## Display list of RUSER definitions

If your OLA security class allows it (see 'Security considerations' on page 27), select the option **RUSER definitions** from the Main Menu (see page 50) to maintain remote user (that is, RUSER) definitions.

The Remote User Definition menu will appear. Processing of RUSER definitions is similar to processing of PROFILE definitions, so refer to 'Display list of PROFILE definitions' on page 51 for a description of how to proceed.

## Display list of APPL definitions

If your OLA security class allows it (see 'Security considerations' on page 27), select the option **APPL definitions** from the Main Menu (see page 50) to maintain APPL definitions.

The Application Definition menu will appear. Processing of APPL definitions is similar to processing of PROFILE definitions, so refer to 'Display list of PROFILE definitions' on page 51 for a description of how to proceed.

### Notes

- 1** Hidden sessions (see page 20) will only be included in the list of application definitions if the user's OLA security class allows it (see 'Security considerations' on page 27).
- 2** If a particular session does not have an APPL name then the APPLID name will be shown.
- 3** When an APPL definition is copied its SESTYPE is set to 0. If the user-customizable 'priority' session facility is being used, and the user can set the common enduser parameter SESSPRIAPPL to NO (the default), then the APPL will need to have a unique SESTYPE allocated to it, either manually or by using ISZSTJOB. See 'Setting priority sessions' on page 21 for details.

## Display list of **TERMINAL** definitions

If your OLA security class allows it (see ‘Security considerations’ on page 27), select the option **TERMINAL definitions** from the Main Menu (see page 50) to maintain TERMINAL definitions.

The Terminal Definition menu will appear. Processing of TERMINAL definitions is similar to processing of PROFILE definitions, so refer to ‘Display list of PROFILE definitions’ on page 51 for a description of how to proceed.

## Display list of **GROUP** definitions

If your OLA security class allows it (see ‘Security considerations’ on page 27), select the option **GROUP definitions** from the Main Menu (see page 50) to maintain GROUP definitions.

The Group Definition menu will appear. Processing of GROUP definitions is similar to processing of PROFILE definitions, so refer to ‘Display list of PROFILE definitions’ on page 51 for a description of how to proceed.

## Display Hardcopy Configuration Statements menu

If your OLA security class allows it (see ‘Security considerations’ on page 27), select the option **Hardcopy definitions** from the Main Menu (see page 50) to maintain Hardcopy definitions.

The Hardcopy Configuration Statements menu will appear, which enables you to maintain:

- HCPROFILE definitions;
- HCFORMAT definitions;
- HCRROUTE definitions.

For details, see ‘Hardcopy Configuration Statements menu’ on page 71.

## Display System Configuration Statements menu

If your OLA security class allows it (see ‘Security considerations’ on page 27), select the option **System definitions** from the Main Menu (see page 50) to maintain system definitions.

The System Configuration Statements menu will appear, which enables you to maintain:

- RANGE definitions;
- LINK definitions;
- command authority definitions;
- SYSTEM definition attributes;
- MESSAGE definitions.

For details, see ‘System Configuration Statements menu’ on page 73.



## Display list of OLA security definitions

If your OLA security class allows it (see ‘Security considerations’ on page 27), select the option **Security definitions** from the Main Menu (see page 50) to maintain OLA security class definitions.

The Security Definition menu will appear. Processing of SECURITY definitions is similar to processing of PROFILE definitions, so refer to ‘Display list of PROFILE definitions’ on page 51 for a description of how to proceed.

**Note** CLASS definitions are held in PDSE members that are prefixed with ISZCLS. The ‘ISZCLS’ is removed when Classes are displayed within OLA.

## View Activate messages

Users can access Activate messages from the last activate action via the ACTMSG command, or PF Key 4, which is set to ACTMSG. It is possible to ‘toggle’ between the messages from the last activate action and the corresponding configuration source, by pressing the **Source/Messages** PF Key.

## USER Definition menu

If your OLA security class allows it (see ‘Security considerations’ on page 27), select one of these options from the Main Menu (see page 50) to maintain USER definitions:

- **My USER definition**

Select this option to maintain your own attribute definitions. The USER Definition menu will appear:

```

Session Manager - Admin      USER Definition      dd/mm/yyyy hh:mm:ss
LU   Luname                  user

1 _ Enduser specific parameters
2 _ Common enduser parameters
3 _ Common session parameters
4 _ List of sessions

Command ==>
Available options: S - Select (default)
PF1 Help      2      3 End      4 Actmsg      5      6
PF7 Backward  8 Forward  9      10      11      12 Cancel

```

This menu contains a separate option for each group of related user attributes (common enduser attributes, common session attributes, and so on).

- **Local USER definitions**

Select this option to maintain a particular user’s attribute definitions. When you edit a particular USER definition, the USER Definition menu will appear (see above).

### See

- ‘Display enduser specific parameters’ below
- ‘Display user’s common enduser parameters’ on page 60
- ‘Display user’s common session parameters’ on page 65
- ‘Display list of sessions associated with user’ on page 66

### See also

- ‘Saving changes to attributes’ on page 63
- ‘Resetting changes to attributes’ on page 65

## Display enduser specific parameters

If your OLA security class allows it (see ‘Security considerations’ on page 27), select the option **Enduser specific parameters** from the USER Definition menu (see above) to maintain enduser specific parameters (PROFs, PASSWORD, TRACE, and so on).

The enduser specific parameter screen will appear:

```

Session Manager - Admin   Enduser parameters          dd/mm/yyyy hh:mm:ss
LU   luname              user

  Parameter              Setting ( USER or inherited )
- PROF(s)                2 Items specified >>>
- PASSWORD
- RENUMDUP
- TRACE

Command ==>
Available options: S - Select (default)  D - Delete  R - Reset  H - Help
PF1 Help          2          3 End          4 Actmsg    5          6
PF7 Backward      8 Forward  9          10         11         12 Cancel

```

Depending on the path used to navigate to the USER Definition menu, and your OLA security class, this list enables you to either:

- view your own enduser specific parameters and modify your own enduser specific parameters
- or
- view or modify enduser specific parameters associated with the selected user.

Processing of enduser specific parameters is similar to processing of common enduser attributes, so refer to 'Display user's common enduser parameters' below for a description of how to proceed.

#### See also

- 'PROFILE definitions sequence' below

## PROFILE definitions sequence

In a User Profile Definition list, the sequence of the PROFILE definitions is important. If a particular attribute value (common enduser or common session) has not been defined by the user, the list of PROFILE definitions 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

## Extracting PROFILES from the ESM

The system can be configured so that a user's Profiles are determined by the External Security Manager (ESM) and not from the configuration. See the SECURITY parameter on the SYSTEM statement in the *Technical Reference* manual and 'Defining Security' in the *Installation and Customization* manual. Users' Profile settings that have been determined by the ESM cannot be modified within OLA. To modify users' Profile settings you must modify the ESM settings. If the system

has been configured to extract users' Profile settings from the External Security Manager (ESM) then the ESM requires a discrete userid. When you select a generic userid from the Local USER definitions you are presented with a panel that requires you to select one of the following two options:

- 1 Use the generic userid.  
The ESM will not be able to extract the user's Profile settings and therefore OLA will not be able to display any inheritance or sessions from the Profiles. The inheritance shown will be from the SYSTEM statement or the system default. This option should be used if you need to display or update the generic user.
- 2 Use a discrete userid.  
You must update the user field with a discrete userid. This userid will be used by the ESM to extract the user's Profiles. These Profiles will be used to show any inheritance values and Profile sessions. Although OLA is displaying the inheritance and sessions from the discrete user's Profiles you are still displaying and updating the generic user that you selected from the Local USER definitions. If the system has also been configured to determine either a user's AUTH or a user's OLAClass or both from the ESM then this userid will also be used by OLA when invoking the ESM for these parameters.

## Display user's common enduser parameters

If your OLA security class allows it (see 'Security considerations' on page 27), select the option **Common enduser parameters** from the USER Definition menu (see page 58) to maintain common enduser (that is, non-session) attributes associated with the user.

The Common Enduser Parameters list will appear:

```

Session Manager - Admin  Common enduser parameters  dd/mm/yyyy hh:mm:ss
LU  luname  user

Parameter          Setting ( USER or inherited )
- ACTIVESESSIONS    9999
- AFFINITY           S09KL010
- AUTH               9
- AUTOSELECT         3
- AUTOSEQ            +*#.
- BACKWARD           <<
- BRECEIVE           YES
- COMMANDPRFXVAL     NO
- CURESC             NO
- CUT
- DEMO
- DOUBLESC           0
- ESCAPE             ;;
- FORWARD            >>
- HCPROF             hcprofb
- HCREQUEST          /H

>>>>

Command ==>
Available options: S - Select (default)  D - Delete  R - Reset  H - Help
PF1 Help      2      3 End      4 Actmsg    5      6
PF7 Backward  8 Forward  9      10      11      12 Cancel

```

Depending on the path used to navigate to the USER Definition menu, and your OLA security class, this list enables you to either:

- view your own attribute definitions and add, modify, reset or delete your own attribute definitions

or

- view, add, modify, reset or delete the selected user's attribute definitions.

In addition to the standard parameters there are five 32-byte user fields (USERDATA 1 - USERDATA5) and read-only variables (*t\_userdata1* - *t\_userdata4*) for general use by the user. These are available for use in scripts, exits and so on, and are not used internally by Session Manager. There is no validation on the data within these fields and in an OLA display USERDATA5 will display as asterisks, so that more sensitive user data can be entered.

### See

- 'Highlighting of attribute definitions' on page 61
- 'Multi-page user attribute definition lists' on page 61
- 'Changing an attribute definition: action codes' on page 61
- 'Pending changes' on page 62

### See also

- 'User 'Attribute Expansion' display' on page 62
- 'Saving changes to attributes' on page 63
- 'Resetting changes to attributes' on page 65

## Highlighting of attribute definitions

Depending on the characteristics of your display device, either different colors or highlighting are used to distinguish between attribute definitions set by the user, and attribute definitions inherited from other configuration definitions.

## Multi-page user attribute definition lists

For a multi-page user attribute definition list, scroll indicators are displayed in the bottom right hand corner of each page:

>>>>> Indicates you can page forward to the next page of attributes.

<<<<< Indicates you can page backward to the previous page of attributes.

## Changing an attribute definition: action codes

Action codes may be entered as line commands for required attribute definitions:

D

Delete an attribute definition, in which case it will be set to the value inherited from other configuration definitions. See also 'Pending changes' on page 62.

H

Display context sensitive help for this attribute.

R

Reset an attribute definition, in which case the previous actions (add, modify or delete) performed on the attribute will be reversed and the attribute reset to its original value.

S  
(The default value.) Select an attribute to view, add or modify, in which case the user ‘Attribute Expansion’ display will appear (see below). See also ‘Pending changes’ on page 62.

How to save your changes

For instructions on how to save changes to attribute definitions, see ‘Saving changes to attributes’ on page 63.

Pending changes

If one of more attribute definitions have been changed using the ‘Attribute Expansion’ display (see below), a letter will appear to the left of the setting(s) of the effected attribute(s) to indicate that changes are pending:

Parameter	Setting ( USER or inherited )
AFFINITY	A S09KL010
AUTH	D 9
AUTOSELECT	M 3

Letter	Description
A	Attribute has been added.
D	Attribute has been deleted.
M	Attribute has been modified.

For instructions on how to save changes to attribute definitions, see ‘Saving changes to attributes’ on page 63.

User ‘Attribute Expansion’ display

If the user’s OLA security class allows it (see ‘Security considerations’ on page 27), to view, add or modify a particular attribute definition use the **S (= Select)** action code on the Common Enduser Parameters list (see page 60).

The user ‘Attribute Expansion’ display will appear; here is an example:

Session Manager - Admin      AUTOSELECT Expansion      dd/mm/yyyy hh:mm:ss  
LU    luname      user  
Parameter: AUTOSELECT  
USER setting: 3

To modify this parameter, enter a new value and optional comment below:

Value: 3  
Comment: \_\_\_\_\_

Activation: Effective at next signon or reconnect

Command --->

PF1 Help      2 Reset      3 End      4 Actmsg      5 Delete      6  
PF7 Backward    8 Forward    9      10 Previous    11 Next      12 Cancel

From here, the user is prompted to add the attribute definition if it does not already exist, or to modify the user attribute definition if it does exist.

**Notes**

- 1 Some attribute definitions may have multiple occurrences and some may have sub-attributes which, when selected, will navigate you to additional sub-menus.
- 2 Some attribute definitions allow one or more values to be entered as a list of values. The list may be wrapped using one of the standard delimiters: a single quotation mark ('), a double quotation mark ("), an exclamation mark (!), or a question mark (?). Alternatively, matched parentheses ( ... ) may be used. If a delimiter is used at the start of the list then a corresponding delimiter must be specified at the end of the list.

**See**

- 'Saving changes to attributes' below
- 'Resetting changes to attributes' on page 65

**See also**

- 'Highlighting of attribute definitions' on page 61
- 'Changing an attribute definition: action codes' on page 61

**Pending changes**

If the attribute definition is changed, a message will appear above the attribute's value to indicate that a change is pending, for example:

MODIFY pending

Value: 3  
 Comment: \_\_\_\_\_

Message	Description
ADD pending	Attribute has been added.
DELETE pending	Attribute has been deleted.
MODIFY pending	Attribute has been modified.

**Saving changes to attributes**

To save changes to the user attributes, press PF3 (= **End**) repeatedly until you reach a panel where the **Save** PF key becomes available, then press PF3 (= **Save**).

If your OLA security class allows it (see 'Security considerations' on page 27), you will be prompted to specify whether or not changes to the user attributes should be:

- Temporary (that is, the changes will be lost when you sign off).
- Activated immediately.

```

Session Manager - Admin          Modify Entry          dd/mm/yyyy hh:mm:ss
LU  luname                      user

You are about to modify

Entry:
ABC

Type :
USER

N Is this a temporary modification? (y/n)
Y Immediate activation of change? (y/n)

Select the required option
Press ENTER to continue
Press CANCEL to discard all modifications

Command ==>

PF1 Help      2          3 End      4          5          6
PF7           8          9          10         11         12 Cancel

```

### See

- ‘Temporary changes?’ on page 64
- ‘Immediate activation of changes?’ on page 64

### Temporary changes?

To specify that changes to the user attributes are temporary, type **Y** against the **Is this a temporary modification?** prompt.

**Note** Temporary changes can be made *only* to your own definitions when navigated using My USER definition menus; other definitions can *not* be changed temporarily.

If temporary changes are made to you own attribute definitions then:

- If other users are signed on using these user attributes, the application sessions for these other users will also be updated with the changes.
- Whenever these attributes are re-visited, a warning will be displayed to the effect that temporary changes have been made and the changes will be lost when you sign off.
- If your OLA security class allows it (see ‘Security considerations’ on page 27), you will be offered the option to convert the temporary changes into permanent changes by using the SAVE command.

### Immediate activation of changes?

Type **Y** against the **Immediate activation of change?** prompt to specify that changes to the user attributes are to be activated immediately. Activating changes is equivalent to issuing a Session Manager PUPDATE command (for details, see the *Technical Reference*).

**Note** If you specify that changes to the user attributes are *not* to be activated immediately, the changes can subsequently be activated by issuing the PUPDATE command.



## Resetting changes to attributes

To reset a particular attribute definition (that is, reverse the previous actions performed on the attribute and reset it to its original value), use the **R** (= **Reset**) action code on the Common Enduser Parameters list (see page 60). For more information, see ‘Changing an attribute definition: action codes’ on page 61.

## Display user’s common session parameters

If your OLA security class allows it (see ‘Security considerations’ on page 27), select the option **Common session parameters** from the USER Definition menu (see page 58) to maintain common session attributes associated with the user.

The Common Session Parameters list will appear:

```

Session Manager - Admin      Common session parameters      dd/mm/yyyy hh:mm:ss
LU      luname              user

  Parameter              Setting ( USER or inherited )
- ACB                      s09k1001
- ADDSID(s)
- ALARM                    NO
- ALLOWESCAPE              YES
- APPLID
- APPLSEL
- AUTOSCRIP
- AUTOSTART                NO
- BRDVAR
- CMD
- COMPRESS                 YES
- DATA
- DESCRIPTION
- DISCACTIVE               YES

                                                                    >>>>

Command ==>
Available options: S - Select (default)  D - Delete  R - Reset  H - Help
PF1 Help      2          3 End          4 Actmsg   5          6
PF7 Backward  8 Forward  9          10         11         12 Cancel

```

Depending on the path used to navigate to the USER Definition menu, and your OLA security class, this list enables you to either:

- view your own attribute definitions and add, modify, reset or delete your own attribute definitions

or

- view, add, modify, reset or delete the selected user’s attribute definitions.

In addition to the standard parameters there are five 32-byte user fields (SESSDATA 1 - SESSDATA5) and read-only variables (*s\_sessdata1* - *s\_sessdata5*) for general use by the user. These are available for use in scripts, exits and so on, and are not used 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.

There is no validation on the data within these fields and in an OLA display SESSDATA5 will display as asterisks, so that more sensitive user data can be entered.

Processing of common session attributes is similar to processing of common enduser attributes, so refer to ‘Display user’s common enduser parameters’ on page 60 for a description of how to proceed.

## Display list of sessions associated with user

If your OLA security class allows it (see 'Security considerations' on page 27), select the option **List of sessions** from the USER Definition menu (see page 58) to maintain application session definitions associated with the user.

For both My USER displays and Local USER displays this panel is also used to display the 'Admin view' of the session list. See also 'Display User View of session list' on page 69.

You will be able to filter the list if you wish; the filtering process is similar to that used when displaying PROFILE definitions (see 'Filtering entries' on page 51). The filter will check the APPLID of each session for a match. If the session does not have an APPLID specified, the filter will check the CMD parameter.

**Note** If a site is running the E22 exit with external security active, the list of My USER sessions associated with the user is limited to those sessions for which the user has access permission.

The Session List panel will appear:

Session Manager - Admin

LU luname

Session List

Name of entry: IM1

dd/mm/yyyy hh:mm:ss

user

APPLID Filter: \_\_\_\_\_

<u>Key/Number</u>	<u>Priority</u>	<u>Session</u> ( USER or inherited )	<u>AUTOSTART</u>	
			<u>USER</u>	<u>Session</u>
— PF01	1	CMD help		(N)
— PF07	2	APPLID ola	N	(Y)
— PF04	3	CMD disc	N	(N)
— PF03		CMD logoff exit		(N)
— PF08		CMD forward	Y	(N)
— PF12		APPLID ola		(N)
— 0033		APPLID cicso	N	(N)
— 0101		APPLID CICS		(N)
— 0102		APPLID CICS		(N)
— 0103		APPLID IMS		(N)
— 0104		APPLID VM		(N)
— 0105		APPLID TSOA		(N)
— 0106		APPLID TSOB		(N)
— 0111		APPLID user		(N)
— 0240		APPLID COMMAND		(N)
— 0259		APPLID COMMAND		(N)
— 0810		APPLID SMM		(N)

>>>>

Command --->

Available options: S - Select (default) C - Copy D - Delete R - Reset

PF1 Help 2 Resetpri 3 End 4 Actmsg 5 Resetas 6 Add

PF7 Backward 8 Forward 9 Userview 10 Top 11 Bottom 12 Cancel

### Notes

- 1 Hidden sessions (see page 20) will only be included in the list of application definitions if the user's OLA security class allows it (see 'Security considerations' on page 27).
- 2 The 'Priority' column will only appear if the user's OLA security class allows it (see 'OLA security settings for miscellaneous entries' on page 34).
- 3 If a particular session does not have an APPL name then the APPLID name will be shown, taken from the appropriate USER, PROFILE or SYSTEM definition.
- 4 The 'AUTOSTART' columns only appear if a security setting has been set, to grant the user DISPLAY authority.

- 5 The Session List will only honour one option (S, C, D or R), or 'Priority', or 'AUTOSTART' change at a time; with the option taking precedence, then 'Priority' and then 'AUTOSTART'.
- 6 Modified sessions can not be reset at this level. To reset a modified session select the session and reset the individual parameter modifications.

Depending on the path used to navigate to the USER Definition menu, and your OLA security class, the Session List panel enables you to either:

- view your own session definitions and modify your own session definitions  
or
- view or modify session definitions associated with the selected user.

### See

- 'Possible operations' below
- 'Session types' on page 69

### Highlighting of session definitions

Depending on the characteristics of your display device, either different colors or highlighting are used to distinguish between session definitions set by the user, and session definitions inherited from other configuration definitions.

## Possible operations

If your OLA security class allows it (see 'Security considerations' on page 27), you can perform these operations on the entries in a Session List panel:

- Add a session definition to the list.
- Copy an existing session definition. (see 'Copying a session definition' on page 67)
- Modify a session definition (see 'Modifying a particular session definition' on page 68).

In this case, you will *not* be able to modify sessions defined on PROFILE definitions or the SYSTEM definition.

- View a session definition.
- Change the order of session definitions (see 'Changing the order of session definitions' on page 68).
- Delete a session definition from the list.
- Customize which sessions get auto-started (see 'Customizing which sessions get auto-started' on page 69).

### Copying a session definition

If a session with a SESTYPE explicitly set or inherited is copied, it will be copied with a duplicate SESTYPE and a warning message displayed. As SESTYPES must be unique, the new session will not appear on the Session Manager main menu (although it will appear on the Session List in OLA) until it has been allocated a unique SESTYPE. This can be done manually or by using the supplied sample job ISZSTJOB (see 'Setting priority sessions' on page 23.).

Also, the following actions apply to the SELCMDS and TRANSID values:

- If the session numbers are the same then these values are copied to the new session.
- If the session numbers are not the same then these values are not copied to the new session as these values need to be unique to the session and therefore new values need to be allocated for the newly copied session.

### Modifying a particular session definition

To modify a particular session definition select it and press Enter or use the E (= Edit) or S (= Select) action code (see 'Codes for List actions' on page 52). The attributes associated with that definition will be shown in an OLA list.

Processing of attributes for a particular session definition is similar to processing of common enduser attributes, so refer to 'Display user's common enduser parameters' on page 60 for a description of how to proceed.

### Changing the order of session definitions

On a Session List panel, sessions are displayed in menu sequence number order (see page 21). To establish the sequence number, Session Manager searches these definitions in turn:

- 'Priority' sessions set up through the Session List of OLA
- USER or TERMINAL
- PROFILE
- APPL  
(only when the session definition contains a REFAPPL parameter)

To specify a session as a priority session, to be displayed at the top of the Session List, move the cursor to the 'Priority' column for the session you wish to move, type the number of the position at which you want the session to appear, and press Enter. If you type a number greater than the currently specified number of priority sessions then the changed session will appear as the last entry on the priority list.

**Note** If any value has been entered in the first column of the Session List panel then it is this that will be actioned when Enter is pressed, not the new priority number for the session.

The priority number that you enter will be shown in a different color to indicate that a change is pending, and the sessions will not appear in their new order until the changes have been saved. Only one new session priority can be entered at a time, but any number can be set (along with other settings on the USER Statement) before the pending changes are saved.

To remove a session from the priority list (returning it to its original configured position) enter a space over the number in the 'Priority' column and press Enter.

To undo *all* pending changes to the user-customized session priority, a command – RESETPRI – can be issued from within the Session List menu. This command, which is also associated with a PF Key, will not reset all priority session definitions; only those changes which are currently *pending* will be undone.

### Customizing which sessions get auto-started

The 'AUTOSTART' information shows, in the 'Session' column, each session's Autostart setting (Y) or (N) as defined in the configuration. You can reverse this setting for a session by entering Y or N in the 'USER' column. Entering spaces will cause the Autostart setting to be removed.

When the Save command is issued, any changes to the 'USER' column are written to the configuration dataset, along with any other changes made to the USER definition.

To reset any pending Autostart changes use the Resetas command (PFkey 5). This does not remove any saved user-defined changes: this is done by removing each setting under 'USER' in turn.

### Session types

On a Session List panel, sessions with a duplicate session type (see page 22) are eliminated. To establish the session type, the usual precedence is used – that is, Session Manager searches these definitions in turn:

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

### Display User View of session list

An authorized user can press PF9 on the Sessions List panel (see 'Display list of sessions associated with user' on page 66) to display a view of the session list that shows what would actually be seen by the user whom they are administering if logged on. This is done by interrogating both the ESM (if installed and active) to obtain permissions for that user and other session attributes such as HIDE.

This is known as the 'User view' of the session list and can be used for both My USER displays and Local USER displays.

The Session List – User View panel is shown below:

Session Manager - Admin		Session List - User View		dd/mm/yyyy hh:mm:ss	
LU	luname	Name of entry: IM1		user	
USER IM1 AUTH 1 OLACCLASS NO RENUMDUP 1001					
<u>Key/Number</u>	<u>APPL</u>	<u>Session</u>			
0003		03 Display user characteristics			
0004		04 Display user characteristics			
PF07	MLOG	CONSOLE MANAGEMENT FACILITY			
PF08	COMMAND	CONSOLE MANAGEMENT FACILITY			
PF09		09 Display user characteristics			
0010		10 Display user characteristics			
PF11	udesc	User description panel			
PF12		12 Display user characteristics			
0013		13 Display user characteristics			
0014		14 Display user characteristics			
PF15	hidden	Applid 'HIDDEN'			
PF17		NOT hidden (no APPL)			
>>>>					
Command ==>>					
PF1 Help	2	3 End	4 Actmsg	5	6
PF7 Backward	8 Forward	9	10 Top	11 Bottom	12 Cancel



## Hardcopy Configuration Statements menu

If your OLA security class allows it (see ‘Security considerations’ on page 27), select the option **Hardcopy definitions** from the Main Menu (see page 50) to maintain hardcopy definitions.

The Hardcopy Configuration Statements menu will appear, from which you can select the group (profile, format or route) of hardcopy definitions to be maintained.

```

Session Manager - Admin Hardcopy Configuration Statements   dd/mm/yyyy hh:mm:ss
LU   luname                                                user

Filter or name:
1 _ HCPROFILE definitions
2 _ HCFORMAT definitions
2 _ HCRROUTE definitions

Command ==>
Available options: S - Select (default)
PF1 Help      2      3 Quit      4 Actmsg    5      6
PF7 Backward  8 Forward  9      10      11      12

```

### See

- ‘Display list of hardcopy profiles’ below
- ‘Display list of hardcopy formats’ below
- ‘Display list of hardcopy routes’ on page 72

## Display list of hardcopy profiles

If your OLA security class allows it (see ‘Security considerations’ on page 27), select the option **HCPROFILE definitions** from the Hardcopy Configuration Statements menu (see above) to maintain hardcopy profile (that is, HCPROFILE) definitions.

The Hardcopy Profile menu will appear. Processing of HCPROFILE definitions is similar to processing of PROFILE definitions, so refer to ‘Display list of PROFILE definitions’ on page 51 for a description of how to proceed.

## Display list of hardcopy formats

If your OLA security class allows it (see ‘Security considerations’ on page 27), select the option **HCFORMAT definitions** from the Hardcopy Configuration Statements menu (see above) to maintain hardcopy format (that is, HCFORMAT) definitions.

The Hardcopy Format menu will appear. Processing of HCFORMAT definitions is similar to processing of PROFILE definitions, so refer to ‘Display list of PROFILE definitions’ on page 51 for a description of how to proceed.

## Display list of hardcopy routes

If your OLA security class allows it (see ‘Security considerations’ on page 27), select the option **HCRROUTE definitions** from the Hardcopy Configuration Statements menu (see page 71) to maintain hardcopy route (that is, HCRROUTE) definitions.

The Hardcopy Route menu will appear. Processing of HCRROUTE definitions is similar to processing of PROFILE definitions, so refer to ‘Display list of PROFILE definitions’ on page 51 for a description of how to proceed.



## System Configuration Statements menu

If your OLA security class allows it (see ‘Security considerations’ on page 27), select the option **SYSTEM definitions** from the Main Menu (see page 50) to maintain system definitions.

The System Configuration Statements menu will appear, from which you can select the group of system definitions to be maintained.

```
Session Manager - Admin      System Configuration Statements      dd/mm/yyyy hh:mm:ss
LU      luname                                     user

Filter or name:
1 _ ACB RANGE definitions
2 _ LINK definitions
3 _ COMMAND definitions
4 _ SYSTEM definitions
5 _ MESSAGE definitions

Command ==>
Available options: S - Select (default)
PF1 Help      2          3 Quit      4 Actmsg      5          6
PF7 Backward  8 Forward  9          10         11         12
```

**See**

- ‘Display list of RANGE definitions’ below
- ‘Display list of LINK definitions’ below
- ‘Display list of COMMAND definitions’ on page 74
- ‘Display list of SYSTEM definitions’ on page 74
- ‘Display list of MESSAGE definitions’ on page 74

### Display list of RANGE definitions

If your OLA security class allows it (see ‘Security considerations’ on page 27), select the option **ACB RANGE definitions** from the System Configuration Statements menu (see above) to maintain RANGE definitions, which are used to specify ranges of ACBs.

The ACB Range menu will appear. Processing of RANGE definitions is similar to processing of PROFILE definitions, so refer to ‘Display list of PROFILE definitions’ on page 51 for a description of how to proceed.

### Display list of LINK definitions

If your OLA security class allows it (see ‘Security considerations’ on page 27), select the option **LINK definitions** from the System Configuration Statements menu (see above) to maintain LINK definitions.

The Link Definition menu will appear. Processing of LINK definitions is similar to processing of PROFILE definitions, so refer to ‘Display list of PROFILE definitions’ on page 51 for a description of how to proceed.

## Display list of **COMMAND** definitions

If your OLA security class allows it (see ‘Security considerations’ on page 27), select the option **COMMAND definitions** from the System Configuration Statements menu (see page 73) to maintain command authority (that is, COMMAND) definitions.

The Command Authority menu will appear. Processing of COMMAND definitions is similar to processing of PROFILE definitions, so refer to ‘Display list of PROFILE definitions’ on page 51 for a description of how to proceed.

**Note** Filtering is not available on commands.

## Display list of **SYSTEM** definitions

If your OLA security class allows it (see ‘Security considerations’ on page 27), select the option **SYSTEM definitions** from the System Configuration Statements menu (see page 73) to maintain SYSTEM definitions.

The System Statement Definition menu will appear. Processing of SYSTEM definitions is similar to processing of PROFILE definitions, so refer to ‘Display list of PROFILE definitions’ on page 51 for a description of how to proceed.

**Note** SYSTEM definitions are held in PDSE members that are prefixed with ISZSYS.

The ISZSYSCM member contains SYSTEM parameters common across all Session Manager instances, and the ISZSYSxx member contains only the SYSTEM parameters unique to that Session Manager instance. If generated by the Enabler, the ISZSYSxx member will contain the ACB, LOCALNODE, SESACB, STANDBY, SYSPLEXTYPE, TCP and VERBOSE parameters; however the ISZSYSxx and ISZSYSCM members may subsequently be modified, using either the Online Administration or Batch facility, to reflect the installation’s requirements.

Any updates to the ISZSYSCM 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.

When displaying a ISZSYSxx member the values displayed will be the values in the ISZSYSxx member. If no value exists in the ISZSYSxx member for a given parameter then the inherited values will be displayed from the ISZSYSCM member.

## Display list of **MESSAGE** definitions

If your OLA security class allows it (see ‘Security considerations’ on page 27), select the option **MESSAGE definitions** from the System Configuration Statements menu (see page 73) to maintain MESSAGE definitions.

The Message Definition menu will appear. Processing of MESSAGE definitions is similar to processing of PROFILE definitions, so refer to ‘Display list of PROFILE definitions’ on page 51 for a description of how to proceed.

**Note** All messages are 4 numeric characters in length and are padded with leading zeros. MESSAGE definitions are held in PDSE members that are prefixed with ISZM. The 'ISZM' is removed when messages are displayed within OLA.



## CHAPTER 5

# Batch administration

This chapter describes Batch Administration capability of IBM Session Manager for z/OS. If many changes are required to a large number of configuration definitions, this capability enables you to tailor Session Manager using a batch job.

**See**

- ‘Introduction’ on page 78
- ‘Running Batch Administration’ on page 81
- ‘ISZSRCH – search the configuration’ on page 85
- ‘ISZUPDT – update the configuration’ on page 93
- ‘ISZACTV – activate pending updates’ on page 104
- ‘ISZREMC – invoke Session Manager command’ on page 108
- ‘Examples’ on page 109
- ‘Batch Administration configuration’ on page 117

**CAUTION** We strongly recommend that you backup the PDS(E)s holding your Session Manager configuration data *before* performing a batch update using the ISZUPDT command. For details of the datasets, see the ‘Configuration DDNAMEs’ section in the ‘Converting a Classic to an OLA system’ chapter of the *Installation and Customization* manual.

## Introduction

The Batch Administration process is based on four supplied commands, plus a series of user-defined control files. See:

- ‘Statement syntax’ below
- ‘Applying updates’ below
- ‘Activating the updates’ on page 79
- ‘Updates based on current values’ on page 79
- ‘Combining the commands’ on page 80

## Statement syntax

When using Batch Administration, *specify configuration statements and parameters in full* – that is, do *not* use a shortened form documented in the *Technical Reference*.

## Applying updates

The Batch Administration capability enables a family of configuration files to be updated in batch mode. The requisite changes are defined in a control file which is read by the ISZUPDT command. An update control file looks like this:

```
* Create new user
ADD DDN(USER) MBR(TOM)
password startup
prof accounts

* Create another new user
ADD DDN(USER) MBR(DICK)
password startup
prof accounts

* One more time...
ADD DDN(USER) MBR(HARRY)
password startup
prof accounts
```

This example shows three ADD requests, creating basic definitions for new users TOM, DICK and HARRY. An update control file can also contain REP, MOD and DEL requests, and the various types of request can be mixed in any combination. Comments (the “\*” lines) are ignored.

```
* Update the ACCOUNTS profile
MOD DDN(PROFILE) MBR(ACCOUNTS)
passtry 5
applid ledger3

* Remove the LEDGER1 application
DEL DDN(APPL) MBR(LEDGER1)
```

**Note** The control information is shown here in upper case, with the keywords and data values in lower case, but you can use upper, lower or mixed case as you wish.

Assuming that you have created a sequential file containing ADD|REP|MOD|DEL requests, allocated to the DDNAME of MYMODS, you apply them like this:

```
ISZUPDT INPUT(DDN(MYMODS))
```

Alternatively, if the update control information is in member MYMODS of a PDS allocated to the DDNAME of CONFIG, then the call becomes:

```
ISZUPDT INPUT(DDN(CONFIG MYMODS))
```

For more information, see 'ISZUPDT – update the configuration' on page 93.

## Activating the updates

The ISZUPDT command applies your changes, but the modifications don't become active until the next time that Session Manager is restarted. If this is inconvenient, you can immediately activate the ADD|REP|MOD requests – but not any DEL requests – by creating an activation control file:

```
ACTIVATE DDN(USER) MBR(TOM)
ACTIVATE DDN(USER) MBR(DICK)
ACTIVATE DDN(USER) MBR(HARRY)
ACTIVATE DDN(PROFILE) MBR(ACCOUNTS)
```

This type of control file is read by the ISZACTV command:

```
ISZACTV INPUT(DDN(MYACTIV))
```

The ISZACTV command causes your changes to be activated immediately, without the need to restart the Session Manager instance. In fact, since you often want to do this, you can arrange for ISZUPDT to generate a suitable activation control file automatically:

```
ISZUPDT INPUT(DDN(MYMODS)) ACTIVATE(DDN(MYACTIV))
```

For more information, see 'ISZACTV – activate pending updates' on page 104.

## Updates based on current values

The updates that you wish to apply may be based on existing configuration definitions. Rather than create your update control file manually, you can use the ISZSRCH command to generate one that is either ready-to-run, or which requires only minor customization before use. ISZSRCH searches some range of existing definitions, optionally matching on keywords and values, and writes out data extracted from those definitions in a user-specified format defined by a 'mask'. For example:

```

ISZSRCH
  STATEMENT(USER PROFILE)
  KEYWORD(IS(APPLID(LEDGER1) APPLID(LEDGER2)))
  OUTPUT(DDN(CONFIG MYMODS) MASK(DDN(CONFIG MYMASK)))

```

This matches all **USER** and **PROFILE** definitions containing the settings **APPLID LEDGER1** or **APPLID LEDGER2**. Information from those definitions is written to member **MYMODS** of the PDS currently allocated to the DDNAME of **CONFIG**, in the format defined by member **MYMASK**. If **MYMASK** contained these lines:

```

#SRCH_TITLE* Update USER and PROFILE definitions
MOD DDN(%SRCH_DDN) MBR(%SRCH_MBR)
session 30
applid ledger3

```

then the resulting output **MYMODS** might be as follows – immediately usable by **ISZUPDT** (where its effect would be to update the definitions of **SESSION 30** to refer to **APPLID LEDGER3**):

```

* Update USER and PROFILE definitions
MOD DDN(USER) MBR(TOM)
session 30
applid ledger3
MOD DDN(USER) MBR(DICK)
session 30
applid ledger3
MOD DDN(USER) MBR(HARRY)
session 30
applid ledger3
MOD DDN(PROFILE) MBR(ACCOUNTS)
session 30
applid ledger3

```

For more information, see ‘**ISZSRCH – search the configuration**’ on page 85.

## Combining the commands

The three commands can be used individually, or invoked from a higher-level script in which, for example, **ISZSRCH** selects some definitions, **ISZUPDT** applies updates to the chosen definitions, and finally **ISZACTV** activates the updated definitions.

For more information, see ‘**Specifying update commands**’ on page 82.



## Running Batch Administration

In order to update your Session Manager configurations in batch, two components are required:

- a **batch program**, which applies a predefined set of modifications to the Session Manager configuration files, and
- one or more **update commands** which specify the nature of those modifications.

The following sections introduce these two components.

### Executing the batch program

In order to execute the program which performs Batch Administration, you can use JCL member ISZBAJOB, which is supplied in library .SISZCONF. Before submitting the job, you should edit this member (or a copy thereof):

- near the beginning, to ensure that the DSNPREF environment variable is set correctly, and
- near the end, to specify the commands defining your required modifications (together with any parameters which those commands require):

```
...
//BATCHGO EXEC PROC=ISMBATCH
//STEP01.SEQ DD *
update_command1
update_command2
...
update_commandN
/*
//
```

The format of the *update\_commands* is described in ‘Specifying update commands’ on page 82.

When you run the job, the processing sequence is as follows:

- 1 ISZBAJOB writes the *update\_command* information to a temporary dataset allocated to SEQ.
- 2 ISZBAJOB executes the “BT” instance of Session Manager.
- 3 The configuration information for this Session Manager instance includes an INITIAL\_CMD=BATSCR parameter, so the Batch instance runs that script.
- 4 BATSCR reads and executes the contents of the SEQ temporary dataset, thus performing the updates that you requested on all connected Session Manager instances.

The BATSCR script can be found in configuration member ISZSBAT, which is supplied in library .SISZCONF. These control options can be used with BATSCR:

LIST NOLIST	LIST lists the input; NOLIST (the default) does not.
-------------	--

CONT|NOCONT|CONTMAXCC    CONT sets the return code to zero, and continues on error; NOCONT (the default) does not continue on error; CONTMAXCC sets the return code as the highest value set so far during the batch job, and continues after an error.

If more than one of the CONT, NOCONT or CONTMAXCC options are used in the script, the most recent option encountered takes precedence.

- 5 The Batch instance then terminates (because of its OPTION START=NO parameter).
- 6 ISZBAJOB terminates.

For additional information on the batch program environment, see 'Batch Administration configuration' on page 117.

## Specifying update commands

The *update\_commands* define your required modifications, and consist of a sequence of one or more lines. A command must start in column 1; any parameters may follow on that same line or on the line below. If a command and its parameters extends over more than one line, it must be indented by at least one space on the continuation lines. Blank lines are ignored, as are comments (with "\*" as the first character in the record). For example:

```
...
//BATCHGO EXEC PROC=ISMBATCH
//STEP01.SEQ DD *
* Run the search command
* If ISZSRCH returns non-zero value, set the job return code
* and exit
* Note: NOCONT is the default behavior, but is specified
* here for clarity
NOCONT
ISZSRCH
STATEMENT(USER)
KEYWORD(IS(USER(HARRY)))
OUTPUT(DDN(ISZUPDT EXCTRL1) REP MASK(DDN(MASK ISZMASK1)))

* Run the update command - NOCONT behavior is still in effect
ISZUPDT
INPUT(DDN(ISZUPDT EXCTRL1))
ACTIVATE(DDN(ISZACTV EXACTV1) REP)

* Finally, activate the changes, but now we ignore failures
CONT
ISZACTV
INPUT(DDN(ISZACTV EXACTV1))
/*
//
```

The commands determine what updates are to be performed. They can be any combination of control commands, standard Session Manager commands, or user-written scripts.

**Control commands**

The following control commands are available:

LIST|NOLIST

LIST writes the *update\_commands* to the audit file; NOLIST (the default) does not.

CONT|NOCONT|CONTMAXCC

If an *update\_command* sets *t\_rc* to a non-zero value then processing terminates unless you've specified CONT, which sets the return code to zero and continues to execute the *update\_commands* after an error has occurred. NOCONT (the default) terminates on error. CONTMAXCC sets the return code as the highest value set so far during the batch job, and continues after an error.

TRACE

produces debugging information.

**Standard commands**

You can run ISZSRCH, ISZUPDT and ISZACTV directly. For example:

```
ISZUPDT PARMS(DSN(MYFILE))
```

causes Session Manager to update its configuration using the ADD|REP|MOD|DEL requests held in the sequential file currently allocated to MYFILE. Another example:

```
ISZACTV ALL
```

causes Session Manager to reload the configurations of all connected instances.

**User-written scripts**

If you wish to combine the use of ISZSRCH, ISZUPDT and ISZACTV in some more sophisticated manner, you can invoke them from a script of your own. For example, your script could be modelled on:

```
let t_scparms = 'STATEMENT(...)' -
               'KEYWORD(...)'   -
               'OUTPUT(...)'

call ISZSRCH
if t_rc ne 0
    auditmsg 'ISZSRCH failed rc=&t_rc&'
    return
end

let t_scparms = 'INPUT(...)'      -
               'ACTIVATE(...)'

call ISZUPDT
if t_rc ne 0
```

```
        auditmsg 'ISZUPDT failed rc=&t_rc&'
        return
    end

    let t_scparms = 'INPUT(...)'
    call ISZACTV
    if t_rc ne 0
        auditmsg 'ISZACTV failed rc=&t_rc&'
        return
    end
```

For an example, see ‘Copying a user definition using a script’ on page 115.

## ISZSRCH – search the configuration

ISZSRCH searches the Session Manager configuration files for definitions which match specified search criteria. When a match is found, one or more output records are written, in user-specified format, containing information derived from the matched definition.

### Format

```
ISZSRCH
  STATEMENT(ddname_list)

  SESSION(session_list)
  SESSION(ALL)

  KEYWORD(keyword_list)
  NOTKEYWORD(keyword_list)

  OUTPUT(DDN(ddname [mbrname]) [disp] [MAX(n)] [formatspec])
  OUTPUT(VAR(varname) [MAX(n)] [formatspec])
  OUTPUT(AUDIT [MAX(n)] [formatspec])

  LISTMEMBERS
  TRACE

  PARMS(DDN(ddname [mbrname]))
  PARMS(VAR(varname))
```

ISZSRCH may be called from a TPSL script, in which case the parameters are passed by assigning their values to the `t_scparms` variable. All input datasets should have edit numbering disabled.

**Note** The total length of the inline parameters supplied to ISZSRCH must not exceed 256 characters. You can use the PARMS parameter to bypass this restriction.

### Parameters

All of the parameters are optional, and may appear in any order (but see 'ISZSRCH processing sequence' on page 91).

STATEMENT(*ddname\_list*)

The optional STATEMENT parameter reduces the search scope. If omitted, the default is to search the entire configuration.

*ddname\_list* contains one or more *ddnames* currently allocated to Session Manager configuration datasets – see the 'Configuration DDNAMEs' section in the 'Converting a Classic to an OLA system' chapter and the 'Customer, samples and system configuration datasets' section in the 'Performing a basic configuration' chapter of the *Installation and Customization* manual. The search is restricted to those definitions identified by the listed *ddnames*.

SESSION(*session\_list*)  
SESSION(ALL)

The optional SESSION parameter reduces the search scope, and is applicable only when searching USER, PROFILE and TERMINAL definitions. If omitted, the default is to search all session definitions. The parameter may be repeated, and its position relative to any KEYWORD and NOKEYWORD parameters is significant – see ‘ISZSRCH processing sequence’ on page 91 for more information.

*session\_list* contains one or more *session\_number* or *session\_pfkey* values referring to Session Manager sessions. The search is restricted to those definitions referring to the listed sessions.

ALL broadens the scope to include all defined sessions.

KEYWORD(*keyword\_list*)  
NOTKEYWORD(*keyword\_list*)

The optional KEYWORD and NOTKEYWORD parameters reduce the search scope. If omitted, the default is to search all definitions. Any combination of these parameters may be supplied, and their position relative to any SESSION parameters is significant – see ‘ISZSRCH processing sequence’ on page 91 for more information.

KEYWORD(*keyword\_list*) matches only definitions containing any of the listed keywords. For example:

```
KEYWORD(IS(SPYABLE SPYGROUP))
```

matches definitions containing either of the keywords.

```
KEYWORD(IS(SPYABLE))  
KEYWORD(IS(SPYGROUP))
```

matches definitions containing both of the keywords.

To match specific keyword values, add (*value*) after any *keyword*. For example:

```
KEYWORD(IS(SPYABLE(TELL) SPYABLE(YES) SPYABLE(ON)  
SPYGROUP(MYGROUP)))
```

matches definitions containing either SPYABLE (set to TELL, YES or ON) or SPYGROUP (set to MYGROUP). Definitions containing SPYABLE and SPYGROUP keywords set to other values are not selected.

```
KEYWORD(IS(SPYABLE(TELL) SPYABLE(YES) SPYABLE(ON)))  
KEYWORD(IS(SPYGROUP(MYGROUP)))
```

matches definitions containing both of the keywords with the specified values.

To exclude specific keyword values, use NOT in place of IS. For example:

```
KEYWORD(NOT(SPYABLE(NO) SPYABLE(OFF) SPYGROUP(HISGROUP)))
```

matches definitions containing either SPYABLE (set to anything other than NO or OFF) or SPYGROUP (set to anything other than HISGROUP).

```
KEYWORD(NOT(SPYABLE(NO) SPYABLE(OFF)))
KEYWORD(NOT(SPYGROUP(HISGROUP)))
```

matches definitions containing both of the keywords with none of the specified values.

**NOTKEYWORD**(*keyword\_list*) matches only definitions **not** containing the listed *keywords*. For example:

```
NOTKEYWORD(SPYABLE SPYGROUP)
```

excludes definitions containing at least one of the keywords.

```
NOTKEYWORD(SPYABLE)
NOTKEYWORD(SPYGROUP)
```

excludes definitions containing both of the keywords.

*keyword\_lists* and *values* may contain the wildcards '\*' (any sequence of characters) and '+' (any single character). Keywords must be given in full – abbreviations are not permitted.

You can also use **KEYWORD** to reduce the search scope by selecting **statement types**. The form:

```
KEYWORD(IS(USER))
```

selects all user definitions, and thus is identical in effect to **STATEMENT**(USER). More usefully,

```
KEYWORD(IS(USER(A*)) IS(USER(B*)))
```

selects only the definitions of users whose name begins with A or B.

```
OUTPUT(DDN(ddname [mbrname]) [disp] [MAX(n)] [formatspec])
OUTPUT(VAR(varname) [MAX(n)] [formatspec])
OUTPUT(AUDIT [MAX(n)] [formatspec])
```

The optional **OUTPUT** parameter defines a location where information derived from the matched definitions is to be written, and the format of that information. Any combination of these parameters may be supplied. See 'ISZSRCH output format' on page 88 for information on the *formatspec*.

**OUTPUT**(DDN(*ddname* . . . )) specifies that information should be written to a member of the PDS(E) which is currently allocated to *ddname* (the default for *mbrname* is ISZSRCHO). The *disposition* can be REP (the default – replace any existing file contents), MOD (append to any existing file contents), or NEW (fail if there are existing contents).

**Note** The only Session Manager PDS(E) to which *ddname* can safely refer is CONFIG – do not attempt to write to any of the other configuration PDS(E)s.

**OUTPUT**(VAR(*varname*) . . . ) specifies that information should be written to the specified subscripted variable. The first output record is in *varname*.1, the second in *varname*.2, and so on; the total number of output records is in *varname*.0. A subscripted variable can contain up to 9999 records.

**OUTPUT**(AUDIT . . . ) specifies that information should be written to the audit file. This is the default if no **OUTPUT** parameter is specified.

In all cases, the optional `MAX(n)` argument specifies a limit on the number of records written to this output destination. If the limit is exceeded, nothing further is written to the destination; processing continues until either all statements have been searched or all output destinations have reached their limits.

#### LISTMEMBERS

The optional LISTMEMBERS parameter causes the output report to mention each *mbrname*, in addition to the default of listing each *ddname* that was searched.

#### TRACE

The optional TRACE parameter produces debugging information.

PARMS(DDN(*ddname* [*mbrname*]))

PARMS(VAR(*varname*))

The optional PARMS parameter defines an additional source of parameter information. Any combination of these parameters may be supplied.

PARMS(DDN(*ddname* [*mbrname*])) specifies that ISZSRCH parameters should be read from the file which is currently allocated to *ddname*. This can be either a sequential file, or a member of a PDS(E).

PARMS(VAR(*varname*)) specifies that ISZSRCH parameters should be read from the specified subscripted variable. The first input record is in *varname.1*, the second in *varname.2*, and so on; the total number of input records is in *varname.0*. A subscripted variable can contain up to 9999 records.

## ISZSRCH output format

The output from the ISZSRCH command is derived from the configuration definitions which match the specified search criteria. (If no STATEMENT, SESSION, KEYWORD or NOTKEYWORD parameters are supplied, then all definitions match, and ISZSRCH outputs the entire configuration.)

### Format specification parameters

For each defined output stream – any combination of *ddnames* and *varnames*, plus the audit file – you can specify the format of the output by means of the *formatspec* arguments of the OUTPUT parameter. An *formatspec* can be any of:

MASK(DDN(*ddname* [*mbrname*]))

MASK(VAR(*varname*))

MASK(STR("mask"))

MASK(DDN(*ddname* [*mbrname*])) specifies that the mask definition should be read from the file which is currently allocated to *ddname*. This can be either a sequential file, or a member of a PDS(E). The definition can occupy one or more lines.

MASK(VAR(*varname*)) specifies that the mask definition should be read from the specified subscripted variable. The first input record is in *varname.1*, the second in *varname.2*, and so on; the total number of input records is in *varname.0*. The definition can occupy up to 9999 lines.



`MASK(STR("mask"))` specifies the mask definition as the literal *mask*. The definition occupies a single line only.

#### WRAP | TRUNC

WRAP (the default) specifies that long output records are wrapped onto the next line if they exceed the defined width. The continuation line normally begins with a blank character; however, if the first character of the output record is an asterisk then the continuation line begins with an asterisk followed by a blank character.

TRUNC specifies that long output records should be truncated at the defined width.

#### WIDTH(*n*)

defines the maximum width of the output records, at which point a long record is either wrapped or truncated. The default is 80.

### Generated control values

A mask specifies a prototype output record: some combination of literal text and control values. The literal text is output as-is for each selected configuration definition, and the control values are replaced by material either derived from that definition, or generated by the ISZSRCH command itself. Control values must be specified in upper case.

The control values providing command-generated data are:

#### #SRCH\_TITLE

One or more lines beginning with #SRCH\_TITLE may be placed at the start of the mask definition. The material following #SRCH\_TITLE on those lines is output just once, at the start of the ISZSRCH command session. This once-only output can itself consist of literal text and the following control values:

#### %SRCH\_DATE

is replaced by the current date.

#### %SRCH\_TIME

is replaced by the current time.

#### %SRCH\_PARMS

is replaced by the STATEMENT, SESSION, KEYWORD and NOTKEYWORD parameters which define the current selection criteria. Each parameter is output on a separate line.

### Derived control values

The control values providing data derived from configuration definitions are:

#### %SRCH\_DDN

is replaced by the *ddname* allocated to the Session Manager configuration dataset containing the selected definition.

#### %SRCH\_MBR

is replaced by the *mbrname* in that dataset.

`%SRCH_STMT_TYPE`

is replaced by the selected definition's statement type (usually, this is the same as `%SRCH_DDN`).

`%SRCH_STMT_NAME`

is replaced by the selected definition's statement name (usually, this is the same as `%SRCH_MBR`).

`%SRCH_MATCH_SESS`

is replaced by the matched *session\_number* or *session\_pfkey* value, or by blank if the definition was not found in the session of a USER or PROFILE. If multiple matches are found, the mask line containing this control value is repeated.

`%SRCH_MATCH_KEYW`

is replaced by the matched keyword name, or by blank if no keyword match was made. If multiple matches are found, the mask line containing this control value is repeated.

`%SRCH_MATCH_KEYW_VALUE`

is replaced by the matched keyword value, or by blank if no value match was made. If multiple matches are found, the mask line containing this control value is repeated.

`%SRCH_KEYW.keyword`

`%SRCH_KEYW.keyword.`

`%SRCH_KEYW.keyword` is replaced by the value of *keyword* in the selected definition, or by blank if the definition does not contain *keyword*. Unless it occurs at the end of a line in the mask definition, the control value must be followed either by a space (which is included in the output record) or by a period (which is omitted from the output record).

`%SRCH_ALL_KEYW`

`%SRCH_ALL_KEYW_VALUE`

`%SRCH_ALL_KEYW` is replaced by a list of all keywords in the selected definition, one to a line. `%SRCH_ALL_KEYW_VALUE` is replaced by a corresponding list of the values associated with those keywords. Normally you would specify both `%SRCH_ALL_KEYW` and `%SRCH_ALL_KEYW_VALUE` on the same line in the mask, in which case the output is a sequence of lines, each presenting a keyword and its associated value or values.

## Typical output masks

For examples of typical output masks, see 'Examples' on page 109.

## Default output mask

If a mask is not specified for a given output stream, the following default mask is used – see member ISZDFMSK in the MASK dataset (which is supplied in library .SISZBMSK):

```
#SRCH_TITLE* ISZSRCH performed on %SRCH_DATE %SRCH_TIME
#SRCH_TITLE* specified parameters were
#SRCH_TITLE* %SRCH_PARMS
#SRCH_TITLE*
DDN=%SRCH_DD,MBR=%SRCH_MBR,SESS=%SRCH_MATCH_SESS
      %SRCH_STMT_TYPE %SRCH_STMT_NAME
      %SRCH_ALL_KEYW %SRCH_ALL_KEYW_VALUE
```

Output using this default mask might appear as follows:

```
* ISZSRCH performed on 08/03/04 17:26:20
* specified parameters were
* STATEMENT(USER)
*
DDN=USER,MBR=ATEST,SESS=
USER atest
TRACE MISER
AUTH 8
IDLELOGOFF 77 AFTERDISC
WARN n
KEY pf9
APPLID s05swit
HIDE y

DDN=USER,MBR=COM0,SESS=
USER com0
AUTH 9
IDLELOGOFF 66 AFTERDISC
WARN y

...

Search - 23 definition(s) read
Search - 23 definition(s) selected
Search - 178 lines for AUDIT
Search complete (rc=0)
```

## ISZSRCH processing sequence

ISZSRCH searches the Session Manager configuration for definitions which match the selection criteria defined by any STATEMENT, SESSION, KEYWORD or NOTKEYWORD parameters; the default is to match all definitions. For most types of definition, only the KEYWORD|NOTKEYWORD parameters determine whether the definition is selected. However, for USER, PROFILE and TERMINAL definitions, the SESSION parameter also affects the selection. This is best explained by an example:

```

ISZSRCH
  STATEMENT(USER PROFILE TERMINAL)

  KEYWORD(keyword_list)           Note 1
  NOTKEYWORD(keyword_list)

  SESSION(ALL)
    KEYWORD(keyword_list)         Note 2
    NOTKEYWORD(keyword_list)

  SESSION(1 2 3 4 5)
    KEYWORD(keyword_list)         Note 3
    NOTKEYWORD(keyword_list)

  SESSION(PF1 PF2 PF3 PF4 PF5)
    KEYWORD(keyword_list)         Note 3
    NOTKEYWORD(keyword_list)

  ...

```

### Notes

- 1** KEYWORD|NOTKEYWORD parameters before the first SESSION parameter look for matches only in the Enduser settings and the Session Default settings.
- 2** KEYWORD|NOTKEYWORD parameters after a SESSION(ALL) parameter look for matches in any Session settings but **not** in Enduser settings and the Session Default settings.
- 3** KEYWORD|NOTKEYWORD parameters after a SESSION(*session\_list*) parameter look for matches only in the listed Session settings.

## ISZSRCH return values

ISZSRCH returns the following values:

Return code	Meaning
0	Search was successful.
4	Search abandoned – all OUTPUTs reached their maximum.
12	Error in parameter parsing, or I/O error; see audit messages.

## ISZUPDT – update the configuration

ISZUPDT updates the Session Manager configuration files. It reads a control file containing ADD, REP, MOD and DEL instructions, typically (but not necessarily) generated by ISZSRCH.

**CAUTION** We strongly recommend that you backup the PDS(E)s holding your Session Manager configuration data *before* performing a batch update using the ISZUPDT command. For details of the datasets, see the ‘Configuration DDNAMEs’ section in the ‘Converting a Classic to an OLA system’ chapter of the *Installation and Customization* manual.

The updates normally take effect only when the Session Manager instances are next restarted. Optionally, ISZUPDT can generate a control file suitable for use by ISZACTV, to activate ADD, REP and MOD updates **without** restarting the instances.

**Note** Customer configuration definitions are split from system configuration definitions, and any definitions added or modified by Batch Administration will be held in the *customer* datasets. Further, if a system definition is modified then it will be stored in the corresponding customer dataset; unlike customer definitions, system and sample definitions can *not* be deleted. For more information, see the ‘Customer, samples and system configuration datasets’ section in the ‘Performing a basic configuration’ chapter of the *Installation and Customization* manual.

### Format

```
ISZUPDT
  INPUT(DDN)
  INPUT(DDN(ddname [mbrname]))
  INPUT(VAR(varname))

  ACTIVATE(DDN(ddname [mbrname]) [disp])
  ACTIVATE(VAR(varname))

  OUTPUT(DDN(ddname [mbrname]) [disp])
  OUTPUT(VAR(varname))
  OUTPUT(AUDIT)

  CONTINUE
  LISTINPUT
  TRACE

  PARMS(DDN(ddname [mbrname]))
  PARMS(VAR(varname))
```

ISZUPDT may be called from a TPSL script, in which case the parameters are passed by assigning their values to the t\_scparms variable. All input datasets should have edit numbering disabled.

**Note** The total length of the inline parameters supplied to ISZUPDT must not exceed 256 characters. You can use the PARMS parameter to bypass this restriction.

## Parameters

Except for INPUT, all of the parameters are optional, and may appear in any order.

```
INPUT(DDN)
INPUT(DDN(ddname [mbrname]))
INPUT(VAR(varname))
```

The INPUT parameter defines the source of control information. Any combination of these parameters may be supplied. See 'ISZUPDT input format' on page 96 for details of how the information is structured.

INPUT(DDN) specifies that ISZUPDT control information should be read from the ISZUPDTI member of the PDS(E) which is currently allocated to CONFIG.

INPUT(DDN(ddname [mbrname])) specifies that ISZUPDT control information should be read from the file which is currently allocated to ddname. This can be either a sequential file, or a member of a PDS(E).

INPUT(VAR(varname)) specifies that ISZUPDT control information should be read from the specified subscripted variable. The first input record is in varname.1, the second in varname.2, and so on; the total number of input records is in varname.0. A subscripted variable can contain up to 9999 records.

```
ACTIVATE(DDN(ddname [mbrname]) [disp])
ACTIVATE(VAR(varname))
```

The optional ACTIVATE parameter defines a location where ISZACTV control information is to be written. Any combination of these parameters may be supplied. See 'ISZACTV input format' on page 106 for information on how the information is presented.

ACTIVATE(DDN(ddname [mbrname]) [disp]) specifies that information should be written to a member of the PDS(E) which is currently allocated to ddname (the default for mbrname is ISZACTVI). The disposition can be REP (the default – replace any existing file contents), MOD (append to any existing file contents), or NEW (fail if there are existing contents).

**Note** The only Session Manager PDS(E) to which ddname can safely refer is CONFIG – do not attempt to write to any of the other configuration PDS(E)s.

ACTIVATE(VAR(varname)) specifies that information should be written to the specified subscripted variable. The first output record is in varname.1, the second in varname.2, and so on; the total number of output records is in varname.0. A subscripted variable can contain up to 9999 records.

```
OUTPUT(DDN(ddname [mbrname]) [disp])
OUTPUT(VAR(varname))
OUTPUT(AUDIT)
```

The optional OUTPUT parameter defines a location where the update report is to be written. Any combination of these parameters may be supplied. See 'ISZUPDT output format' on page 101 for information on how the information is presented.

OUTPUT(DDN(*ddname* [*mbrname*])) [*disp*]) specifies that information should be written to a member of the PDS(E) which is currently allocated to *ddname* (the default for *mbrname* is ISZUPDTO). The *disposition* can be REP (the default – replace any existing file contents), MOD (append to any existing file contents), or NEW (fail if there are existing contents).

**Note** The only Session Manager PDS(E) to which *ddname* can safely refer is CONFIG – do not attempt to write to any of the other configuration PDS(E)s.

OUTPUT(VAR(*varname*)) specifies that information should be written to the specified subscripted variable. The first output record is in *varname*.1, the second in *varname*.2, and so on; the total number of output records is in *varname*.0. A subscripted variable can contain up to 9999 records.

OUTPUT(AUDIT) specifies that information should be written to the audit file. This is the default if no OUTPUT parameter is specified.

#### CONTINUE

The optional CONTINUE parameter causes processing to continue if an error is detected; the default is to terminate.

**Note** Using the CONTINUE parameter will set the return code to eight, regardless of any errors. CONT and CONTMAXCC can also be specified as a control option on the BATSCR script, to alter the final return code setting.

#### LISTINPUT

The optional LISTINPUT parameter causes the output report to include a copy of the input control information.

#### TRACE

The optional TRACE parameter produces debugging information.

PARMS(DDN(*ddname* [*mbrname*]))

PARMS(VAR(*varname*))

The optional PARMS parameter defines an additional source of parameter information. Any combination of these parameters may be supplied.

PARMS(DDN(*ddname* [*mbrname*])) specifies that ISZUPDT parameters should be read from the file which is currently allocated to *ddname*. This can be either a sequential file, or a member of a PDS(E).

PARMS(VAR(*varname*)) specifies that ISZUPDT parameters should be read from the specified subscripted variable. The first input record is in *varname*.1, the second in *varname*.2, and so on; the total number of input records is in *varname*.0. A subscripted variable can contain up to 9999 records.

## ISZUPDT input format

ISZUPDT expects control information to consist of any combination of ADD|REP|MOD|DEL requests. Additionally, it ignores blank lines, and comments (with “\*” as the first character in the record). Each request comprises a single header line, followed (except for DEL) by any number of lines of *keyword* and *value* pairs. Each *keyword* must start in column 1, and must be given in full – abbreviations are not permitted. If a *value* extends over more than one line, it must be indented by at least one space on the continuation lines.

**Note** The definitions of SCRIPT statement types comprise free-format text rather than *keyword* and *value* pairs. This means that you cannot issue MOD requests against such definitions; to change a script you should run ISZSRCH to export the current definition to a PDS(E) member, make your changes using the ISPF editor (or equivalent), and finally run ISZUPDT with a REP request to import the complete changed definition. See also the notes about SCRIPT in ‘ISZUPDT statement types’ on page 98.

### ADD a definition

An ADD request creates a new definition, or reports an error if the definition already exists. Each ADD request has this format:

```
ADD DDN(ddname) MBR(mbrname) [TYPE(stype)] [NAME(sname)]
keyword1 value1
keyword2 value2
...
keywordN valueN
```

*ddname*

is currently allocated to a Session Manager configuration dataset – see ‘Configuration DDNAMEs’ section in the ‘Converting a Classic to an OLA system’ chapter and the ‘Customer, samples and system configuration datasets’ section in the ‘Performing a basic configuration’ chapter of the *Installation and Customization* manual.

*mbrname*

is the name of a new definition to be added to that dataset.

*stype* and *sname*

The statement type *stype* is normally given by the value of *ddname*, and the statement name *sname* is normally given by the value of *mbrname*. There are certain exceptional circumstances where this is not true, requiring *stype* and *sname* to be explicitly specified – see ‘ISZUPDT statement types’ on page 98 for more information.

*keyword value*

The *keyword* and *value* pairs are the configuration settings for the new definition, and may appear in any order (but see ‘ISZUPDT processing sequence’ on page 102).



## REPLACE a definition

A REP request creates a new definition, or replaces a definition that already exists. Each REP request has this format; the parameter values are the same as for an ADD request:

```
REP DDN(ddname) MBR(mbrname) [TYPE(stype)] [NAME(sname)]
keyword1 value1
keyword2 value2
...
keywordN valueN
```

## MODIFY a definition

A MOD request updates an existing definition, or reports an error if the definition does not exist. Each MOD request has this format:

```
MOD DDN(ddname) MBR(mbrname) [TYPE(stype)] [NAME(sname)]
keyword1 value1
keyword2 value2
...
keywordN valueN
```

*ddname*

is currently allocated to a Session Manager configuration dataset – see ‘Configuration DDNAMEs’ section in the ‘Converting a Classic to an OLA system’ chapter and the ‘Customer, samples and system configuration datasets’ section in the ‘Performing a basic configuration’ chapter of the *Installation and Customization* manual.

*mbrname*

is the name of an existing definition in that dataset.

*stype* and *sname*

The statement type *stype* is normally given by the value of *ddname*, and the statement name *sname* is normally given by the value of *mbrname*. There are certain exceptional circumstances where this is not true, requiring *stype* and *sname* to be explicitly specified – see ‘ISZUPDT statement types’ on page 98 for more information.

*keyword value*

The *keyword* and *value* pairs are the changed configuration settings for the definition, and may appear in any order (but see ‘ISZUPDT processing sequence’ on page 102). Parameters not listed here retain their current settings.

There are a few special rules when modifying and adding sub-parameters.

If the sub-parameter already exists within the definition (a modify) then you only need to specify the sub-parameter.

For example:

```
MOD DDN(ddname) MBR(mbrname) [TYPE(stype)] [NAME(sname)]
NFILLC YES
```

If the sub-parameter already exists within the definition (a modify) and the sub-parameter is valid for more than one parameter then you need to specify the specific parameter with its existing setting plus the sub-parameter.

For example:

```
MOD DDN(ddname) MBR(mbrname) [TYPE(stype)] [NAME(sname)]
IDLELOGOFF 10
DIRECTION INOUT
```

If the sub-parameter does not already exist within the definition (an add) then you need to specify the specific parameter with its existing setting plus the sub-parameter.

For example:

```
MOD DDN(ddname) MBR(mbrname) [TYPE(stype)] [NAME(sname)]
CUT *C
NFILLC YES
```

The list of *keyword* and *value* pairs can also include these special cases:

```
#DELETE keyword [value]
```

This definition's setting for *keyword* is deleted, thus reinstating the *keyword*'s default value. Alternatively, the version of *keyword* with the specified *value* is deleted.

```
#REPLACE keyword old_value new_value
```

This definition's version of *keyword* with the specified *old\_value* is changed to *new\_value*.

## DELETE a definition

A DEL request removes an existing definition, or reports an error if the definition does not exist. Each DEL request has this format:

```
DEL DDN(ddname) MBR(mbrname)
```

*ddname*

is currently allocated to a Session Manager configuration dataset – see 'Configuration DDNAMEs' section in the 'Converting a Classic to an OLA system' chapter and the 'Customer, samples and system configuration datasets' section in the 'Performing a basic configuration' chapter of the *Installation and Customization* manual.

*mbrname*

is the name of an existing definition in that dataset.

## ISZUPDT statement types

For most update requests:

- The **statement type** is the same as the *ddname* allocated to the configuration file which contains all of the definitions of that type. This one-to-one correspondence is true for these statement types:

APPL  
 COMMAND – for ‘command name to PDSE member name’ conversion  
                   table, see ‘Parameters for “File I/O”’ on page 35  
 GROUP  
 HCFORMAT  
 HCRROUTE  
 LINK  
 MESSAGE – all PDSE members are prefixed with ISZM and are 4 numeric  
                   characters padded with leading zeros (for example, ISZM0012)  
 PROFILE  
 RANGE  
 TERMINAL  
 USER

See also ‘Updating RUSER statements’ on page 100.

- The **statement name** is the same as the *mbrname* which contains the definition for that item.

For example, you could update the definition of USER TOM with a request of this format:

```
MOD DDN(USER) MBR(TOM)
keyword_and_value_pairs
```

However, there are some special cases.

- 1 The HCPROFILE and TRANSTABLE statement types are referenced by these shortened *ddname* values:

HCPROF  
 TRANSTAB

For example, you could update the definition of HCPROFILE TOMPROF1 with a request of this format:

```
MOD DDN(HCPROF) MBR(TOMPROF1)
keyword_and_value_pairs
```

- 2 The SYSTEM statement type does not have a matching statement name; to update such a definition, you must supply a blank NAME parameter. For example, you could update a particular SYSTEM definition with a request of this format:

```
MOD DDN(SYSTEM) MBR(ISZSYSxx) NAME( )
keyword_and_value_pairs
```

However, we recommend that you use OLA to update SYSTEM definitions.

The ISZSYSCM member contains SYSTEM parameters common across all Session Manager instances and the ISZSYSxx member just contains the SYSTEM parameters unique to that Session Manager instance.

If generated by the Enabler, the ISZSYSxx member will contain the ACB, LOCALNODE, SESACB, STANDBY, SYSPLEXTYPE, TCP and VERBOSE parameters; however the ISZSYSxx and ISZSYSCM members may subsequently be modified, using either the Online Administration or Batch facility, to reflect the installation’s requirements.

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

- 3 Several specialized statement types are stored in the ISZCONxx members rather than having a configuration file of their own. This is true for these statement types:

```
APPLYSU
AUDITROUTE
COPY
INSTALLSU
OPTION
REMOVESU
TRACEROUTE
```

To update these definitions, you must edit the members directly.

- 4 The OLACCLASS security class parameter to various statement types is defined by a script whose name differs from the name of the member holding it. For example, you could update a particular CLASS definition with a request of this format:

**Note** All CLASS PDSE members are prefixed with ISZCLS.

```
MOD DDN(CLASS) MBR(ISZCLSxx) TYPE(SCRIPT) NAME(OLASC_xx)
keyword_and_value_pairs
```

However, we strongly recommend that you use OLA to update CLASS definitions.

- 5 You can rename an existing definition by using this format:

```
MOD DDN(ddname) MBR(mbrname) NAME(new_mbrname)
```

More dangerously, you can change the type of an existing definition by using this format:

```
MOD DDN(ddname) MBR(mbrname) TYPE(new_ddname)
```

- 6 Definitions of the customer-defined SCRIPT statement type are held in one of two places, depending on whether they refer to System (authorized) scripts or Session (unauthorized, user) scripts; ASCRIPT or USCRIPT. To update the definition of a System script, use a request of this format:

```
REP DDN(ASCRIPT) MBR(mbrname) TYPE(SCRIPT)
script_definition_statements
```

To update the definition of a Session script, use a request of this format:

```
REP DDN(USCRIPT) MBR(mbrname) TYPE(SCRIPT)
script_definition_statements
```

## Updating RUSER Single Session Manager instance statements

In an environment where there is just a single Session Manager instance, whose LOCALNODE name is ONLINE1, the OLA Enabler will have created an RUSER statement for the Batch user ISZSMGR. In this case, there is no requirement to add additional RUSER statements as there is only one instance of Session Manager with one RUSER defined for the Batch user.

## Multiple Session Manager instances

However, in a non-Sysplex environment where multiple Session Manager instances share a common configuration, there is a requirement for multiple RUSER statements.

For example, suppose that there are two Session Manager instances, whose LOCALNODE names are ONLINE1 and ONLINE2. In this case, a Session Manager administrator named VIVIENNE might sign on to instance ONLINE1 and need to access instance ONLINE2 using the SEND command. Later, the same administrator might sign on to instance ONLINE2 and need to access instance ONLINE1 using the SEND command. To support this scenario, two RUSER statements are required:

```
RUSER VIVIENNE NODE ONLINE1 ...
RUSER VIVIENNE NODE ONLINE2 ...
```

Both RUSER statements have the same name, VIVIENNE. However, the two configuration members that each contain one of these statements must have different names – therefore, the member name of (at least) one of the RUSER statements has to be different from its definition name.

For example, suppose both of these RUSER statements needed to be replaced, the MBR parameter and the NAME parameter would need to be specified:

```
MOD DDN(RUSER) MBR(VIVON1) NAME(VIVIENNE)
NODE ONLINE1
...
MOD DDN(RUSER) MBR(VIVON2) NAME(VIVIENNE)
NODE ONLINE2
...
```

## ISZUPDT output format

A typical Update Report is as follows:

```
ISZUPDT - Update report

Update date:09/03/04 time:17:21:02

parameters
  LISTINPUT
  INPUT(VAR(LC_INP))
  ACTIVATE(VAR(LC_ACTOUT))
  ACTIVATE(DDN(OLD ACTOUT))
  OUTPUT(AUDIT)
  OUTPUT(VAR(LC_OUT))
  OUTPUT(DDN(OLD))

ISZ6018 Update - processing variable LC_INP
*      test replace
rep ddn(system) mbr(ISZSYShh) name()
tcp yes
stn3270 2399
ibm
siteddn sitexx
iucvname icvbbb
```

```

auth 2
ISZ6087 Update - member ISZSYSHH in SYSTEM replaced

rep ddn(user) mbr(JC20)
paste zzz
nfillp n
spillw y
underiszmgr
curesc y
applid fred
prof prodprof
key    pf1
description 'this ia a long description on 3 lines - line 1'-
'line 2'-
'line 3'
applid s05swit
closedisc
transid jh jk
key    pf2
description 'this starts off small'
applid s09tso
key    pf3
applid s09ims
session 444
applid ola
ISZ6087 Update - member JC20 in USER replaced

...

mod ddn(user) mbr(jc3xx)
applid s09xxx
description 'zzis ia a long description on 3 lines - line 1'-
'zzne 2'-
'zzne 3'
ISZ6087 Update - member JC3XX in USER modified

1 ADD request(s) processed
4 REPLACE request(s) processed
4 MODIFY request(s) processed
1 DELETE request(s) processed
0 error(s)

ISZUPDTE complete rc(0)

```

## ISZUPDT processing sequence

ISZUPDT applies updates to one or more definitions in the configuration, according to an input stream of control information containing ADD|REP|MOD|DEL requests. For most types of definition, the requests contain keywords which uniquely determine how the update is applied. However, for USER, PROFILE and TERMINAL definitions, the SESSION keyword also affects the processing. This is best explained by an example:

```
MOD DDN(USER) MBR(SOMEUSER)
```

*keyword\_and\_value\_pairs* Notes 1,5

```
SESSION ALL
```

*keyword\_and\_value\_pairs* Note 2

```
SESSION ANY
```

*keyword\_and\_value\_pairs* Note 3

```
SESSION 1
```

*keyword\_and\_value\_pairs* Note 4

```
KEY PF1
```

*keyword\_and\_value\_pairs* Note 4

...

### Notes

- 1** *keyword* and *value* pairs before the first SESSION keyword cause updates only in the Enduser settings and the Session Default settings.
- 2** *keyword* and *value* pairs after a SESSION ALL keyword cause updates in all Session settings and in the Session Default settings.
- 3** *keyword* and *value* pairs after a SESSION ANY keyword cause updates in all Session settings but **not** in the Session Default settings.
- 4** *keyword* and *value* pairs after any other SESSION or KEY keyword cause updates only in the specified Session settings.
- 5** *keyword* and *value* pairs that are for the Enduser settings and the Session Default settings **must** be entered before the SESSION or KEY keywords, otherwise they will be ignored.
- 6** SESSION and KEY keywords are mutually exclusive and therefore only one keyword of SESSION or KEY should be specified for any session. Specifying both SESSION and KEY will produce an invalid configuration.

## ISZUPDT return values

ISZUPDT returns the following values:

Return code	Meaning
0	Update was successful.
8	One or more errors occurred; processing continued as CONTINUE was specified.
12	Error in parameter parsing, or I/O error; see audit messages.

## ISZACTV – activate pending updates

ISZACTV activates updates made to the Session Manager configuration files, **without** restarting the instances. Updates normally take effect only when the Session Manager instances are next restarted (and this is always true for DEL updates). ISZACTV reads a control file containing ACTIVATE instructions, typically (but not necessarily) generated by ISZUPDT. The updates can have been applied by running ISZUPDT, or by using the OLA panels.

### Format

```
ISZACTV
  ALL

  INPUT(DDN)
  INPUT(DDN(ddname [mbrname]))
  INPUT(VAR(varname))

  OUTPUT(DDN(ddname [mbrname]) [disp])
  OUTPUT(VAR(varname))
  OUTPUT(AUDIT)

  CONTINUE
  LISTINPUT
  TRACE

  PARMS(DDN(ddname [mbrname]))
  PARMS(VAR(varname))
```

ISZACTV may be called from a TPSL script, in which case the parameters are passed by assigning their values to the `t_scparms` variable. All input datasets should have edit numbering disabled.

**Note** The total length of the inline parameters supplied to ISZACTV must not exceed 256 characters. You can use the PARMS parameter to bypass this restriction.

### Parameters

Except for INPUT, all of the parameters are optional, and may appear in any order.

ALL

The optional ALL parameter specifies that all Session Manager instances are to reload their entire definitions from the configuration files. The ALL and INPUT parameters are mutually exclusive.

```
INPUT(DDN)
INPUT(DDN(ddname [mbrname]))
INPUT(VAR(varname))
```

The INPUT parameter defines the source of control information. Any combination of these parameters may be supplied. See 'ISZACTV input format' on page 106 for details of how the information is structured. The ALL and INPUT parameters are mutually exclusive.



INPUT(DDN) specifies that ISZACTV control information should be read from the ISZACTVI member of the PDS(E) which is currently allocated to CONFIG.

INPUT(DDN(*ddname* [*mbrname*])) specifies that ISZACTV control information should be read from the file which is currently allocated to *ddname*. This can be either a sequential file, or a member of a PDS(E).

INPUT(VAR(*varname*)) specifies that ISZACTV control information should be read from the specified subscripted variable. The first input record is in *varname*.1, the second in *varname*.2, and so on; the total number of input records is in *varname*.0. A subscripted variable can contain up to 9999 records.

OUTPUT(DDN(*ddname* [*mbrname*]) [*disp*])

OUTPUT(VAR(*varname*))

OUTPUT(AUDIT)

The optional OUTPUT parameter defines a location where the update report is to be written. Any combination of these parameters may be supplied. See 'ISZACTV output format' on page 106 for information on how the information is presented.

OUTPUT(DDN(*ddname* [*mbrname*]) [*disp*]) specifies that information should be written to a member of the PDS(E) which is currently allocated to *ddname* (the default for *mbrname* is ISZACTVO). The *disposition* can be REP (the default – replace any existing file contents), MOD (append to any existing file contents), or NEW (fail if there are existing contents).

**Note** The only Session Manager PDS(E) to which *ddname* can safely refer is CONFIG – do not attempt to write to any of the other configuration PDS(E)s.

OUTPUT(VAR(*varname*)) specifies that information should be written to the specified subscripted variable. The first output record is in *varname*.1, the second in *varname*.2, and so on; the total number of output records is in *varname*.0. A subscripted variable can contain up to 9999 records.

OUTPUT(AUDIT) specifies that information should be written to the audit file. This is the default if no OUTPUT parameter is specified.

CONTINUE

The optional CONTINUE parameter causes processing to continue if an error is detected; the default is to terminate.

**Note** Using the CONTINUE parameter will set the return code to eight, regardless of any errors. CONT and CONTMAXCC can also be specified as a control option on the BATSCR script, to alter the final return code setting.

LISTINPUT

The optional LISTINPUT parameter causes the output report to include a copy of the input control information.

TRACE

The optional TRACE parameter produces debugging information.

```
PARMS(DDN(ddname [mbrname]))
PARMS(VAR(varname))
```

The optional `PARMS` parameter defines an additional source of parameter information. Any combination of these parameters may be supplied.

`PARMS(DDN(ddname [mbrname]))` specifies that ISZACTV parameters should be read from the file which is currently allocated to `ddname`. This can be either a sequential file, or a member of a PDS(E).

`PARMS(VAR(varname))` specifies that ISZACTV parameters should be read from the specified subscripted variable. The first input record is in `varname.1`, the second in `varname.2`, and so on; the total number of input records is in `varname.0`. A subscripted variable can contain up to 9999 records.

## ISZACTV input format

ISZACTV expects control information to consist of `ACTIVATE` requests. Additionally, it ignores blank lines, and comments (with “\*” as the first character in the record).

### ACTIVATE a definition

Each `ACTIVATE` request has this format:

```
ACTIVATE DDN(ddname) MBR(mbrname)
```

*ddname*

is currently allocated to a Session Manager configuration dataset – see ‘Configuration DDNAMEs’ section in the ‘Converting a Classic to an OLA system’ chapter and the ‘Customer, samples and system configuration datasets’ section in the ‘Performing a basic configuration’ chapter of the *Installation and Customization* manual.

*mbrname*

is the name of an existing definition in that dataset.

**Note** From Session Manager release 2.2.00 all messages are prefixed with ISZM (for example, ISZM1234), all classes are prefixed with ISZCLS (for example, ISZCLSIM) and all commands are prefixed with ISZ or ISZZ (for example, ISZUPDT and ISZZINIT). For ‘command name to PDSE member name’ conversion table, see ‘Parameters for “File I/O”’ on page 35.

## ISZACTV output format

A typical Activation Report is as follows:

ISZACTV - Activate report

Activate date:09/03/04 time:17:45:31

parameters

CONTINUE  
 INPUT(VAR(LC\_INP)  
 OUTPUT(AUDIT)  
 OUTPUT(DDN(OLD) REP)

ISZ8018 Activate - processing variable LC\_INP

ISZ8070 Activate - successful for USER COM0 on local node  
 JC3ISM

ISZ8075 Activate - successful for USER COM0 on node JC3ISM2

ISZ8070 Activate - successful for USER ATEST on local node  
 JC3ISM

ISZ8075 Activate - successful for USER ATEST on node JC3ISM2

2 successful activate(s) for local node JC3ISM  
 0 failed activate(s) for local node JC3ISM

2 successful activate(s) for remote node JC3ISM2  
 0 failed activate(s) for remote node JC3ISM2

ISZACTV complete rc(0)

## ISZACTV return values

ISZACTV returns the following values:

Return code	Meaning
0	Activation was successful.
4	One or more remote activations failed.
8	One or more errors occurred; processing continued as CONTINUE was specified.
12	Error in parameter parsing, or I/O error; see audit messages.

## ISZREMC – invoke Session Manager command

ISZREMC invokes a Session Manager command. It is typically used as the value of `INITIAL_CMD` in `SYSTEM` definitions – see ‘Batch Administration configuration’ on page 117.

### Format

ISZREMC *command*

ISZREMC (*command parameters*)

### Parameters

*command* [*parameters*]

*command* is any Session Manager command, and *parameters* are any parameters required by that command.

## ISZREMC return values

ISZREMC returns the value returned by the *command*.

## Examples

### See

- 'Creating a user definition' below
- 'Copying a user definition' on page 110
- 'Deleting user definitions' on page 111
- 'Modifying a user definition' on page 112
- 'Creating user definitions using a script' on page 113
- 'Deleting user definitions using a script' on page 114
- 'Copying a user definition using a script' on page 115
- 'Deleting a specific session from all users' on page 115

## Creating a user definition

To create definitions for several users, create and submit the following job.

**Note** A sample job to perform this task can be found in member ISZBASM0, which is supplied in library .SISZCONF. You should modify the job to suit your circumstances before using it to make changes.

```
//BATCHGO EXEC PROC=ISMBATCH
//STEP01.SEQ DD *      COMMAND SCRIPT AND PARAMETERS TO BE EXECUTED
* SET DO NOT TERMINATE THE JOB IF AN ERROR OCCURS
CONT
* UPDATE THE CONFIGURATION
* TAKE INPUT FROM SYSIN (DD *)
* WRITE ACTIVATE DATA TO MEMBER BATCHACT IN DD SYSUT1 (&&TEMP)
* SET DO NOT TERMINATE THE UPDATE IF AN ERROR OCCURS
ISZUPDT
  INPUT(DDN(SYSIN))
  ACTIVATE(DDN(SYSUT1 BATCHACT))
  CONTINUE
* ACTIVATE THE CHANGES
* READ ACTIVATE DATA FROM MEMBER BATCHACT IN DD SYSUT1
ISZACTV
  INPUT(DDN(SYSUT1 BATCHACT))
/*
//STEP01.SYSIN DD *      USER DATA
* Create new user TOM
ADD DDN(USER) MBR(TOM)
password startup
prof exitprof
* Create new user DICK
ADD DDN(USER) MBR(DICK)
password startup
prof exitprof
* Create new user HARRY
ADD DDN(USER) MBR(HARRY)
password startup
prof exitprof
/*
```

In this example, user data is coded in-line rather than in a dataset referenced by the JCL (which makes it much easier to use for simple changes). Also, 'activate' data is written to a temporary dataset that is discarded automatically when the job ends.

## Copying a user definition

To create a definition for a new user *HARRIET* (for example) based on that of an existing user *SYSADMIN*, complete the following steps.

**Note** A sample job to perform these steps can be found in member *ISZBASM1*, which is supplied in library *.SISZCONF*. You should modify the job to suit your circumstances before using it to make changes.

- 1 Create the following output mask for *ISZSRCH*, changing *HARRIET* to the required new user name – see member *ISZMASK1* in the *MASK* dataset (which is supplied in library *.SISZBMSK*):

```
#SRCH_TITLE* EXAMPLE 1
ADD DDN(%SRCH_DDN) MBR(HARRIET)
%SRCH_ALL_KEYW %SRCH_ALL_KEYW_VALUE
```

- 2 Run *ISZSRCH* with the following parameters, to select the definition of *SYSADMIN*:

```
ISZSRCH
STATEMENT(USER)
KEYWORD(IS(USER(SYSADMIN)))
OUTPUT(DDN(UPDATE EXCTRL1) REP MASK(DDN(MASK ISZMASK1)))
```

The output is an *ADD* request in *UPDATE(EXCTRL1)*:

```
* EXAMPLE 1
ADD DDN(USER) MBR(HARRIET)
keyword1 value1
keyword2 value2
...
keywordN valueN
```

- 3 Run *ISZUPDT* with the following parameters, to add the definition of *HARRIET*:

```
ISZUPDT
INPUT(DDN(UPDATE EXCTRL1))
ACTIVATE(DDN(ACTIVE EXACTV1) REP)
```

The output is an *ACTIVATE* request in *ACTIVE(EXACTV1)*:

```
ACTIVATE DDN(USER) MBR(HARRIET)
```

- 4 Run *ISZACTV* with the following parameter, to activate the definition of *HARRIET*:

```
ISZACTV
INPUT(DDN(ACTIVE EXACTV1))
```

## Deleting user definitions

To delete the definitions of existing users that have a prefix of *HARR\**, complete the following steps.

**Note** A sample job to perform these steps can be found in member ISZBASM2, which is supplied in library .SISZCONF. You should modify the job to suit your circumstances before using it to make changes.

- 1 Create the following output mask for ISZSRCH – see member ISZMASK2 in the MASK dataset (which is supplied in library .SISZBMSK):

```
#SRCH_TITLE* EXAMPLE 2
DEL DDN(%SRCH_DDN) MBR(%SRCH_MBR)
```

- 2 Run ISZSRCH with the following parameters, to select the definitions of *HARR\**:

```
ISZSRCH
STATEMENT(USER)
KEYWORD(IS(USER(HARR*)))
OUTPUT(DDN(UPDATE EXCTRL2) REP MASK(DDN(MASK ISZMASK2)))
```

The output is a series of DEL requests in UPDATE(EXCTRL2), for example:

```
* EXAMPLE 2
DEL DDN(USER) MBR(HARRIET)
DEL DDN(USER) MBR(HARRY)
...
```

- 3 Run ISZUPDT with the following parameter, to delete the definitions (there is no output):

```
ISZUPDT
INPUT(DDN(UPDATE EXCTRL2))
```

## Modifying a user definition

To modify an existing user statement named *HARRIET* by:

- changing the profile to SYSPROGS,
- replacing all references of ISZACICA with ISZACICB in any DESC parameter,
- modifying session 21 to refer to an IMS APPL definition, or adding the session if it didn't exist,
- deleting all MISER parameters in session definitions, but leaving any MISER session default parameter,
- deleting the PF3 session,
- deleting the DESC parameter on the PF2 session.

complete the following steps.

**Note** A sample job to perform these steps can be found in member ISZBASM3, which is supplied in library .SISZCONF. You should modify the job to suit your circumstances before using it to make changes.

- 1 Create an ISZUPDT control file – see member ISZEXCT3 in the UPDATE dataset (which is supplied in library .SISZBUPD):

```
* EXAMPLE 3
MOD DDN(USER) MBR(HARRIET)

* Change the profile parameter to sysprogs
prof sysprogs

* Replace references to 'ISZACICA' in all DESCRIPTION
parameters
* with 'ISZACICB'
session all
#REPLACE description ISZACICA ISZACICB

* Modify or define session 21 APPL parameter
session 21
applid IMS

* Remove any miser definitions on all sessions, but leave
* the session defaults
session any
#DELETE miser

* Remove session key pf3
#DELETE key pf3

* Remove the DESC parameter from session key pf2
key pf2
#DELETE description
```

- 2 Run ISZUPDT with the following parameter, to modify the definition (there is no output.):



```
ISZUPDT
  INPUT(DDN(UPDATE ISZEXCT3))
```

## Creating user definitions using a script

To create 3000 user definitions by:

- adding a script,
- activating it,
- executing the script,

complete the following steps.

**Note** A sample job to perform these steps can be found in member ISZBASM4, which is supplied in library .SISZCONF. You should modify the job to suit your circumstances before using it to make changes.

- 1 Run ISZUPDT with the following parameters – see member ISZADUSR in the UPDATE dataset (which is supplied in library .SISZBUPD):

```
ISZUPDT
  INPUT(DDN(UPDATE ISZADUSR))
  ACTIVATE(DDN(ACTIVE ACTIVATE))
```

This will add a script called ISZADUSR to the ASCRIPT dataset.

- 2 Run ISZACTV to activate script ISZADUSR in the ASCRIPT dataset:

```
ISZACTV
  INPUT(DDN(ACTIVE ACTIVATE))
```

- 3 Run script ISZADUSR to build the lc\_newuser subscripted variables:

```
ISZADUSR
```

- 4 Run ISZUPDT using the lc\_newuser variables as input. This will add the new USER statements to the USER dataset:

```
ISZUPDT INPUT(VAR(LC_NEWUSER))
  ACTIVATE(DDN(ACTIVE ACTIVATE))
```

- 5 Run ISZACTV to activate the newly added users:

```
ISZACTV
  INPUT(DDN(ACTIVE ACTIVATE))
```

- 6 Run ISZUPDT with the following parameters – see member ISZTDUP4 in the UPDATE dataset (which is supplied in library .SISZBUPD):

```
ISZUPDT INPUT(DDN(UPDATE ISZTDUP4))
```

This will delete script ISZADUSR from the ASCRIPT dataset.

## Deleting user definitions using a script

To delete the 3000 user definitions created in the previous example by:

- adding a script,
- activating it,
- executing the script,

complete the following steps.

**Note** A sample job to perform these steps can be found in member ISZBASM5, which is supplied in library .SISZCONF. You should modify the job to suit your circumstances before using it to make changes.

- 1 Run ISZUPDT with the following parameters – see member ISZDLUSR in the UPDATE dataset (which is supplied in library .SISZBUPD):

```
ISZUPDT
  INPUT(DDN(UPDATE ISZDLUSR))
  ACTIVATE(DDN(ACTIVE ACTIVATE))
```

This will add a script called ISZDLUSR to the ASCRIPT dataset.

- 2 Run ISZACTV to activate script ISZDLUSR in the ASCRIPT dataset:

```
ISZACTV
  INPUT(DDN(ACTIVE ACTIVATE))
```

- 3 Run script ISZDLUSR to build the lc\_deluser subscripted variables:

```
ISZDLUSR
```

- 4 Run ISZUPDT using the lc\_deluser variables as input. This will delete the USER statements:

```
ISZUPDT INPUT(VAR(LC_DELUSER))
```

- 5 Run ISZUPDT with the following parameters – see member ISZTDUP5 in the UPDATE dataset (which is supplied in library .SISZBUPD):

```
ISZUPDT INPUT(DDN(UPDATE ISZTDUP5))
```

This will delete script ISZDLUSR from the ASCRIPT dataset.

## Copying a user definition using a script

To create a definition for a user *HARRIET* based on that of an existing user *SYSADMIN* by:

- adding a script,
- activating it,
- executing the script,

complete the following steps.

**Note** A sample job to perform these steps can be found in member ISZBASM6, which is supplied in library .SISZCONF. You should modify the job to suit your circumstances before using it to make changes.

- 1 Run ISZUPDT with the following parameters – see member ISZCPUSR in the UPDATE dataset (which is supplied in library .SISZBUPD):

```
ISZUPDT
  INPUT(DDN(UPDATE ISZCPUSR))
  ACTIVATE(DDN(ACTIVE ACTIVATE))
```

This will add a script called ISZCPUSR to the ASCRIPT dataset.

- 2 Run ISZACTV to activate script ISZCPUSR in the ASCRIPT dataset:

```
ISZACTV
  INPUT(DDN(ACTIVE ACTIVATE))
```

- 3 Run script ISZCPUSR to create and activate a new user called *HARRIET* based on user *SYSADMIN*:

```
ISZCPUSR
```

- 4 Run ISZUPDT with the following parameters – see member ISZTDUP6 in the UPDATE dataset (which is supplied in library .SISZBUPD):

```
ISZUPDT INPUT(DDN(UPDATE ISZTDUP6))
```

This will delete script ISZCPUSR from the ASCRIPT dataset.

## Deleting a specific session from all users

To delete all sessions with APPLID set to ISZATSO (for example), complete the following steps.

**Note** A sample job to perform these steps can be found in member ISZBASM7, which is supplied in library .SISZCONF. You should modify the job to suit your circumstances before using it to make changes.

- 1 Create the following output mask for ISZSRCH, changing ISZATSO to the required session APPLID name – see member ISZMASK7 in the MASK dataset (which is supplied in library .SISZBMSK):

```
#SRCH_TITLE* ISZSRCH performed on %SRCH_DATE %SRCH_TIME
#SRCH_TITLE* specified parameters were
#SRCH_TITLE* %SRCH_PARMs
#SRCH_TITLE* NOTE - only 1 (one) space between parms if you
#SRCH_TITLE*          want to preserve current format.
#SRCH_TITLE*
MOD DDN(%SRCH_DDN) MBR(%SRCH_MBR)
#DELETE %SRCH_MATCH_SESS
```

- 2** Run ISZSRCH with the following parameters, to select the all the USER definitions that have a session matching the search criteria above.

```
ISZSRCH
STATEMENT(USER)
KEYWORD(IS(APPLID(ISZATS0)))
OUTPUT(DDN(UPDATE EXCTRL7) REP MASK(DDN(MASK ISZMASK7)))
```

The output is a MOD with #DELETE request in UPDATE(EXCTRL7) where xxxxxxxx and yyyyyyyy are the USER names and nnnn are the session ids.

There will be a MOD and a #DELETE for each USER that has a ISZATS0 APPLID specified:

```
* ISZSRCH performed on 20/06/08 14:15:16
* specified parameters were
* STATEMENT(USER)
* KEYWORD(IS(APPLID(ISZATS0)))
* OUTPUT(DDN(UPDATE EXCTRL7) REPMASK(DDN(MASK ISZMASK7)))
* NOTE - only 1 (one) space between parms if you want
*          to preserve current format.
*
MOD DDN(USER) MBR(xxxxxxxx)
#DELETE SESSION nnnn
MOD DDN(USER) MBR(yyyyyyyy)
#DELETE SESSION nnnn
```

- 3** Run ISZUPDT with the following parameters, to delete the sessions:

```
ISZUPDT
INPUT(DDN(UPDATE EXCTRL7))
OUTPUT(DDN(UPDATE EXUPDT7) REP)
ACTIVATE(DDN(ACTIVE EXACTV7) REP)
```

The output are ACTIVATE requests in ACTIVE(EXACTV7):

```
ACTIVATE DDN(USER) MBR(xxxxxxxx)
ACTIVATE DDN(USER) MBR(yyyyyyyy)
```

- 4** Run ISZACTV with the following parameter, to activate the modified users:

```
ISZACTV
INPUT(DDN(ACTIVE EXACTV7))
```

## Batch Administration configuration

**Note** The material in this section is provided for reference purposes. If you are an existing user of a Classic configuration (that is, all configuration definitions are stored in members of PDS(s) allocated to the DDNAME of CONFIG) then the OLA Enabler performs most of the configuration steps automatically – see ‘OLA Enabler’ in the ‘Converting a Classic to an OLA system’ chapter of the *Installation and Customization* manual.

The Batch Administration program is actually a normal instance of Session Manager, albeit one with a specialized configuration definition. The definition comprises a CONFIG(ISZCONxx) member and a SYSTEM(ISZSYSxx) member. For general information on configuration files see the ‘Configuration DDNAMEs’ section and the ‘Multiple Session Manager instances sharing a configuration’ section in the ‘Converting a Classic to an OLA system’ chapter of the *Installation and Customization* manual.

**Note** When you run the OLA Enabler, the tool automatically creates appropriate members having BT as the value of xx. These members are also supplied with the product.

The ISZCONBT member is created to contain these statements:

```
OPTION START=NO
GFS Y
STORLIM 72M
THRESHOLD LOG
COPY ISZCINIT
INSTALLSU
%% let gc_acb_prefix = 'acb_prefix'
%% let gc_acb_suffix = 'acb_suffix'
AUDITROUTE CLASS=A
TRACERoute CLASS=A
COPY ISZCOMON
```

The ISZSYSBT member is created by copying the equivalent SYSTEM member for your standard online configuration, and then making a small number of changes:

```
INITIAL_CMD=command_spec
```

This parameter must be inserted; see ‘INITIAL\_CMD parameter’ on page 119.

**Note** When you run the OLA Enabler, the INITIAL\_CMD is set to “BATSCR”; see ‘Executing the batch program’ on page 81.

```
LOCALNODE=nodename
```

Choose a suitable name for this Batch instance of Session Manager.

**Note** When you run the OLA Enabler, the LOCALNODE is set to “BATCH”.

```
ACB=0
SESACB=0
```

ACB and SESACB must be defined as zero.

```
TCP=STN3270
```

Any parameters of this form must be removed.

**CAUTION** Do not activate the definition yet. When setup is complete, the instance can be started in the usual manner, whereupon it initializes itself, executes the command script specified by the `INITIAL_CMD=command_spec` parameter of its `SYSTEM` statement, and then terminates (because of the `START=NO` parameter of its `OPTION` statement).

In addition, Online and Batch Administration requires:

- for a non-Sysplex environment, a full set of `LINK(nodename)` members specifying the links to each Session Manager instance which is to be administered. The required `LINK` statements may be generated by the Enabler process or created in an OLA system. For more information on `LINKs` generated by the Enabler process see the example in the ‘Converting a Classic to an OLA system’ chapter in the *Installation and Customization* manual.

The table below shows the recommended set-up for one instance and Batch:

Proc/job	ISMSTC01	ISMBATCH
Config	ISZCON02	ISZCONBT
Local node	Online1	Batch
Main acb	ISZSMGR	none
gc_acb_pre	SX9	S09
gc_acb_suf	ON1	BAT
gc_common	SD	SD
Opens acbs	SX9SDBAT	S09SDON1
Links	BATCH <-----> (via SX9SDBAT)	ONLINE1 (via S09SDON1)

**Note** `gc_acb_pre`, `gc_acb_suf` and `gc_common` are substitution variables that should be defined in your `ISZCONxx` members.

The table below shows the recommended set-up for two instances and Batch:

Proc/job	ISMSTC01	ISMBATCH	ISMSTC02
Config	ISZCON02	ISZCONBT	ISZCON03
Local node	Online1	Batch	Online2
Main acb	ISZSMGR	none	ISZSMGR2
gc_acb_pre	SX9	S09	S29
gc_acb_suf	ON1	BAT	ON2
gc_common	SD	SD	SD
Opens acbs	SX9SDBAT SX9SDON2	S09SDON1 S09SDON2	S29SDBAT S29SDON1
Links	BATCH <-----> ONLINE1 (via SX9SDBAT) (via S09SDON1)		
	ONLINE2 <-----> BATCH (via S09SDON2) (via S29SDBAT)		
	ONLINE2 <-----> ONLINE1 (via SX9SDON2) (via S29SDON1)		

**Note** gc\_acb\_pre, gc\_acb\_suf and gc\_common are substitution variables that should be defined in your ISZCONxx members.

For a Sysplex environment the SYSPLEXGROUP parameter should already be specified in the ISZSYSCM member. The Batch system will administer all the Session Manager instances that have a matching SYSPLEXGROUP name.

- for a non-Sysplex environment, an RUSER(ISZSMGR) member specifying the user authorized to run Batch Administration:

```
RUSER ISZSMGR
      NODE=BATCH
      AUTH=9
      OLAClass=IM
```

For a Sysplex environment RUSER statements are not required for nodes within a SYSPLEXGROUP.

**Note** These members are automatically created when you run the OLA Enabler.

## INITIAL\_CMD parameter

The *command\_spec* argument may be defined in several ways:

```
INITIAL_CMD=script
INITIAL_CMD=(script parameters)
```

where *script* is a script that exists at this instance of Session Manager, and any *parameters* are passed to it in *t\_scparms*.

**Note** We recommend that, to avoid contention on the configuration files, you use this method only when no online instances of Session Manager are active.

```
INITIAL_CMD="ISZREMC command"  
INITIAL_CMD="ISZREMC (command parameters)"
```

where *command* is a command that exists at this instance of Session Manager, and any *parameters* are passed to it in *t\_scparms*.

```
INITIAL_CMD="ISZREMC 'SEND nodename command' "  
INITIAL_CMD="ISZREMC 'SEND nodename (command parameters)' "
```

where *command* is a remote command or script that exists at the *nodename* instance of Session Manager, and any *parameters* are passed to it in *t\_scparms*.

Optionally, *command* may have been mentioned in a Session Manager COMMAND statement (but this offers no extra capabilities):

```
COMMAND command CMDSCRIPT=YES
```

See also 'ISZREMC – invoke Session Manager command' on page 108.



# Index

## A

Accessibility 127  
 addsestype parameter 23  
 APPLYSU statement 100  
 Asterisk (\*)  
   as a wildcard character 51  
 Attribute definitions  
   Codes for List actions 61  
   Highlighting 43  
   Immediate activation of changes 64  
   Multi-page lists 43  
   Pending changes 62, 63  
   Resetting changes 65  
   Saving changes 63  
   Saving changes, temporarily 64  
 AUDITROUTE statement 100, 117  
 AUTH parameter 29  
 AUTOSTART column (on session list) 66  
 Autostart setting 69

## B

Batch Administration 77–120  
 BATSCR script 81

## C

Classes, example OLA security 29  
 CONFDSN= parameter 24  
 Configuration changes 14, 16  
 Configuration data 14, 39  
 Configuration definitions 41  
   ‘default options’ panel 53  
   ‘list’ panel 51  
   Delete confirmation 45, 54  
   Filtering 51  
   Immediate activation of changes 53

Multi-page lists 44  
 Saving 54

COPY statement 100

## D

DEFMENU parameter (of SYSTEM statement) 17  
 Disabled access 127  
 DSNPREF= parameter 24  
 Dynamic menus 16

## E

E22 security exit 20  
 E22 user exit 25  
 Escape processing 46  
 ESM (External Security Manager) 55  
 Exit E21 55  
 External Security Manager (ESM) 55

## F

Filtering member and session lists 42

## G

gc\_acb\_prefix variable 117  
 gc\_acb\_suffix variable 117  
 GFS 117

## H

Help 16  
 Hidden application sessions  
   Description of 20  
   Specifying 20  
 HIDE common session parameter 20  
 Hummingbird HostExplorer 127

**I**

INSTALLSU statement 100, 117  
 ISZ command prefix 35  
 ISZACTV command 79, 104–107  
 ISZADUSR supplied member 113  
 ISZBAJOB batch job 81  
 ISZBASM0 supplied member 109  
 ISZBASM1 supplied member 110  
 ISZBASM2 supplied member 111  
 ISZBASM3 supplied member 112  
 ISZBASM4 supplied member 113  
 ISZBASM5 supplied member 114  
 ISZBASM6 supplied member 115  
 ISZCINIT configuration member 117  
 ISZCLS class prefix 35  
 ISZCONBT configuration member 117  
 ISZCONxx configuration member 100, 117  
 ISZCPUSR supplied member 115  
 ISZDFMSK supplied member 90  
 ISZDLUSR supplied member 114  
 ISZEXCT3 supplied member 112  
 ISZLENPU sample menus 17  
 ISZM message prefix 35  
 ISZMASK1 supplied member 110  
 ISZMASK2 supplied member 111  
 ISZREMC command 108, 120  
 ISZSBAT supplied member 81  
 ISZSRCH command 79, 85–92  
 ISZSTJOB batch job 23  
 ISZSTJOB sample job 23  
 ISZSYSBT configuration member 117  
 ISZSYSCM configuration member 74, 99  
 ISZSYSxx configuration member 74, 99, 117  
 ISZTDUP4 supplied member 113  
 ISZTDUP5 supplied member 114  
 ISZTDUP6 supplied member 115  
 ISZUPDT command 78, 93–103  
 ISZZ command prefix 35

**J**

Jaws 127

**L**

LOADDSN= parameter 24

**M**

Members, configuration  
   ISZCINIT 117  
   ISZCONBT 117  
   ISZCONxx 100, 117  
   ISZSYSBT 117

ISZSYSCM 74, 99  
 ISZSYSxx 74, 99, 117

Members, supplied

ISZADUSR 113  
 ISZBASM0 109  
 ISZBASM1 110  
 ISZBASM2 111  
 ISZBASM3 112  
 ISZBASM4 113  
 ISZBASM5 114  
 ISZBASM6 115  
 ISZCPUSR 115  
 ISZDFMSK 90  
 ISZDLUSR 114  
 ISZEXCT3 112  
 ISZMASK1 110  
 ISZMASK2 111  
 ISZSBAT 81  
 ISZTDUP4 113  
 ISZTDUP5 114  
 ISZTDUP6 115

Menu sequence numbers for sessions

Description of 21  
 Specifying 21

Menus

Dynamic 16  
 Sample 17

**O**

OLA

Facilities 14  
 Introduction to 13  
 Menus, lists and displays, *see* OLA  
   menus, lists and displays  
 Online Help 16  
 Security, *see* OLA security

OLA Hardcopy menu

Hardcopy formats 71  
 Hardcopy profiles 71  
 Hardcopy routes 72

OLA Main Menu

APPL definitions 55  
 GROUP definitions 56  
 Hardcopy definitions 56  
 Local USER definitions 54  
 My USER definition 50  
 PROFILE definitions 51  
 Remote USER definitions 55  
 Security definitions 57  
 SYSTEM definitions 56  
 TERMINAL definitions 56

OLA menus, lists and displays

Codes for List actions 61  
 Configuration definition ‘default  
   options’ panel 53

- Configuration definition 'list' panel 51
- Delete confirmation 45, 54
- Details of 39
- Filtering configuration definitions 51
- Hardcopy menu, *see* OLA Hardcopy menu
- Highlighting of attribute definitions 43
- Immediate activation of changes 53, 64
- Keywords used for display only 47
- Main Menu, *see* OLA Main Menu
- Multi-page attribute definition lists 43
- Multi-page configuration definition lists 44
- Overview of how to use 41
- Pending changes 62, 63
- Resetting changes 65
- Saving changes to attributes 63
- Saving changes to attributes, temporarily 64
- Saving changes to configuration definitions 54
- Selecting a menu option 42
- Summary list of 14
- Summary of PK Key settings 48
- Summary of selection options 47
- System menu, *see* OLA System menu
- USER Definition menu, *see* OLA USER Definition menu
- Using 'Attribute Expansion' displays 45
- Using lists 42
- Using menus 41
- OLA security
  - Example classes 29
  - OLAClass attribute 29
  - Overview of 28
  - Security classes, assigning a user to 30
  - Security classes, description of 29
  - Security classes, example classes 29
  - Security settings, description of 32
  - Security settings, list of 32
  - Setting up, for menu items 34
  - Setting up, for miscellaneous entries 34
  - Setting up, for parameter definitions 34
  - Setting up, overview of 33
- OLA set up 14
- OLA System menu
  - ACB RANGE 73
  - Command authority 74
  - Link definitions 73
  - Message statement 74
  - SYSTEM statement 74
- OLA USER Definition menu
  - Common enduser parameters 60

- Common session parameters 65
- Enduser specific parameters 58
- List of sessions 66
- OLAClass common enduser parameter 28
- OLAClass common enduser parameter 29, 100
- OLARENAME subparameter 30
- Online Administration, *see* OLA
- Online Help 16
- OPTION statement 82, 100, 117

## P

- Panel and Script Language 9
- Parameter validation 46
- PASSWORD parameter 29
- Plus (+)
  - as a wildcard character 51
- Priority sessions 21, 23, 68
- PUPDATE command 44

## R

- REMOVESU statement 100
- RUSER statements, and Sysplex 101

## S

- s\_hidden variable 17, 20
- s\_sequence variable 17, 21
- s\_sessdata variables 65
- s\_sesstype variable 22
- Sample menus 17
- Screen reader 127
- SECURITY parameter 28
- Security, *see* OLA security
- SEQUENCE common session parameter 21
- SESSDATA fields 65
- Session List 23
- Session list operations 67
- Session types for sessions
  - Description of 22
  - Specifying 22
- Sessions
  - duplicates 16
  - hidden, description of 20
  - hidden, specifying 20
  - menu sequence numbers, description of 21
  - menu sequence numbers, specifying 21
  - order on menus 16
  - priority 21, 23
  - session types, description of 22
  - session types, specifying 22

SESSPRIAPPL common enduser  
    parameter 23  
SESTYPE 23  
SESTYPE common session parameter 22,  
    23  
Setting session priority 21, 23, 68  
STORLIM 117  
Sysplex, and RUSER statements 101

## T

t\_scparms variable 85, 93, 104, 119  
t\_security\_class variable 31  
t\_userdata variables 61  
THRESHOLD 117  
TPSL 9  
TRACE parameter 29, 34  
TRACEROUTE statement 100, 117

## U

USERDATA fields 61  
utostart setting 38

## V

Variables  
    gc\_acb\_prefix 117  
    gc\_acb\_suffix 117  
    s\_hidden 17, 20  
    s\_sequence 17, 21  
    s\_setype 22  
    t\_security\_class 31

## W

WindowEyes 127

---

# Bibliography

---

## IBM Session Manager library

The following publications contain information about IBM Session Manager.

	<i>Installation and Customization</i>	GC34-7146-00
	<i>Technical Reference</i>	SC34-7147-00
	<i>User and Administrator</i>	SC34-7150-00
	<i>Panels, Scripts and Variables</i>	SC34-7148-00
	<i>Messages and Codes</i>	GC34-7152-00
	<i>Quick Reference</i>	SC34-7151-00
	<i>Online and Batch Administration</i>	SC34-7149-00
	<i>Program Directory</i>	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

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (<sup>®</sup> or <sup>™</sup>), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at “Copyright and trademark information” at <http://www.ibm.com/legal/copytrade.shtml>

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Other company, product, and service names may be trademarks or service marks of others.

---

## 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.



SC34-7149-00