

Interactive System Productivity Facility (ISPF)



# Planning and Customizing

*z/OS Version 1 Release 11.0*



Interactive System Productivity Facility (ISPF)



# Planning and Customizing

*z/OS Version 1 Release 11.0*

**Note**

Before using this document, read the general information under "Notices" on page 311.

**Ninth Edition (September 2009)**

This edition applies to ISPF for Version 1 Release 11.0 of the licensed program z/OS (program number 5694-A01) and to all subsequent releases and modifications until otherwise indicated in new editions.

IBM® welcomes your comments. A form for comments appears at the back of this publication. If the form has been removed and you have ISPF-specific comments, address your comments to:

IBM Corporation  
Reader Comments  
DTX/E269  
555 Bailey Avenue  
San Jose, CA 95141-1003  
U.S.A.

Internet: [comments@us.ibm.com](mailto:comments@us.ibm.com)

If you would like a reply, be sure to include your name and your address, telephone number, e-mail address, or FAX number.

Make sure to include the following in your comment or note:

- Title and order number of this document
- Page number or topic related to your comment

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

The ISPF development team maintains a site on the World Wide Web. The URL for the site is: <http://www.ibm.com/software/awdtools/ispf/>

© Copyright International Business Machines Corporation 1984, 2009.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

# Contents

## Preface . . . . . vii

About this document . . . . .	vii
Who should use this document . . . . .	vii
What is in this document? . . . . .	vii
The z/OS Basic Skills Information Center . . . . .	viii
How to read the syntax diagrams . . . . .	viii

## Summary of changes . . . . . xiii

Product function changes for z/OS V1R11.0 ISPF . . . . .	xiii
ISPF product changes. . . . .	xiii
ISPF Dialog Manager component changes . . . . .	xiii
ISPF PDF component changes . . . . .	xiv
ISPF SCLM component changes . . . . .	xiv
ISPF Client/Server component changes . . . . .	xiv
Migration considerations. . . . .	xiv
Changes to this document for z/OS V1R11.0 ISPF . . . . .	xv

## What's in the z/OS V1R11.0 ISPF library? . . . . . xvii

## Part 1. Planning and customizing . . . . . 1

### Chapter 1. z/OS V1R11.0 ISPF planning . . . . . 3

Hardware and software requirements . . . . .	3
Notes on migrating from previous releases to z/OS V1R11.0 ISPF . . . . .	3
Installation, maintenance, and migration diagnostics . . . . .	4
Language Environment considerations. . . . .	5
Software Configuration and Library Manager (SCLM) . . . . .	5
Maintenance considerations . . . . .	6
Load module search order. . . . .	6
TSO logon procedure . . . . .	7

### Chapter 2. The ISPF Configuration Table . . . . . 9

Using the ISPF Configuration utility before installation is complete . . . . .	10
System requirements . . . . .	10
Running the Configuration utility with a special logon procedure. . . . .	10
The ISPF Configuration utility . . . . .	12
Create/Modify Settings and Regenerate Keyword File . . . . .	14
Edit Keyword File Configuration Table . . . . .	31
Verify Keyword Table Contents. . . . .	33
Build Configuration Table Load Module. . . . .	33
Convert Assembler Configuration Table to Keyword File. . . . .	35
Build SMP/E USERMOD. . . . .	35

### Chapter 3. Customizing ISPF . . . . . 39

Allocating optional image ISPF library . . . . .	39
Customizing action bars . . . . .	39

Invoking the ISPD TLC conversion utility . . . . .	40
Customizing for profile sharing. . . . .	45
Customizing the ISPF Configuration utility. . . . .	45
Temporary data set names . . . . .	46
SYSIKJUA enqueue. . . . .	46
Limitation of other products. . . . .	46
TSO/ISPF client gateway. . . . .	46
Gateway description . . . . .	46
Installing and customizing the gateway . . . . .	47
Using the gateway . . . . .	51

### Chapter 4. Improving ISPF performance . . . . . 67

Use virtual I/O . . . . .	67
Remove or tailor Edit functions. . . . .	67
Change the block size of the edit recovery data set . . . . .	67
Disable SETUNDO STORAGE . . . . .	68
Disable Edit extended highlighting . . . . .	68
Remove action bars from ISPF panels. . . . .	68
Remove scrollable areas from ISPF panels . . . . .	69
Preprocess all ISPF panels . . . . .	69
Preprocessed panel utility . . . . .	69
Add load modules to the pageable link-pack area (LPA) . . . . .	71
Moving load modules . . . . .	71
ISPF load module descriptions . . . . .	72
Allocate execution data sets . . . . .	73
Disable generic high-level qualifiers . . . . .	74
Preallocate ISPF temporary data sets to VIO . . . . .	76
Compress SCLM listings . . . . .	78

### Chapter 5. Customizing DM . . . . . 79

Changing the default name for the ISPF Client/Server component download data set . . . . .	79
Set up ISPF GUI for TSO Line Mode support . . . . .	79
SMF command accounting . . . . .	80
Preallocation of List/Log data sets to SYSOUT . . . . .	83
Specifying the maximum number of split screens. . . . .	83
Setting ISPF site-wide defaults . . . . .	84
Customizing command tables . . . . .	84
Application command table . . . . .	85
User command tables . . . . .	85
Site command tables . . . . .	85
System command table . . . . .	86
Creating ISPF terminal translation tables. . . . .	86
Uppercase character translation table . . . . .	87
Lowercase character translation table . . . . .	87
Valid terminal output translation table . . . . .	87
Generic string master translation table . . . . .	88
Alphabetic character translation tables . . . . .	89
Collating sequence translation table . . . . .	89
Specifying terminal types. . . . .	90
Creating ISPF code page translation tables . . . . .	95
Base code pages for terminals . . . . .	97
ISPD SID translation load modules . . . . .	98

Adding translation tables for extended code page support . . . . .	98
Displaying square brackets used in C programs	104
ISPEXEC processing . . . . .	104
ISPF-to-APL2 terminal type mappings . . . . .	105
Load APL2 workspace . . . . .	105
Tailoring ISPF defaults . . . . .	106
Customizing the ISPF TSO command table (ISPTCM). . . . .	106
Sample ISPTCM definition . . . . .	109
ISPTCM usage notes . . . . .	109
Alternate option 7.1 panels . . . . .	110
ISPF multicultural support . . . . .	111
Changing the session language default value	111

## Chapter 6. ISPF installation-wide exits 113

How to install the installation-wide exits . . . . .	114
How to use the macros to define ISPXDT . . . . .	115
Sample ISPXDT definition . . . . .	118
Exit parameter list . . . . .	118
Error processing . . . . .	119

## Chapter 7. Customizing PDF . . . . . 121

Edit mode defaults . . . . .	121
Site-wide edit profile initialization . . . . .	122
Creating a ZDEFAULT profile . . . . .	122
Action bars and extended color in Edit . . . . .	123
Edit backup and recovery . . . . .	123
Data set allocation defaults for the Outlist utility	124
Using the Hardcopy utility with DBCS support . . . . .	125
SCLM Batch considerations. . . . .	125
Converting from LMF To SCLM . . . . .	127
General conversion . . . . .	127
Converting LMF dialogs to SCLM . . . . .	129
Creating PDF translation tables . . . . .	129
PDF Foreground and Batch customizing . . . . .	130
Foreground processing panels and CLISTs. . . . .	130
Batch processing panels, CLISTs, and skeletons	145
ISRSCAN and ISRLEMX programs . . . . .	163
Customizing Browse and Edit . . . . .	167
Providing customized Browse and Edit panels	168
Enabling Browse panels to display Unicode data	172
Customizing member list panels . . . . .	172
Customizing the z/OS UNIX directory list panel	175
IBM Products option . . . . .	175

## Part 2. Reference . . . . . 179

### Chapter 8. PDF installation-wide exits 181

Data set list filter exit. . . . .	181
Analyze Dsname and Volume . . . . .	181
Analyze data set name . . . . .	182
Return codes . . . . .	182
Data set allocation exit . . . . .	182
Exit parameters. . . . .	182
Return codes . . . . .	186
Activity monitoring exits . . . . .	186
Exit parameters. . . . .	186
Usage notes . . . . .	186
Exit 1: BROWSE primary command start . . . . .	187

Exit 2: BROWSE primary command end . . . . .	187
Exit 3: EDIT primary command start . . . . .	187
Exit 4: EDIT primary command end . . . . .	187
Exit 5: EDIT macro start . . . . .	188
Exit 6: EDIT macro end . . . . .	188
Exit 7: EDIT line command start . . . . .	188
Exit 8: EDIT line command end . . . . .	188
Exit 9: Library/Data Set utility sub-function start	188
Exit 10: Library/Data Set utility sub-function end	189
Exit 11: Data Set List utility line command start	189
Exit 12: Data Set List utility line command end . . . . .	190
Error processing . . . . .	190
Return codes . . . . .	190
Data set list line command exit . . . . .	190
Exit parameters. . . . .	190
Usage notes . . . . .	190
Error processing . . . . .	191
Return codes . . . . .	191
Print utility exit . . . . .	192
Exit parameters. . . . .	192
Return codes . . . . .	193
Error processing . . . . .	193
Print utility exit on ISPF termination and	
LOG/LIST commands . . . . .	193
Exit parameters. . . . .	194
Return codes . . . . .	195
Error processing . . . . .	196
Compress request exit . . . . .	196
Exit parameters. . . . .	196
Return codes . . . . .	197
Error processing . . . . .	197
Data set name change exit . . . . .	197
Exit parameters. . . . .	198
Usage notes . . . . .	199
Member list filter exit . . . . .	199
Analyze patterns . . . . .	200
Analyze member names. . . . .	200
Member list built-in line command exit. . . . .	201
Exit parameters. . . . .	201
Usage notes . . . . .	201
Error processing . . . . .	202
Return codes . . . . .	202

### Chapter 9. ISPF Configuration Table keywords and values . . . . . 203

PDF exits. . . . .	203
Data set allocation settings . . . . .	204
Outlist data set specifications . . . . .	205
SuperC data set specifications . . . . .	205
LMF . . . . .	206
Edit recovery data set specifications . . . . .	206
Move/Copy settings . . . . .	207
DSLISIT removable media interface settings . . . . .	207
Edit-related settings . . . . .	209
Edit site-wide profile customizations . . . . .	212
ISPF site-wide profile customizations . . . . .	217
Default CUA color settings . . . . .	227
Miscellaneous settings . . . . .	231
Values formerly in ISPDFLTS . . . . .	235
VSAM data set restrictions . . . . .	235
ISPSPROF general values . . . . .	236

ISPSPROF workstation defaults . . . . .	243
ISPSPROF workstation download defaults. . . . .	247
<b>Chapter 10. Exits . . . . .</b>	<b>249</b>
Exit macros . . . . .	249
Exit 1: ISPF session initialization exit . . . . .	251
Exit parameters. . . . .	251
Return codes . . . . .	251
Exit 2: ISPF session termination exit . . . . .	251
Exit parameters. . . . .	252
Return codes . . . . .	252
Exit 3: SELECT service start exit . . . . .	252
Exit parameters. . . . .	252
Return codes . . . . .	253
Exit 4: select service end exit . . . . .	254
Exit parameters. . . . .	254
Return codes . . . . .	255
Exit 5: TSO command start exit . . . . .	255
Exit parameters. . . . .	255
Return codes . . . . .	256
Exit 6: TSO command end exit . . . . .	256
Exit parameters. . . . .	256
Return codes . . . . .	256
Exit 7: LIBDEF service exit . . . . .	257
Exit parameters. . . . .	257
Return codes . . . . .	257
Exit 8: RESERVE exit . . . . .	258
Exit parameters. . . . .	258
Return codes . . . . .	258
Exit 9: RELEASE exit . . . . .	258
Exit parameters. . . . .	259
Return codes . . . . .	259
Exit 10: logical screen start exit . . . . .	259
Exit parameters. . . . .	259
Return codes . . . . .	259
Exit 11: logical screen end exit. . . . .	259
Exit parameters. . . . .	260
Return codes . . . . .	260
Exit 12: ISPF service start exit . . . . .	260
Exit parameters. . . . .	260
Return codes . . . . .	260
Exit 13: ISPF service end exit . . . . .	260
Exit parameters. . . . .	261
Return codes . . . . .	261
Exit 14: SWAP exit. . . . .	261
Exit parameters. . . . .	261
Return codes . . . . .	261
Exit 15: DISPLAY service exit . . . . .	261
Exit parameters. . . . .	262
Return codes . . . . .	263
Exit 16: log, list, trace, and temporary data set allocation exit . . . . .	263
Exit parameters. . . . .	263
Return codes . . . . .	264
<b>Chapter 11. PDF translation tables . . . . .</b>	<b>265</b>
Translation table for valid data set name characters . . . . .	265
Translation table for invalid data set name characters . . . . .	265
Translation table for hexadecimal characters . . . . .	265

Translation table for numeric characters . . . . .	266
Translation table for alphanumeric characters. . . . .	266
Translation table for edit terminal output characters . . . . .	267
Translation table for generic string characters. . . . .	267
Translation table for generic string special characters . . . . .	267
Usage notes for the GSC and GSS tables . . . . .	268
Translation table for uppercase characters . . . . .	268
Translation table for lowercase characters . . . . .	269
Modifying the GSM to use the user character subset . . . . .	269
Generic string master translation table . . . . .	269
ISPF and PDF terminal translation table relationship . . . . .	271

**Chapter 12. Dialog development  
model listings . . . . . 273**

DM and PDF services in CLIST commands . . . . .	273
DM and PDF services in COBOL programs . . . . .	275
DM and PDF services in EXEC commands . . . . .	278
DM and PDF services in FORTRAN programs . . . . .	279
Message format . . . . .	282
Panel formats and statements . . . . .	282
DM and PDF services in PL/I programs . . . . .	285
File tailoring control statements . . . . .	287
DM and PDF services in Pascal programs . . . . .	287
DM and PDF services in TSO/REXX commands . . . . .	290
SCLM architecture definition formats . . . . .	292
SCLM project definition macros and templates . . . . .	293
ISPF Dialog Tag Language models . . . . .	293
DM and PDF services in C/370 programs . . . . .	293
Dialog Tag Language models . . . . .	296

**Chapter 13. Programming interface  
macros for customers . . . . . 299**

**Chapter 14. ISPF data set descriptions 301**

**Part 3. Appendixes . . . . . 303**

**Appendix A. ISPF enqueue processing  
for data integrity . . . . . 305**

Serializing with non-ISPF TSO and BATCH . . . . .	305
ISPF data set integrity enqueue . . . . .	305
Member name enqueue . . . . .	306
z/OS UNIX file Enqueue . . . . .	306
ISPF GUI TSO line mode message queue enqueue . . . . .	306
Workstation file name enqueue . . . . .	307
SCLM VSAM enqueue . . . . .	307

**Appendix B. Accessibility . . . . . 309**

Using assistive technologies . . . . .	309
Keyboard navigation of the user interface . . . . .	309
z/OS information . . . . .	309

**Notices . . . . . 311**

Programming Interface Information . . . . .	312
Trademarks . . . . .	312

**Index . . . . . 313**

---

## Preface

The Interactive System Productivity Facility (ISPF) product assists in program development. It is designed to take advantage of the characteristics of IBM display terminals, and to increase programmer productivity in an interactive environment.

---

### About this document

This document provides guidance and reference information for the planning and customizing of ISPF under z/OS with the Time Sharing Option Extensions (TSO/E). It consists of three parts:

**Part 1. Planning and customizing**

Contains detailed planning information and procedures you need to install and customize ISPF. It explains how to specify configuration options, modify the distributed menus and options, and tune ISPF performance.

**Part 2. Reference**

Provides a lookup reference for static information such as the ISPF Configuration Table keywords and values, PDF translation tables, dialog development model listings, and so on.

**Part 3. Appendixes**

This part provides information about ISPF enqueue processing for data integrity, accessibility, notices, and index.

---

### Who should use this document

*z/OS ISPF Planning and Customizing* is designed for system programmers or other people whose responsibilities include installing, customizing, and tuning ISPF. You should be familiar with MVS, ISPF concepts and terminology as described in the *z/OS ISPF Dialog Developer's Guide and Reference*, and with the System Modification Program Extended (SMP/E).

---

### What is in this document?

This publication contains:

- Chapter 1, “z/OS V1R11.0 ISPF planning,” on page 3 describes planning procedures you need to know before installing ISPF.
- Chapter 2, “The ISPF Configuration Table,” on page 9 describes the ISPF Configuration Utility.
- Chapter 3, “Customizing ISPF,” on page 39 describes how to modify items that affect both the DM and PDF components to suit the needs of your installation.
- Chapter 4, “Improving ISPF performance,” on page 67 describes how to tune ISPF to reduce system load and improve response times for interactive users.
- Chapter 5, “Customizing DM,” on page 79 describes how to modify items that affect the DM function of ISPF to suit the needs of your installation.
- Chapter 6, “ISPF installation-wide exits,” on page 113 describes how you can write exit routines to collect system-related information.
- Chapter 7, “Customizing PDF,” on page 121 describes how to modify the distributed release of PDF to suit the needs of your installation.

- Chapter 8, “PDF installation-wide exits,” on page 181 describes the installation-wide exits that allow you to customize the behavior of ISPF utilities such as Edit, Browse, Print, and Compress, as well as data set lists and member lists.
- Chapter 9, “ISPF Configuration Table keywords and values,” on page 203 lists the keywords in the ISPF Configuration Table and their allowable and default values.
- Chapter 10, “Exits,” on page 249 lists ISPF the installation-wide exits.
- Chapter 11, “PDF translation tables,” on page 265 lists the PDF translation tables.
- Chapter 12, “Dialog development model listings,” on page 273 lists all of the edit models included in the ISPF SKELS library.
- Chapter 13, “Programming interface macros for customers,” on page 299 lists the macros provided by ISPF as programming interfaces.
- Chapter 14, “ISPF data set descriptions,” on page 301 lists the target and distribution data sets used by ISPF, and their contents.
- Appendix A, “ISPF enqueue processing for data integrity,” on page 305 explains how ISPF ensures data integrity.

---

## The z/OS Basic Skills Information Center

The z/OS Basic Skills Information Center is a Web-based information resource intended to help users learn the basic concepts of z/OS, the operating system that runs most of the IBM mainframe computers in use today. The Information Center is designed to introduce a new generation of Information Technology professionals to basic concepts and help them prepare for a career as a z/OS professional, such as a z/OS system programmer.

Specifically, the z/OS Basic Skills Information Center is intended to achieve the following objectives:

- Provide basic education and information about z/OS without charge
- Shorten the time it takes for people to become productive on the mainframe
- Make it easier for new people to learn z/OS.

To access the z/OS Basic Skills Information Center, open your Web browser to the following Web site, which is available to all users (no login required):  
<http://publib.boulder.ibm.com/infocenter/zos/basics/index.jsp>

---

## How to read the syntax diagrams

The syntactical structure of commands described in this document is shown by means of syntax diagrams.

Figure 1 on page ix shows a sample syntax diagram that includes the various notations used to indicate such things as whether:

- An item is a keyword or a variable.
- An item is required or optional.
- A choice is available.
- A default applies if you do not specify a value.
- You can repeat an item.

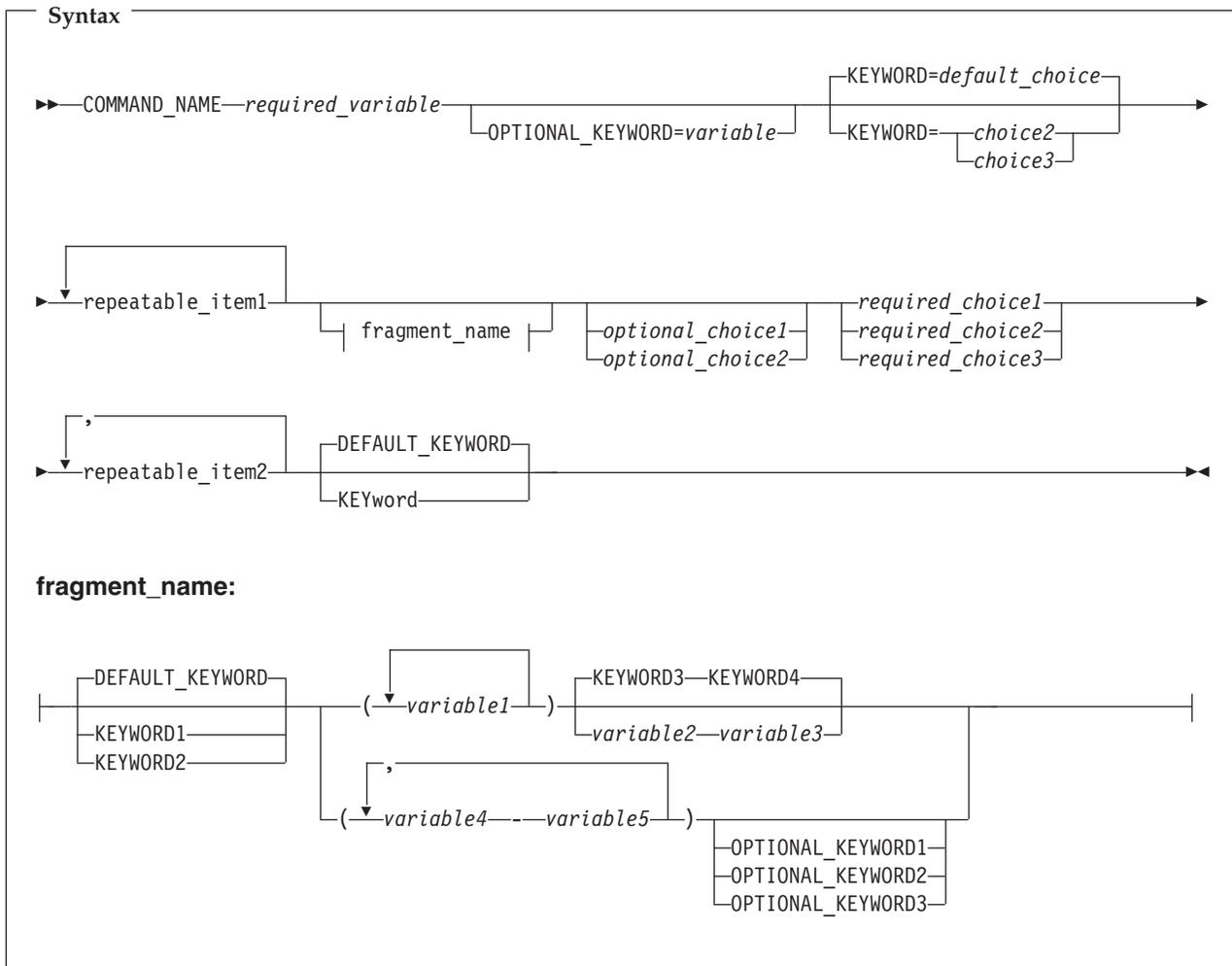


Figure 1. Sample syntax diagram

Here are some tips for reading and understanding syntax diagrams:

**Order of reading**

Read the syntax diagrams from left to right, from top to bottom, following the path of the line.

The **▶▶** symbol indicates the beginning of a statement.

The **▶▶** symbol indicates that a statement is continued on the next line.

The **▶▶** symbol indicates that a statement is continued from the previous line.

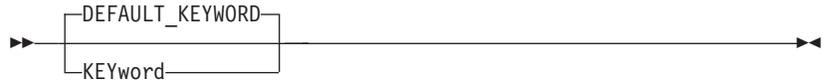
The **▶▶▶▶** symbol indicates the end of a statement.

**Keywords**

Keywords appear in uppercase letters.



Sometimes you only need to type the first few letters of a keyword, The required part of the keyword appears in uppercase letters.



In this example, you could type "KEY", "KEYW", "KEYWO", "KEYWOR" or "KEYWORD".

The abbreviated or whole keyword you enter must be spelled exactly as shown.

**Variables**

Variables appear in lowercase letters. They represent user-supplied names or values.



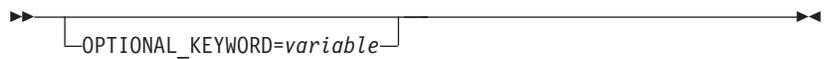
**Required items**

Required items appear on the horizontal line (the main path).



**Optional items**

Optional items appear below the main path.



**Choice of items**

If you can choose from two or more items, they appear vertically, in a stack.

If you *must* choose one of the items, one item of the stack appears on the main path.



If choosing one of the items is optional, the entire stack appears below the main path.



If a default value applies when you do not choose any of the items, the default value appears above the main path.



## Repeatable items

An arrow returning to the left above the main line indicates an item that can be repeated.



If you need to specify a separator character (such as a comma) between repeatable items, the line with the arrow returning to the left shows the separator character you must specify.



## Fragments

Where it makes the syntax diagram easier to read, a section or *fragment* of the syntax is sometimes shown separately.



⋮

**fragment\_name:**





---

## Summary of changes

This summary lists changes and enhancements for z/OS V1R11.0 ISPF. It has two parts:

### Product function changes

Describes the functional changes to z/OS V1R11.0 ISPF, listed by each ISPF component. This part appears in most of the ISPF documents.

### Changes to this document

Lists the changes and enhancements for z/OS V1R11.0 ISPF which affect this document, including cross-references to the new or changed sections.

---

## Product function changes for z/OS V1R11.0 ISPF

z/OS V1R11.0 ISPF contains the following changes and enhancements:

- ISPF product changes
- ISPF Dialog Manager component changes (including DTL changes)
- ISPF PDF component changes
- ISPF SCLM component changes
- ISPF Client/Server component changes

For details of migration actions relating to ISPF and other z/OS elements, see *z/OS Migration*.

## ISPF product changes

- Changes to the ZENVIR variable. Characters 1 through 8 contain the product name and sequence number in the format ISPF *x.y*, where *x.y* indicates the version number and release. Note that the *x.y* value is not the same as the operating system version. For example, a value of "ISPF 6.1" represents ISPF for z/OS Version 1 Release 11.0.
- The ZOS390RL variable contains the level of the z/OS release running on your system.
- The ZISPFOS system variable contains the level of ISPF that is running as part of the operating system release on your system. This might or might not match ZOS390RL. For this release of ISPF, the variable contains ISPF for z/OS 01.11.00.

## ISPF Dialog Manager component changes

The DM component of ISPF includes these new functions and enhancements:

- The new )INEXIT panel processing statement allows a panel source statement input exit to be specified. The input exit can perform dynamic modification of the ISPF panel. The exit is passed each panel source record as it is read by ISPF and is able to change, insert, or delete panel source records.
- Support for extended member statistics. This allows ISPF to store number of lines values greater than 65,535.
- The File Tailoring )DOT control statement supports the OPT parameter. If the OPT parameter is used and the table specified with the )DOT control statement does not exist ISPF will behave as though the table is empty rather than terminate file tailoring processing with message ISPF122.
- ISPF Configuration Utility changes:

- New keyword USE\_ADDITIONAL\_QUAL\_FOR\_PDF\_DATA\_SETS is used to specify an additional qualifier included in the default data set name for data sets generated by PDF utilities.
- Dialog Tag Language (DTL) changes:
  - There are no changes to Dialog Tag Language (DTL) for this release.

## ISPF PDF component changes

The ISPF PDF component contains these new functions and enhancements:

- Support for the display of Extended Address Volumes (EAV) data set level attribute EATTR.
- The ISPF Editor COMPARE command is enhanced to allow the name of a data set containing SuperC process statements and allocated to the SYSIN DD to be specified as part of the command syntax.
- New ISPF Editor line commands HX and HXX are provided to allow the display of individual records in hexadecimal format.
- New option Prefix Dsname Level is added to the Data Set list Utility entry panel. When this option is selected the Dsname Level can be specified with or without quotes. When the quotes are omitted the TSO prefix (if running with PREFIX ON) will be prefixed as the first qualifier of the Dsname Level. When the option is not selected the behavior of the entry panel remains unchanged.
- These enhancements to the z/OS UNIX Directory List Utility are provided:
  - New line commands are available to display and update these attributes for files and directories:
    - Owning user
    - Owning group
    - File format and tag information
    - User auditing options
    - Auditor auditing options
  - A new line command is available to display information about the file system for a file or directory.
  - z/OS UNIX commands can be entered from the primary command field on the directory list display panel.
  - Options are provided to allow a user to define default line commands for different file types. The default line command is invoked when a user puts the cursor in the line command field and presses enter without supplying a line command.

## ISPF SCLM component changes

The ISPF SCLM component contains these new functions and enhancements:

- The Search utility is provided to allow users to search a combination of SCLM groups, types and members for one or more strings.
- The ability to encode members stored in SCLM is available. These members are stored in a compressed format and are not viewable from outside SCLM.

## ISPF Client/Server component changes

There are no new functions or enhancements for the Client/Server component of ISPF.

## Migration considerations

There are no migration considerations relating to this release.

---

## Changes to this document for z/OS V1R11.0 ISPF

### Note

This book contains terminology, maintenance, and editorial changes. Technical changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

### Configuration option to add qualifier to default PDF output data set names

A new configuration option allows you to add a qualifier to the default names of data sets created by PDF utilities.

For changes to this document relating to the new configuration option, see:

- “ISPF site-wide profile customizations” on page 217

### PDS extended statistics

ISPF can now be configured to record PDS member statistics exceeding 32756 lines.

For changes to this document relating to PDS extended statistics, see:

- “Edit-related settings” on page 209
- Chapter 13, “Programming interface macros for customers,” on page 299

### Print utility exit

When the print utility exit is used, a job card will not be inserted in the ZTEMPF file prior to invoking the exit.

There are no documentation changes involved for this modification.

However, for details about the print utility exit, see:

- “Print utility exit” on page 192

### Document format

Since the last edition, the format of this document has been modified. The document is now divided into three *parts*:

- Part 1. Planning and customizing
- Part 2. Reference
- Part 3. Appendixes

For more information, see:

- “About this document” on page vii



---

## What's in the z/OS V1R11.0 ISPF library?

You can order the ISPF books using the numbers provided below.

<b>Title</b>	<b>Order Number</b>
<i>z/OS ISPF Dialog Developer's Guide and Reference</i>	SC34-4821-08
<i>z/OS ISPF Dialog Tag Language Guide and Reference</i>	SC34-4824-08
<i>z/OS ISPF Edit and Edit Macros</i>	SC34-4820-09
<i>z/OS ISPF Messages and Codes</i>	SC34-4815-09
<i>z/OS ISPF Planning and Customizing</i>	GC34-4814-08
<i>z/OS ISPF Reference Summary</i>	SC34-4816-09
<i>z/OS ISPF Software Configuration and Library Manager Guide and Reference</i>	SC34-4817-10
<i>z/OS ISPF Services Guide</i>	SC34-4819-09
<i>z/OS ISPF User's Guide Vol I</i>	SC34-4822-09
<i>z/OS ISPF User's Guide Vol II</i>	SC34-4823-09



---

## Part 1. Planning and customizing

<b>Chapter 1. z/OS V1R11.0 ISPF planning</b> . . . . .	3
Hardware and software requirements . . . . .	3
Notes on migrating from previous releases to z/OS V1R11.0 ISPF . . . . .	3
Installation, maintenance, and migration diagnostics . . . . .	4
Language Environment considerations. . . . .	5
Software Configuration and Library Manager (SCLM) . . . . .	5
Maintenance considerations . . . . .	6
Load module search order. . . . .	6
TSO logon procedure . . . . .	7
<b>Chapter 2. The ISPF Configuration Table</b> . . . . .	9
Using the ISPF Configuration utility before installation is complete . . . . .	10
System requirements . . . . .	10
Running the Configuration utility with a special logon procedure. . . . .	10
Running the Configuration utility with a REXX Exec . . . . .	11
The ISPF Configuration utility . . . . .	12
Create/Modify Settings and Regenerate Keyword File . . . . .	14
General ISPF Settings Panels. . . . .	15
System Profile (ISPSPROF) Settings panels . . . . .	25
Edit Keyword File Configuration Table . . . . .	31
Verify Keyword Table Contents. . . . .	33
Build Configuration Table Load Module. . . . .	33
Convert Assembler Configuration Table to Keyword File. . . . .	35
Build SMP/E USERMOD. . . . .	35
Installing the USERMOD on your system . . . . .	37
<b>Chapter 3. Customizing ISPF</b> . . . . .	39
Allocating optional image ISPF library . . . . .	39
Customizing action bars . . . . .	39
Invoking the ISPDTLC conversion utility . . . . .	40
Running interactively . . . . .	40
Running in batch mode . . . . .	44
Customizing for profile sharing. . . . .	45
Customizing the ISPF Configuration utility. . . . .	45
Temporary data set names . . . . .	46
SYSIKJUA enqueue. . . . .	46
Limitation of other products. . . . .	46
TSO/ISPF client gateway. . . . .	46
Gateway description . . . . .	46
Installing and customizing the gateway . . . . .	47
Installation considerations . . . . .	47
Installing the gateway . . . . .	48
Customizing the gateway. . . . .	49
Customizing the HTTP server . . . . .	49
Customizing Remote Systems Explorer . . . . .	50
Configuration for other environments . . . . .	50
Using the gateway . . . . .	51
Passing requests to the gateway . . . . .	51
Receiving output from the gateway . . . . .	54
Return and reason codes from the gateway. . . . .	59
Using the native API . . . . .	60
<b>Chapter 4. Improving ISPF performance</b> . . . . .	67
Use virtual I/O . . . . .	67
Remove or tailor Edit functions. . . . .	67
Change the block size of the edit recovery data set . . . . .	67
Disable SETUNDO STORAGE . . . . .	68
Disable Edit extended highlighting . . . . .	68
Remove action bars from ISPF panels. . . . .	68
Remove scrollable areas from ISPF panels . . . . .	69
Preprocess all ISPF panels . . . . .	69
Preprocessed panel utility . . . . .	69
Add load modules to the pageable link-pack area (LPA) . . . . .	71
Moving load modules . . . . .	71
ISPF load module descriptions . . . . .	72
Allocate execution data sets . . . . .	73
Disable generic high-level qualifiers . . . . .	74
Preallocate ISPF temporary data sets to VIO . . . . .	76
Compress SCLM listings . . . . .	78
<b>Chapter 5. Customizing DM</b> . . . . .	79
Changing the default name for the ISPF Client/Server component download data set . . . . .	79
Set up ISPF GUI for TSO Line Mode support . . . . .	79
SMF command accounting . . . . .	80
Preallocation of List/Log data sets to SYSOUT . . . . .	83
Specifying the maximum number of split screens. . . . .	83
Setting ISPF site-wide defaults . . . . .	84
Customizing command tables . . . . .	84
Application command table . . . . .	85
User command tables . . . . .	85
Site command tables . . . . .	85
System command table . . . . .	86
Creating ISPF terminal translation tables. . . . .	86
Uppercase character translation table . . . . .	87
Lowercase character translation table . . . . .	87
Valid terminal output translation table . . . . .	87
Generic string master translation table . . . . .	88
Modifying the GSM to use the user character subset . . . . .	88
Alphabetic character translation tables . . . . .	89
Collating sequence translation table . . . . .	89
Specifying terminal types. . . . .	90
Changing the DTL source for ISPISMMN . . . . .	91
Changing the DTL source for ISPOPTxx panels . . . . .	93
Invoking ISPTTDEF . . . . .	94
Creating ISPF code page translation tables . . . . .	95
Base code pages for terminals . . . . .	97
ISPCCSID translation load modules . . . . .	98
Adding translation tables for extended code page support. . . . .	98

Extended code page translation tables provided by ISPF . . . . .	99
ISPCCSID translation load module generation macro . . . . .	100
ISPCCSID translation load module definition examples . . . . .	101
Example of user-modifiable ISPF translation table . . . . .	101
Displaying square brackets used in C programs	104
ISPEXEC processing . . . . .	104
ISPF-to-APL2 terminal type mappings . . . . .	105
Load APL2 workspace . . . . .	105
Tailoring ISPF defaults . . . . .	106
Customizing the ISPF TSO command table (ISPTCM). . . . .	106
Sample ISPTCM definition . . . . .	109
ISPTCM usage notes . . . . .	109
Search sequence for attaching commands . . . . .	110
Alternate option 7.1 panels . . . . .	110
ISPF multicultural support . . . . .	111
Changing the session language default value	111
<b>Chapter 6. ISPF installation-wide exits . . . . .</b>	<b>113</b>
How to install the installation-wide exits . . . . .	114
How to use the macros to define ISPXDT . . . . .	115
Sample ISPXDT definition . . . . .	118
Exit parameter list . . . . .	118
Error processing . . . . .	119
<b>Chapter 7. Customizing PDF . . . . .</b>	<b>121</b>
Edit mode defaults . . . . .	121
Site-wide edit profile initialization . . . . .	122
Creating a ZDEFAULT profile . . . . .	122
Action bars and extended color in Edit . . . . .	123
Edit backup and recovery . . . . .	123
Data set allocation defaults for the Outlist utility	124
Using the Hardcopy utility with DBCS support . . . . .	125
SCLM Batch considerations. . . . .	125
Converting from LMF To SCLM . . . . .	127
General conversion . . . . .	127
Converting LMF dialogs to SCLM . . . . .	129
Creating PDF translation tables . . . . .	129
PDF Foreground and Batch customizing . . . . .	130
Foreground processing panels and CLISTs. . . . .	130
Required option entry panel variables . . . . .	141
Other option entry panel variables . . . . .	142
Variables that control automatic Browse and Print . . . . .	142
Variables that control option panel redisplay	143
Variables used by Foreground CLISTs . . . . .	143
Foreground modifications . . . . .	143
Batch processing panels, CLISTs, and skeletons	145
Variables that control batch processing . . . . .	159
Generated JCL . . . . .	160
Variables that control option panel redisplay	160
Variable used by Batch CLISTs . . . . .	161
Batch skeletons . . . . .	161
Batch modifications . . . . .	161
Batch processing options considerations . . . . .	163
ISRSCAN and ISRLEMX programs . . . . .	163
Adding user-defined triggers . . . . .	166

Customizing Browse and Edit . . . . .	167
Providing customized Browse and Edit panels	168
Enabling Browse panels to display Unicode data	172
Customizing member list panels . . . . .	172
Customizing the z/OS UNIX directory list panel	175
IBM Products option . . . . .	175

---

## Chapter 1. z/OS V1R11.0 ISPF planning

Application dialogs that were created to run under earlier versions of ISPF will run under z/OS V1R11.0 ISPF with no change.

All the components of ISPF (DM, PDF, SCLM, and the Client/Server) are considered one element in all releases of z/OS. Attempting to run one of the components from one release of z/OS with a component from a different release of z/OS or with any component of a standalone release of ISPF or ISPF/PDF is not supported.

---

### Hardware and software requirements

For hardware and software requirements for ISPF, refer to *z/OS Planning for Installation*, number GA22-7504.

---

### Notes on migrating from previous releases to z/OS V1R11.0 ISPF

For information about migrating from a previous supported release of ISPF, refer to *z/OS Migration*.

It is also possible to migrate to z/OS V1R11.0 ISPF from earlier versions of ISPF.

Customization of ISPF had major changes in OS/390® Version 2 Release 8.0. When migrating from a version of OS/390 earlier than V2R8.0 you must convert your ISPF customization to the new format. See Chapter 2, “The ISPF Configuration Table,” on page 9 for more information.

Major changes were made to the ISPF profiles for OS/390 V1R1.0. The profiles for earlier versions of ISPF are not compatible with z/OS ISPF.

**IPV6** ISPF C/S component does not support TCPIP connections in an IPV6 environment. ZIPADD6 will be formatted correctly however.

Use this information to help you during the transition stage.

- The z/OS V1R11.0 ISPF data sets are all named ISP.SISPxxxx and ISP.AISPxxxx. Version 4 ISPF data sets were named ISP.V4RxM0.xxxxx.
- Using SMP to install z/OS V1R11.0 ISPF deletes these licensed programs:
  - ISPF Version 3.n and ISPF/PDF Version 3.n
  - ISPF Version 4.n
  - OS/390 VnRn.0 ISPF
  - z/OS V1R2.0 ISPF
  - z/OS V1R5.0 ISPF
  - z/OS V1R6.0 ISPF
  - z/OS V1R7.0 ISPF
  - z/OS V1R8.0 ISPF

During the transition stage, maintain the previous programs on a separate set of back-up packs or in a set of libraries that does not interfere with the execution of z/OS V1R11.0 ISPF.

- During the transition stage, it is assumed that the predecessor programs reside in the system pageable link pack area. ISPF and the new programs are assumed to be running from a set of execution libraries, as described in the *z/OS Program Directory*.
- Many of the modules in ISPF have the same names as modules in the previous licensed programs. Therefore, the ISPF module libraries should not be in your system LNKLSTxx definition during the transition period.
- During the initial testing period, ISPF should be run from the SISpload and SISPLPA data sets created during the SMP APPLY processing step. Use a STEPLIB DD statement in the TSO LOGON procedure to accomplish this.  
If you are using the ISPF client/server feature or the z/OS UNIX Directory List utility, include the Language Environment run-time library data sets SCEERUN and SCEERUN2 in the STEPLIB if they are not already in the LNKLIST.  
If you are also running with an ISPLLIB DD statement, you must include the z/OS V1R11.0 ISPF SISpload and SISPLPA data sets in the ISPLLIB concatenation and remove any load library data sets from a previous version or release of ISPF.
- If only the top modules of z/OS V1R11.0 ISPF (ISPICP and alias ISPSTART as well as ISRPCP and alias entries PDF and ISPF) are in the step libraries, and the other load modules for the new and previous ISPF versions are in libraries allocated to the ISPLLIB ddname, both the old programs and the new programs can be executed concurrently under the same LOGON procedure. Concurrent execution assumes that a CLIST is used to provide the allocations of the other required ISPF libraries.
- *Do not* use the command names ISPF, PDF, or ISPSTART during the transition stage. Conflicts can occur if you invoke modules from earlier versions of ISPF with the same names. One solution is to rename the alias entries of the top ISPF modules in the step library to temporary names (for example, from ISPF to NEWISPF) until the transition stage is complete.
- After the transition stage, the z/OS V1R11.0 ISPF LPA-eligible load modules (data set SISPLPA) should be put in LPA, and the LPA-ineligible load modules (data set SISpload) should be put in LNKLST. See Chapter 4, “Improving ISPF performance,” on page 67 for more information. At that time, remove any DD statements for a STEPLIB or ISPLLIB definition of the ISpload/ISPLPA data set from the LOGON procedure.
- Panel ISRLOGO was renamed to ISR@LOGO at z/OS Version 1 Release 1.0. If you have a modified version of ISR@PRIM, check that the correct panel value is used in the HELP Action Bar pull-down menu.

## Installation, maintenance, and migration diagnostics

This information describes common situations you may encounter during installation, maintenance, and migration.

- |                  |  |
|------------------|--|
| <b>Problem:</b>  | Unpredictable results such as 0C4 or 0C1 abends upon invocation of ISPF.   |
| <b>Cause:</b>    | This problem can occur when internal LINKs and LOADs are used in conjunction with the ISPF ISPLLIB DCB.  |
| <b>Solution:</b> | When using STEPLIB to test new maintenance, releases, or versions of products and an ISPLLIB is allocated, data sets allocated to STEPLIB should also be allocated to ISPLLIB. |
| <b>Problem:</b>  | New levels of code residing in STEPLIB do not appear to be   |

executed. This includes changes to customer applications, new levels of code, and changes made by PTFs, for example.

- Cause:** This problem can occur when internal LINKS and LOADs are used in conjunction with the ISPF ISPLLIB DCB.
- Solution:** When using STEPLIB to test new maintenance, releases, or versions of products and an ISPLLIB is allocated, data sets allocated to STEPLIB should also be allocated to ISPLLIB.
- Problem:** Abends occur when invoking or exiting HELP, KEYS, or ISPPREP.
- Cause:** An older level of ISPF exists in LPA or LINKLIB; an ISPLLIB or STEPLIB is used to allocate libraries for the new release of ISPF; and a LIBDEF for ISPLLIB that does not include the new level of ISPF libraries has been issued.
- Not all of ISPF is loaded at initialization, for example, ISPTUTOR is linked when HELP is requested. The LIBDEF of ISPLLIB results in the older level of any modules that are not loaded on ISPF entry to be picked up from LPA or LINKLIB due to internal search orders. For more information, refer to the LIBDEF command in *z/OS ISPF Services Guide*.
- Solution:** Include the new level of ISPF in any LIBDEFs issued for ISPLLIB.
- Problem:** Abend code 806 for ISPLINK or Abend code 0C1 in user application.
- Cause:** During installation, the SMP/E install logic for ISPF deletes the existing ISPLINK load module.
- Solution:** Relink-edit ISPLINK to user application.
- Problem:** Abends in load modules IGC0009C and IGC0009D.
- Cause:** Failure to install ISPF in the same zone as TSO/E will result in the supplied ISPF SVC 93 and SVC 94 exits (ISPSC93, ISPSC94), creating SVC 93 and SVC 94 load modules without the proper link-edit information.
- Failure to use the correct version of either IGC0009C or IGC0009D or both.
- Solution:** TSO/E, ISPF, and ISPF/PDF must all be installed in the same zone. Verify that the correct versions are copied from a test system to the production system.

---

## Language Environment considerations

The ISPF client/server (ISPF C/S) feature and the z/OS UNIX Directory List utility include code are compiled with the IBM C/C++ compiler. Therefore, the Language Environment SCEERUN and SCEERUN2 run-time library data sets must be available to this code through either STEPLIB or LNKLIST at execution time. Using the usual ISPLLIB allocation will not make the run-time library available to this code.

---

## Software Configuration and Library Manager (SCLM)

Use the Software Configuration and Library Manager (SCLM) to create, control, maintain, and track software components for a project.

The SCLM project database consists of a series of related ISPF libraries (partitioned data sets). These contain source and non-source software components. SCLM project definition and control information is contained in an assembled and linked PROJDEFS data set. SCLM project cross-reference and accounting data sets are VSAM clusters.

SCLM requires RACF® (or equivalent) to protect the SCLM-controlled data sets. Individual developers should have READ authority to all levels of the hierarchy to draw down members from the hierarchy using the SCLM edit option. Those developers who promote the modified members up the hierarchy will need UPDATE authority for the libraries into which they are promoting. The Build Coordinator should have UPDATE access to the libraries containing the non-editable parts (non-editable parts are the output from the build process) and READ access to all levels of the hierarchy to allow the builds to be done.

For information about converting from LMF to SCLM see “Converting from LMF To SCLM” on page 127. For detailed information about SCLM, refer to *z/OS ISPF Software Configuration and Library Manager Guide and Reference*.

---

## Maintenance considerations

ISPF is packaged using the features of SMP/E that allow multiple multicultural support language features to be installed in a single target zone. We recommend that you take advantage of this by installing all of the base code and all multicultural support language features in one target zone and one distribution zone. This simplifies the installation of maintenance.

If you do not install your multicultural support features in the same target and distribution zones as the base, SMP/E cannot properly track maintenance (PTFs) that contains REQ or IF REQ statements. When you install PTFs for the multicultural support features that contain REQ statements for PTFs that apply to the base or PTFs for the base that contain IF REQ statements for PTFs that apply to the multicultural support features, you must use this procedure:

1. APPLY the base PTF that is REQed by the multicultural support feature PTF or contains an IF REQ for an multicultural support feature PTF.
2. APPLY the multicultural support feature PTF using BYPASS(REQ).
3. Test the maintenance according to your normal procedures.
4. ACCEPT the base PTF that is REQed by the multicultural support feature PTF or contains and IF REQ for a multicultural support feature PTF.
5. ACCEPT the multicultural support feature PTF using BYPASS(REQ).

PTFs for ISPF do not contain preprocessed panels. If you are using preprocessed panels in your execution data sets, after installing PTFs that contain panels you must preprocess the panels in the PTF into your execution data set to fully install the maintenance.

---

## Load module search order

When using STEPLIB to test either new maintenance, releases, or versions of ISPF, and an ISPLLIB is allocated, those data sets allocated to STEPLIB that contain ISPF load modules should also be allocated to ISPLLIB. This prevents the possibility of mixed code (production code versus code to be tested).

The exceptions to this search order are the Language Environment run-time library data sets SCEERUN and SCEERUN2. Modules in these data sets are not searched

for using the ISPLLIB task library. SCEERUN and SCEERUN2 must be in STEPLIB or LNKLST if you are using the ISPF C/S feature or the z/OS UNIX Directory List utility. See “Language Environment considerations” on page 5 for more information about SCEERUN and SCEERUN2.

For more information about search order, refer to *z/OS ISPF User's Guide Vol I*.

## TSO logon procedure

Figure 2 is a sample TSO logon procedure (logon proc) to allocate the ISPF data sets required to execute ISPF.

```
//ISPF      PROC
//ISPF      EXEC PGM=IKJEFT01,DYNAMBR=100
//*
//STEPLIB  DD DSN=ISP.SISPLPA,DISP=SHR
//          DD DSN=ISP.SISPLoad,DISP=SHR
//          DD DSN=CEE.SCEERUN,DISP=SHR
//          DD DSN=CEE.SCEERUN2,DISP=SHR
//*
//ISPLIB   DD DSN=ISP.SISPMxxx,DISP=SHR
//          DD DSN=SYS1.SBPXMENU,DISP=SHR
//*
//*SYSHelp DD DSN=ISP.SISPHELP,DISP=SHR
//*
//ISPLLIB  DD DSN=ISP.SISPPxxx,DISP=SHR
//          DD DSN=SYS1.SBPXPENU,DISP=SHR
//*
//ISPSLIB  DD DSN=ISP.SISPSxxx,DISP=SHR
//          DD DSN=ISP.SISPSLIB,DISP=SHR
//*
//ISPTLIB  DD DSN=userid.ISPTABLE,DISP=SHR
//          DD DSN=ISP.SISPTxxx,DISP=SHR
//          DD DSN=SYS1.SBPXTENU,DISP=SHR
//*
//ISPIlib  DD DSN=ISP.SISPSAMP,DISP=SHR
//*
//SYSPROC  DD DSN=ISP.SISPCLIB,DISP=SHR
//          DD DSN=SYS1.SBPXEXEC,DISP=SHR
//*
//SYSEXEC  DD DSN=ISP.SISPEXEC,DISP=SHR
//*
//ISPTABL  DD DSN=userid.ISPTABLE,DISP=SHR
//*
//ISPPROF  DD DSN=userid.ISPTABLE,DISP=OLD
//*
```

Figure 2. TSO logon procedure

The default names `ISP.SISPyxxx` are used for the ISPF data sets, where `xxx` represents the national language as follows:

Language	xxx
US English	ENU
Japanese	JPN
German	DEU
Swiss German	DES
Uppercase English	ENP

**Note:** For information about data sets that are required by other products for them to run under ISPF, see the *z/OS Program Directory* and the documentation for the specific product.

This logon procedure assumes that all data sets are cataloged and that SYSEXEC is included in the search order for REXX execs on your system. See *z/OS TSO/E REXX User's Guide* and *z/OS TSO/E Command Reference* for details about controlling the search order for REXX execs.

If you are using TSO line mode support, your logon procedure must invoke the alternate entry point **PGM=IKJEFT11**. The TSO Line mode support cannot be used with the ISPF load modules in STEPLIB. Many of the ISPF load modules must be in the pageable link-pack area. See "Set up ISPF GUI for TSO Line Mode support" on page 79 for more information.

This logon procedure includes the optional ISPILIB allocation. ISPILIB is used for images in Graphic Image Format (GIF) that are used by ISPF when running in GUI mode. For more information about using ISPF's image support, refer to the *z/OS ISPF User's Guide Vol 1*.

If an ISPLLIB is allocated when you use a STEPLIB to test new maintenance, releases, or versions of ISPF, you must include the data sets allocated to STEPLIB that contain ISPF load modules in the ISPLLIB allocation. This prevents mixing production code with code to be tested. For more information about library search order, refer to the *z/OS ISPF Services Guide*.

The SCEERUN and SCEERUN2 libraries are required in STEPLIB or LNKLST if you plan to use either the ISPF Client/Server feature or the z/OS UNIX Directory List utility. SCEERUN and SCEERUN2 are the Language Environment run-time library data sets and contain run-time modules required by C/C++ code used by these ISPF features. These libraries must be in STEPLIB or LNKLST for the modules to be loaded at execution time. These modules are not searched for using the ISPLLIB allocation. See "Language Environment considerations" on page 5 for more information.

If you plan to use the z/OS UNIX Directory List utility, these libraries must be allocated as shown in Figure 2 on page 7: SYS1.SBPXMENU, SYS1.SBPXPENU, SYS1.SBPTMENU, SYS1.SBPXEXEC.

You must use a unique user profile data set (userid.ISPTABLE) in the sample logon procedure for each national language. If you do not, the result is mixed language messages and prompts. The data set *userid.ISPTABLE* must have record format FB, LRECL 80, and a block size that is a multiple of 80. You might want to allocate the ISPPROF, ISPTABLE, and ISPTLIB DDs in a REXX exec that is executed by the **PARM** on the EXEC statement of the logon procedure.

---

## Chapter 2. The ISPF Configuration Table

In earlier versions of ISPF, changing configuration defaults involved two separate entities: the ISPF Configuration table (ISRCONFIG) and the ISPF Defaults member (ISPDFLT). In OS/390 Version 2 Release 8.0 these two parts were combined into a single configuration table, which included keywords for all the existing ISRCONFIG and ISPDFLT values.

The ISPF Configuration table is a keyword-driven flat file. Each entry is in the format `KEYWORD = value`. A slash and asterisk (`/*`) in columns 1 and 2 indicate a comment line. Some advantages of this keyword approach are:

- Only one place to go for all configuration needs.
- No need to rename sample ISRCNFIG to ISRCONFIG to get the correctly named load module created.
- No need to modify ISPF standard parts.
- No need to modify any keywords or values EXCEPT those for which you DO NOT want the default value.
- Different keyword files, representing different configurations, can be saved in different members of a PDS. Each one can be generated into a load module for different sets of users.

Use the ISPF Configuration table to change site-wide defaults and to indicate that installation exit routines are provided for some of the ISPF functions. The ISPF functions that allow installation-written exit routines are data set allocation, print utility, data set compression, the data set list utility, member list filter, and data set name change. ISPF checks the configuration table to determine, first, if exit routines are provided, and second, whether those routines are programs or CLISTS. If you specify both a CLIST and a program, ISPF uses the program.

A default configuration load module (ISPCFIG) is included with ISPF. It is used if you choose to make no customizations for your installation. If you do want to create your own configuration table, simply use the ISPF Configuration utility to edit the flat file (name and data set of your choosing, but it must be a PDS) and convert it into a load module called ISPCFIGU. Optionally, you can use the same file to create load module called ISPCFIGV for VSAM support. The converted load module must be placed in the standard MVS search sequence or allocated to ISPLLIB for ISPF to use the values specified. For more information about the ISPF Configuration utility, see “The ISPF Configuration utility” on page 12.

It is recommended that you always use the latest release as the common configuration, though older releases will work with the newest table. Newer releases will also work with an older table, but the ability to configure the settings added in the latest release is lost.

---

## Using the ISPF Configuration utility before installation is complete

You might want to run the ISPF Configuration utility for a new release to set configuration options added by the new release before you actually IPL your new system for the first time. If you are installing with CBPDO you can run the utility after the SMP/E APPLY step of the ISPF FMIDs. If you are installing with ServerPac you can run the utility after the ISPF target libraries are restored to your system.

### System requirements

The utility must be run on a supported release of z/OS, and have High Level Assembler (HLASM) 1.3.0 or later, and a supported release of SMP/E. If your system does not have the required level of HLASM or SMP/E, you can use STEPLIB DD statements to access HLASM in the target system's SASMMOD1 data set, and SMP/E in the target system's SYS1.MIGLIB data set.

### Running the Configuration utility with a special logon procedure

To run the ISPF Configuration utility using a special TSO logon procedure, follow these steps.

1. Allocate the ISPF target libraries in the appropriate DDs in a TSO logon procedure as shown in Figure 3 on page 11. The ISPF libraries should be first in each concatenation except ISPTLIB where your profile data set should be first. The default ISPF names are given with xxx representing the national language identifier for your installed language. If the ISPF data sets for the new release are not cataloged on the system you are using to run the utility then you must include **UNIT** and **VOL=SER** parameters on the DD statements for those data sets.
2. Include **HLASM(SASMMOD1)** and **SMP/E(SYS1.MIGLIB)** in STEPLIB if they are not at the required level on your system. If the data sets are not cataloged on the system you are using to run the ISPF Configuration utility then you must include **UNIT** and **VOL=SER** parameters on the DD statements for those data sets.
3. Include your ISPF profile data set in ISPTLIB and ISPTABL as shown in Figure 3 on page 11.
4. Logon to TSO using your modified logon procedure.
5. Start ISPF with the **ISPF** or **PDF** command.
6. Enter command **TSO ISPCCONF** to start the ISPF Configuration utility.

```

//STEPLIB DD DISP=SHR,DSN=ISP.SISPLPA      ISPF LPA load library
//        DD DISP=SHR,DSN=ISP.SISPLOAD    ISPF LNKLST load library
//        DD DISP=SHR,DSN=ASM.SASMMOD1    High Level Assembler
//        DD DISP=SHR,DSN=SYS1.MIGLIB     SMP/E library
//*
//SYSEXEC DD DISP=SHR,DSN=ISP.SISPEXEC    ISPF Exec library
//*
//ISPMLIB DD DISP=SHR,DSN=ISP.SISPMxxx    ISPF Messages library
//*
//ISPPLIB DD DISP=SHR,DSN=ISP.SISPPxxx    ISPF Panels library
//*
//ISPSLIB DD DISP=SHR,DSN=ISP.SISPSxxx    ISPF Language Skels library
//        DD DISP=SHR,DSN=ISP.SISPSLIB    ISPF Base Skels library
//ISPTLIB DD DISP=SHR,DSN=your.profile    Your ISPF Profile
//        DD DISP=SHR,DSN=ISP.SISPTxxx    ISPF Tables library
//*
//ISPTABL DD DISP=SHR,DSN=your.profile    Your ISPF Profile
//*
//ISPPROF DD DISP=SHR,DSN=your.profile    Your ISPF Profile

```

Figure 3. ISPF Configuration utility logon procedure

### Running the Configuration utility with a REXX Exec

To run the ISPF Configuration utility using a REXX Exec, the ISPF data sets for the new release must be cataloged on the system you are using to run the utility. If the system you are using to run the utility does not have the required level of HLASM or SMP/E, you can do one of two things. You must either use STEPLIB DD statements in your logon procedure to access HLASM in the target system's SASMMOD1 data set, and SMPE in the target system's SYS1.MIGLIB data set, or catalog the HLASM and SMP/E data sets on the system you are using to run the utility.

To run the utility using a REXX Exec, follow these steps.

1. Copy member ISPFCU from the SISPSAMP library for the new release to a library that is in your normal SYSEXEC or SYSPROC concatenation. You can rename it if you want to, but avoid names that start with the characters "ISP".
2. Modify the data set names in the exec as indicated in the comments. Use fully qualified data set names.
3. If the required level of HLASM is in LPA or LNKLST, or in the STEPLIB of the TSO logon procedure that you will be using then set the value for **asm\_load** to null (''). If you want to access the required level of HLASM through this exec the data set must be cataloged on your running system.
4. If the required level of SMP/E is in LPA or LNKLST, or in the STEPLIB of the logon procedure that you will be using, then set the value for **gim\_load** to null (''). If you want to access the required level of SMP/E through this exec the data set must be cataloged on your running system.
5. The default names for the ISPF libraries are supplied. They must be cataloged on your running system. Change them as needed.
6. Logon to TSO.
7. Run the exec from TSO Ready. ISPF for the new release will start.
8. Enter the command **TSO ISPCCONF** to start the ISPF Configuration utility.

---

## The ISPF Configuration utility

The ISPF Configuration utility enables you to:

- modify the configuration settings saved in the keyword files
- generate a configuration table load module or SMP/E USERMOD for use on your ISPF system
- convert existing ISPF configuration table assembler files from earlier releases of ISPF (SAMPLIB member ISRCNFIG) into the keyword file format

The ISPF Configuration utility sets two types of values: system-wide values and user-specific values. System-wide values are values that are used for all users and are re-read from the Configuration Table at the beginning of each ISPF session. For example, the PDF Exits and PDF Default Unit are system-wide values.

User-specific values are used as initial values when a new profile is created. For example, if a user has an ISPSPROF or Edit Profile, it is used instead of the Configuration Table values. These profiles hold user-specific values.

As system administrator you can, if you choose, ensure that even user-specific values are set according to your specifications. Some of the user-specific Edit Profile fields have corresponding FORCE fields. The FORCE fields enable you to require the user to use the specified configuration values even if the user already has an edit profile. If the user attempts to modify one of the "forced" values, an error message is displayed. Some of the user-specific ISPSPROF fields have corresponding RESET fields. The RESET fields enable you to reset these values once for each user. The reset is done when the Sitewide Defaults Version Level is incremented. Any ISPSPROF fields that do not have RESET fields are not used by a user unless they are creating a new ISPSPROF table.

You can generate the keyword file in one of two ways:

- Use the **Convert Assembler Configuration Table to Keyword File** option on the ISPF Configuration utility main menu.
- Use the **Create/Modify Settings and Regenerate Keyword File** option on the ISPF Configuration utility main menu.

To start the ISPF Configuration utility run the command **TSO ISPCCONF** while in ISPF. This displays the ISPF Configuration utility main menu panel, as shown in Figure 4 on page 13.

```

                                ISPF Configuration Utility

1 Create/Modify Settings and Regenerate Keyword File
2 Edit Keyword File Configuration Table
3 Verify Keyword Table Contents
4 Build Configuration Table Load Module
5 Convert Assembler Configuration Table to Keyword File
6 Build SMP/E USERMOD

Keyword File Data Set
Data Set . . . _____
Member . . . . _____

Configuration Table Assembler Source Data Set
Data Set . . . _____
Member . . . . _____

Output File Content for Keyword File
2 1. Include only non-default values
   2. Include defaults as comments
   3. Include all values

Current Configuration Table
Keyword File : not available
Identifier . . : ISPCFIGU                      Level . . . . : 480R8001
Compile Date : 2001/11/22                      Compile Time :
Option ==> _____
F1=Help      F2=Split    F3=Exit      F7=Backward F8=Forward  F9=Swap
F12=Cancel

```

Figure 4. ISPF configuration utility main menu panel (ISPPCONF)

These options are available on the ISPF Configuration utility main menu:

1. Create/Modify Settings and Regenerate Keyword File (see page 14)
2. Edit Keyword File Configuration Table (see page 31)
3. Verify Keyword Table Contents (see page 33)
4. Build Configuration Table Load Module (see page 33)
5. Convert Assembler Configuration Table to Keyword File (see page 35)
6. Build SMP/E USERMOD (see page 35).

These input fields are available on the ISPF Configuration utility main menu panel:

**Keyword File Data Set**

The name of the data set and member containing the keyword file you want to use. The data set must exist and have a record length of at least 255 bytes for variable-length files, and 251 bytes for fixed-length files. This must be a Partitioned Data Set (PDS).

**Configuration Table Assembler Source Data Set**

The name of the sequential data set or partitioned data set and member containing the Configuration Table assembler source code to be converted into a keyword file.

**Output File Content for Keyword File**

The entry in this field determines the type of data written to the keyword file when the Convert Assembler Configuration Table to Keyword file option, or the Create/Modify Settings and Regenerate Keyword File option is used. Valid values are:

- 1 Only those values that are different than the default values are included in the keyword file.
- 2 Those values different than the default values are included in the keyword file, and all default values are included as comment lines.
- 3 All values are included in the keyword file.

## Create/Modify Settings and Regenerate Keyword File

If you choose Option 1, **Create/Modify Settings and Regenerate Keyword File** on the ISPF Configuration utility main menu panel, the Create/Modify ISPF Configuration panel appears, as shown in Figure 5.

This option enables you to create a new keyword file or modify an existing keyword file containing ISPF configuration settings. Each time you use this option the keyword file is regenerated with the new or modified settings values. Note that any comments that have been added, or any reformatting changes that you might have made to the keyword file by editing it directly *do not* appear in the regenerated file.

```

                                Create/Modify ISPF Configuration           Defaults loaded

General ISPF Settings                               System Profile (ISPSPROF) Settings
1 Editor Settings                                   6 Log and List Defaults
2 Edit/View/Browse VSAM Settings                   7 Terminal and User Defaults
3 PDF Exits and Other PDF Settings                 8 Workstation Defaults
4 ISPF Site-wide Defaults                           9 Workstation Download Defaults
5 ISPDFLTS, CUA Colors, and Other
  DM Settings

Output Keyword File
Data Set . . . CONFIG.KEYWORD.FILE
Member . . . . SYSTEM1

Instructions:
Enter option to change configuration settings,
END or EXIT command to generate keyword file, or
CANCEL command to exit without keyword file generation

Option ==>> _____
F1=Help      F2=Split    F3=Exit     F7=Backward F8=Forward  F9=Swap
F12=Cancel

```

Figure 5. Create/modify ISPF Configuration panel (ISPPMOD)

From this panel you can choose to modify these groups of configuration options by entering the option number on the command line.

- General ISPF Settings (see “General ISPF Settings Panels” on page 15)
  - 1 Editor Settings
  - 2 Edit/View/Browse VSAM Settings
  - 3 PDF Exits and Other PDF Settings
  - 4 ISPF Site-wide Defaults
  - 5 ISPDFLTS, CUA Colors, and Other DM Settings
- System Profile (ISPSPROF) Settings (see “System Profile (ISPSPROF) Settings panels” on page 25)
  - 6 Log and List Defaults

- 7 Terminal and User Defaults
- 8 Workstation Defaults
- 9 Workstation Download Defaults

As you choose a group of settings to modify, a panel specific to that group of settings is displayed. If you are creating a new keyword file, the panel fields are initialized with the default values for each setting. If you are modifying an existing keyword file, the panel reflects the current values in that keyword file.

Each of the selected panels has field-level help for the individual fields on the panel. To display a pop-up help panel for a field, put the cursor on the field and press the Help function key, or enter HELP on the command line.

The Output Keyword File entry fields enable you to save the generated keyword file to a data set or member other than the data set or member used as input. If you leave this field blank, the data is saved back to the input data set and member specified on the ISPF Configuration utility main menu panel (ISPPCONF).

### General ISPF Settings Panels

Selecting Option 1, **Editor Settings** on the Create/Modify ISPF Configuration panel (ISPPMOD) displays the Modify PDF Edit Configuration Settings panel (ISPPMOD1). Scroll down to display all the fields on this panel.

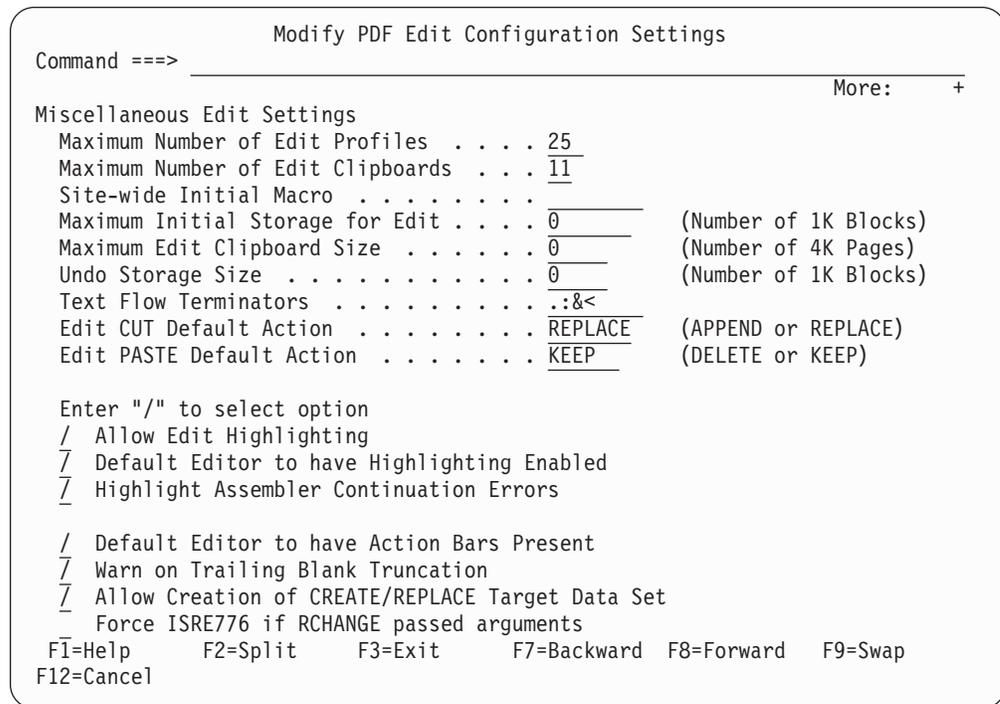


Figure 6. Modify PDF EDIT Configuration Settings panel (ISPPMOD1)

```

Modify PDF Edit Configuration Settings
Command ==>>> _____ More: - +

Edit Preserve Settings
Enter "/" to select option
  _ Preserve VB record length
  _ Force the Preserve VB record length Selection

Edit Recovery Data Set Settings
Block Size . . . . . 13680
Primary Blocks . . . . . 40
Secondary Blocks . . . . . 200

SCLM Warning Level
  2 1. None
  _ 2. Warn
    3. Error

-----
New Edit Profile Settings and Overrides

-----
Please press HELP to see important information about setting edit profile
defaults.

-----
Profile Initial Macro _____ Force initial macro
F1=Help      F2=Split      F3=Exit      F7=Backward  F8=Forward  F9=Swap
F12=Cancel

```

Figure 7. Modify PDF Edit Configuration Settings panel (ISPPMOD1)

```

Modify PDF Edit Configuration Settings
Command ==>>> _____ More: - +

Enter "/" to select option      Enter "/" to force settings
/  STATS ON                     _  STATS
  _ RECOVERY ON                 _  RECOVERY
  _ RECOVERY warning message    _  RECOVERY warning
  _ SETUNDO ON                  _  SETUNDO
  _ PACK ON                     _  PACK
  _ CAPS ON
  _ NOTE ON                     HEX Mode . . . 2 1. ON
  _ NUMBER ON                   _ 2. OFF
  _ COBOL Numbers               _ 3. VERT
  _ Standard Numbers            _ 4. DATA
  _ AUTONUM ON
  _ AUTOLIST ON                 NULLS Mode . . 1 1. ON STD
  _ PROFILE LOCK                _ 2. ON ALL
  _ AUTOSAVE ON                 _ 3. OFF
  _ AUTOSAVE PROMPT

-----
Edit Highlighting
Language
  1 1. Automatic determination   Enter "/" to select option
  _ HILITE ON
F1=Help      F2=Split      F3=Exit      F7=Backward  F8=Forward  F9=Swap
F12=Cancel

```

Figure 8. Modify PDF Edit Configuration Settings panel (ISPPMOD1)

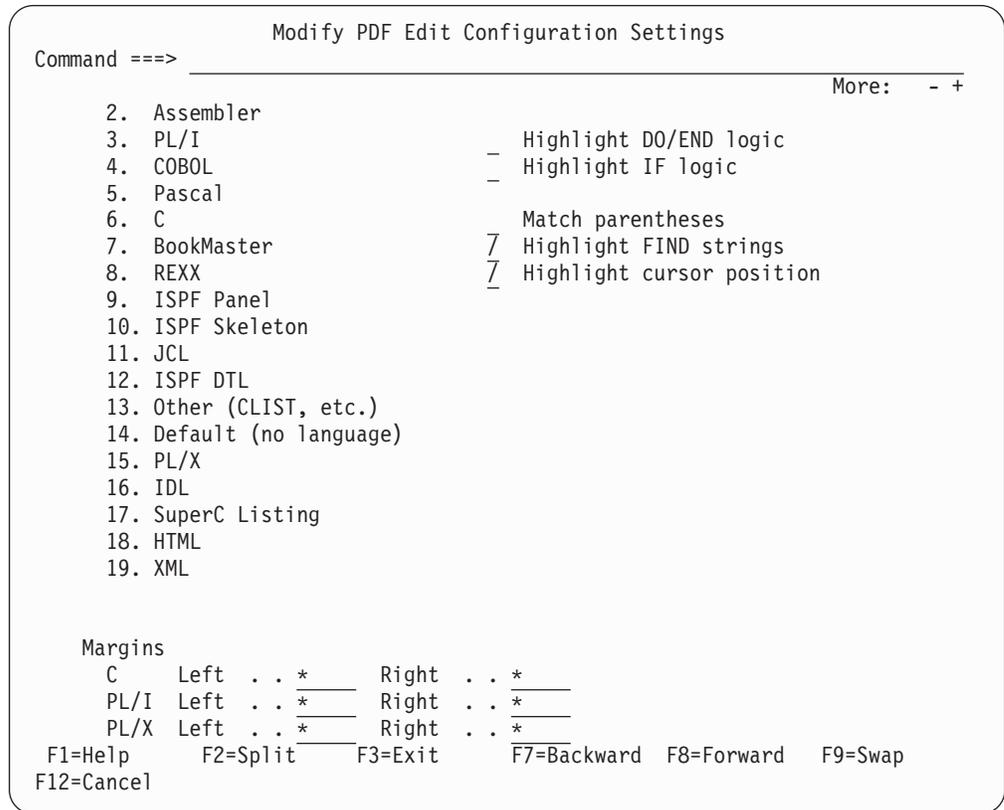


Figure 9. Modify PDF Edit Configuration Settings panel (ISPPMOD1)

Selecting Option 2, **Edit/View/Browse VSAM Settings** on the Create/Modify ISPF Configuration panel (ISPPMOD) displays the Modify Edit/View/Browse VSAM Settings panel (ISPPMOD2).

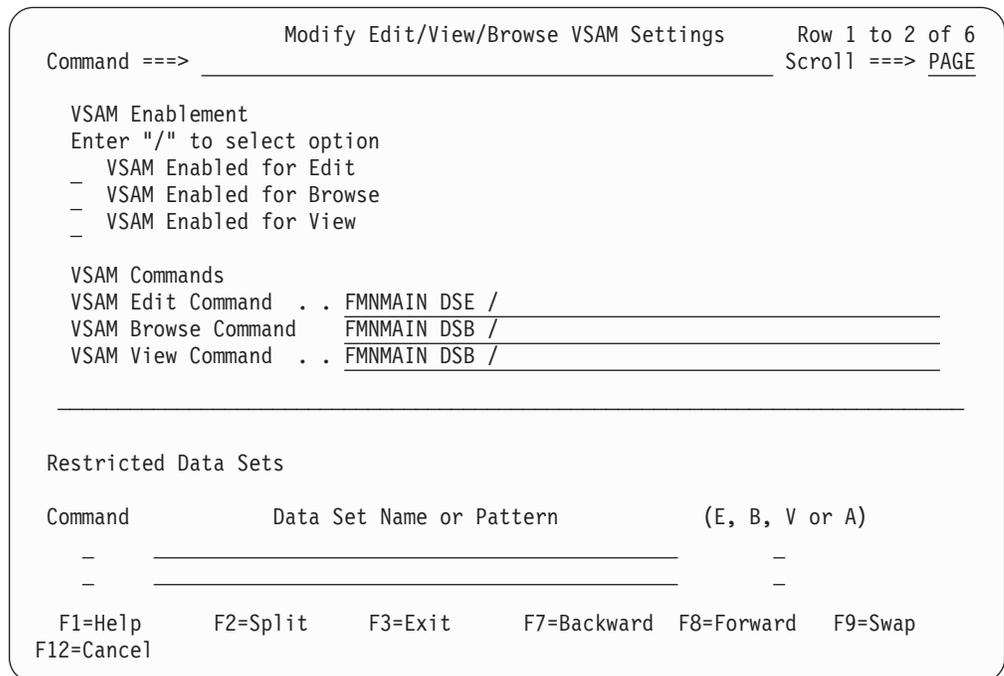


Figure 10. Modify Edit/View/Browse VSAM Settings panel (ISPPMOD2)

Selecting Option 3, **PDF Exits and Other PDF Settings** on the Create/Modify ISPF Configuration panel (ISPPMOD) displays the **Modify PDF Configuration Settings** panel (ISPPMOD3). Scroll down to display all the fields on this panel.

```

                                Modify PDF Configuration Settings
Command ==>>> _____ More: +

PDF Exits
Data Set Allocation Program Exit . . . . . _____
Print Utility Program Exit . . . . . _____
Print Utility Command Exit . . . . . _____
Compress Program Exit . . . . . _____
Compress Command Exit . . . . . _____
Data Set List Filter Program Exit . . . . . _____
Member List Filter Program Exit . . . . . _____
Data Set Name Change Program Exit . . . . . _____
Data Set List Line Command Program Exit . . . . . _____
Activity Monitoring Program Exit . . . . . _____
Member List Line Command Program Exit . . . . . _____
Member List Line Command Command Exit . . . . . _____

PDF Data Set Characteristics

Outlist Utility                               SuperC Block Sizes
Record Length . . . 133                       List Data Set . . . . . 0
Block Size . . . . . 13566                     Update Data Set . . . . . 0
F1=Help    F2=Split    F3=Exit    F7=Backward F8=Forward F9=Swap
F12=Cancel

```

Figure 11. Modify PDF Configuration Settings panel (ISPPMOD3)

```

                                Modify PDF Configuration Settings
Command ==>>> _____ More: - +

Block Size . . . . . 13566      Update Data Set . . . . . 0
Primary Blocks . . . 200       Profile Data Set . . . . . 0
Secondary Blocks . . 100      Statements Data Set . . . . . 0
                               Listing Primary Quantity . . 50
                               Listing Secondary Quantity 100
                               Update Primary Quantity . . 15
                               Update Secondary Quantity . . 30

Move/Copy Settings
Enter "/" to select option
/ Allow Creation of Move/Copy Target Data Set

When to Use IEBCOPY
0 0. Use when processing PDSEs or when using COPYMOD to Copy to a Smaller
  Block Size
  1. Always use IEBCOPY for Load Modules
  2. Use IEBCOPY for PDSEs only

When to use COPY or COPYMOD
F1=Help    F2=Split    F3=Exit    F7=Backward F8=Forward F9=Swap
F12=Cancel

```

Figure 12. Modify PDF Configuration Settings panel (ISPPMOD3)

```

                                Modify PDF Configuration Settings
Command ===> _____ More: - +

When to use COPY or COPYMOD
2 1. Use COPY if the target block size is equal to or greater than the
   source block size, COPYMOD otherwise
  2. Use COPY if the target block size is equal to the source block size,
   COPYMOD otherwise
  3. Always use COPYMOD

DSLIST Removable Media Settings
Enter "/" to select option
_ Enable RM/Tape Commands

RM/Tape Command . . . %EDGRPD34
Command APPLID . . . EDG

Other PDF Settings
Default PDF Unit . . . . . SYSALLDA
Volume for Migrated Data Sets . . . . . MIGRAT
Delete Command for Migrated Data Sets HDELETE
Allowed Allocation Units . . . . . ANY
F1=Help      F2=Split      F3=Exit      F7=Backward  F8=Forward  F9=Swap
F12=Cancel

```

Figure 13. Modify PDF Configuration Settings panel (ISPPMOD3)

```

                                Modify PDF Configuration Settings
Command ===> _____ More: - +

Allowed Allocation Units . . . . . ANY
Maximum IEBCOPY Return Code . . . . . 0
Pathname Substitution Character . . . . . !

Enter "/" to select option
  Allocate Before Uncatalog
  / Verify Expiration Dates
  / Use SuperC Program Interface
  - Monitor Edit Macro Commands via the Activity Monitoring Exit
  / Allow SUBMIT from Browse
  / Allow SUBMIT from View
  / Warn when rename target could be a GDG
  / Default Edit/Browse/View member list from Option 3.4
  / Enable View
  - Use Panel ISRTSOA in Option 6
  - Print using ICF
  - Disallow wildcards in the high level qualifier for Data Set List
  - Disable all ENQ displays
  / Fail on LMF lock requests
F1=Help      F2=Split      F3=Exit      F7=Backward  F8=Forward  F9=Swap
F12=Cancel

```

Figure 14. Modify PDF Configuration Settings panel (ISPPMOD3)

Selecting Option 4, **ISPF Site-wide Defaults** on the Create/Modify ISPF Configuration panel (ISPPMOD) displays the Modify ISPF Sitewide Defaults panel (ISPPMOD4). Scroll down to display all the fields on this panel.

```

                                Modify ISPF Sitewide Defaults
Command ==> _____ More: +
If you select any RESET fields in the sections: ISPF Site-wide Defaults, CUA
Color Settings, Log and List Defaults, Terminal and User Defaults, Workstation
Defaults, or Workstation Download Defaults you must increment the Sitewide
Defaults Version Level field to enable the RESET fields you have selected.
Increment only the last 3 digits of the Sitewide Defaults Version Level. ISPF
is always shipped with the Sitewide Defaults Version Level field set to 43000.
This value does not change with new versions or releases of ISPF.

Sitewide Defaults Version Level . . 43001

General settings                                Reset flags
Enter "/" to select option                      Enter "/" to select option
/ Tab to Point and Shoot                        - Reset Tab to Point and Shoot
/ Tab to Action Bars                           - Reset Tab to Action Bars
/ Use Session Manager                          - Reset Use Session Manager
/ Jump From Leader Dots                        - Reset Jump From Leader Dots
/ Always Show Split Line                       - Reset Show Split Line
/ Long Messages in Pop-ups                     - Reset Long Messages in Pop-ups
- Edit PRINTDS Command                         - Reset Edit PRINTDS Command
/ Restore Test/Trace Options                   - Reset Restore Test/Trace Options
/ Display Panels in CUA Mode                   - Reset Display Panels in CUA Mode
/ Use Keylists                                 - Reset Use Keylists
F1=Help    F2=Split    F3=Exit                - F7=Backward F8=Forward F9=Swap
F12=Cancel

```

Figure 15. Modify ISPF Sitewide Defaults panel 1 (ISPPMOD4)

```

                                Modify ISPF Sitewide Defaults
Command ==> _____ More: - +
/ Show Pfkeys                                  - Reset Show Pfkeys
Select Option 7.1 Dialog Test Panel            - Reset LOG Data Set Disposition
1 1. ISPYFP                                    - Reset LIST Data Set Disposition
- 2. ISPYFPA                                   - Reset Command Line Placement
  3. ISPYFPB

Command Line Placement                          PRINTDS Option
2 1. Bottom                                    1 1. DEST
- 2. Asis                                       2. WRITER

Scroll Defaults                                 Status Area Default
4 1. PAGE                                       2 1. Calendar
- 2. HALF                                       2. Session
  3. MAX                                        3. Function Keys
  4. CSR                                        4. User Point and Shoot
  5. DATA                                      5. None

Minimum Scroll Value 0                          Reset flags
Maximum Scroll Value 9999                       - Reset Scroll Values

F1=Help    F2=Split    F3=Exit                F7=Backward F8=Forward F9=Swap
F12=Cancel

```

Figure 16. Modify ISPF Sitewide Defaults panel 2 (ISPPMOD4)

```

                                Modify ISPF Sitewide Defaults
Command ==>>> _____ More: - +

Member list options
Enter "/" to select option
/ Scroll Member List           _ Reset Scroll Member List
_ Allow Empty Member List     _ Reset Empty Member List Options
_ Allow Empty Member List
  (nomatch)
_ Empty Member List for Edit Only

ISPF Data Set Characteristics

Log Data Set                   List Data Set
Record Length . . . . : 125    Records per Block . . . 26
Block Size . . . . . 0

ISPCTL0 Data Set              ISPLSTx Data Set
Record Length . . . . : 80    Record Length . . . . : 121
Block Size . . . . . 800     Block Size . . . . . 3146
Primary Quantity . . . 10     Primary Quantity . . . 10
Secondary Quantity . . 100    Secondary Quantity . . 100

ISPCTLx Data Set              ISPWRKx Data Set
Record Length . . . . : 80    Record Length . . . . : 256
F1=Help      F2=Split      F3=Exit      F7=Backward F8=Forward F9=Swap
F12=Cancel

```

Figure 17. Modify ISPF Sitewide Defaults panel 3 (ISPPMOD4)

```

                                Modify ISPF Sitewide Defaults
Command ==>>> _____ More: - +

Block Size . . . . . 800     Block Size . . . . . 0
Primary Quantity . . . 10     Primary Quantity . . . 10
Secondary Quantity . . 100    Secondary Quantity . . 100

LOG Data Set Disposition      LIST Data Set Disposition
1 1. No LOG Data Set         1 1. No LIST Data Set
_ 2. Print and Delete        _ 2. Print and Delete
  3. Delete Without Printing 3. Delete Without Printing
  4. Keep                    4. Keep
  5. Keep and Allocate a New LOG 5. Keep and Allocate a New LIST

Enter "/" to select option
_ Use Default PDF Unit for ISPF Data Sets

Temporary Data Sets
Additional Qualifier . . ISP&SEQ.

ISP Multi-logon Profile Options
F1=Help      F2=Split      F3=Exit      F7=Backward F8=Forward F9=Swap
F12=Cancel

```

Figure 18. Modify ISPF Sitewide Defaults panel 4 (ISPPMOD4)

```

                                Modify ISPF Sitewide Defaults
Command ===> _____ More: - +

Enter "/" to select option
/ Multi-logon Profile Sharing
/ Prompt for Profile ENQ Lockout   ENQ Lock Wait . . . . . 500
_ Reset Shared Profile Settings   ENQ Lock Retry Count . . 2

System Profile conflicts          Reference List conflicts
3 1. Keep                         3 1. Keep
_ 2. Discard                       2. Discard
  3. Prompt                        3. Prompt

ISPF Profile conflicts           Edit Profile conflicts
3 1. Keep                         3 1. Keep
_ 2. Discard                       2. Discard
  3. Prompt                        3. Prompt

Application Profile conflicts     Batch Profile conflicts
3 1. Keep                         2 1. Keep
_ 2. Discard                       2. Discard
  3. Prompt

Other Profile conflicts
3 1. Keep
F1=Help      F2=Split      F3=Exit      F7=Backward F8=Forward F9=Swap
F12=Cancel

```

Figure 19. Modify ISPF Sitewide Defaults panel 5 (ISPPMOD4)

```

                                Modify ISPF Sitewide Defaults
Command ===> _____ More: - +

  2. Discard
  3. Prompt

_____

Local PRINTDS Options . . . NONUM

_____

_____

F1=Help      F2=Split      F3=Exit      F7=Backward F8=Forward F9=Swap
F12=Cancel

```

Figure 20. Modify ISPF Sitewide Defaults panel 6 (ISPPMOD4)

Selecting Option 5, **ISPDFLTS**, **CUA Colors**, and **Other DM Settings** on the Create/Modify ISPF Configuration panel (ISPPMOD) displays the Modify

ISPDFLTS and Other DM Settings panel (ISPPMOD5). Scroll down to display all the fields on this panel.

```

Modify ISPDFLTS and Other DM Settings
Command ===> _____ More: +
ISPDFLTS Settings
Number of Rows for TBADD  1_____ Enter "/" to select option
                                _ Enable ISPF Exits

Command Table Settings
APPLID for User Command Table 1 . . USER
APPLID for User Command Table 2 . . _____
APPLID for User Command Table 3 . . _____
APPLID for Site Command Table 1 . . SITE Site Command Table Search Order
APPLID for Site Command Table 2 . . _____ - 1. Before
APPLID for Site Command Table 3 . . _____ - 1. After

Miscellaneous DM Settings
Maximum Number of Split Screens . . 8
Year 2000 Sliding Rule . . . . . 65
Retrieve Command Stack Size . . . . 512
TPUT Buffer Size . . . . . 0
Default Primary Panel . . . . . _____
Default LIBDEF Processing Option _____ (COND UNCOND STACK or STKADD)
F1=HELP      F2=SPLIT      F3=END      F4=RETURN   F5=RFIND   F6=RCHANGE
F7=UP        F8=DOWN      F9=SWAP    F10=LEFT   F11=RIGHT  F12=RETRIEVE

```

Figure 21. Modify ISPDFLTS and Other DM Settings panel (ISPPMOD5)

```

Modify ISPDFLTS and Other DM Settings
Command ===> _____ More: - +
Default session language
- 1. English
  2. Uppercase English
  3. German
  4. Swiss German
  5. Japanese

ZDATEFD may use the national language
convention to replace the characters YY,
MM, DD and the national language delimiter.
Date Format (ZDATEFD) . . _____

ZDATEF must use the characters YY, MM, DD
(in any order) and the national language
delimiter.

Examples:
English - YY/MM/DD
German - TT.MM.JJ

Examples:
English - YY/MM/DD
German - DD.MM.YY

F1=HELP      F2=SPLIT      F3=END      F4=RETURN   F5=RFIND   F6=RCHANGE
F7=UP        F8=DOWN      F9=SWAP    F10=LEFT   F11=RIGHT  F12=RETRIEVE

```

Figure 22. Modify ISPDFLTS and Other DM Settings panel (ISPPMOD5)

```

                                Modify ISPDFLTS and Other DM Settings
Command ==>> _____ More: - +

Date Format (ZDATEF) . . . _____ |
Time Separator Character  _

-----

If you select any RESET fields in the sections: ISPF Site-wide Defaults, CUA
Color Settings, Log and List Defaults, Terminal and User Defaults,
Workstation Defaults, or Workstation Download Defaults you must increment
the Sitewide Defaults Version Level field to enable the RESET fields you
have selected. Increment only the last 3 digits of the Sitewide Defaults
Version Level. ISPF is always shipped with the Sitewide Defaults Version
Level field set to 43000. This value does not change with new versions or
releases of ISPF.

Sitewide Defaults Version Level . . 43000

CUA Color Settings
F1=HELP      F2=SPLIT      F3=END        F4=RETURN     F5=RFIND      F6=RCHANGE
F7=UP        F8=DOWN        F9=SWAP       F10=LEFT      F11=RIGHT     F12=RETRIEVE

```

Figure 23. Modify ISPDFLTS and Other DM Settings panel (ISPPMOD5)

The remaining screens enable you to specify the CUA Color settings for these panel elements:

- Action Bar Selected Choice
- Action Bar Separator Line
- Action Bar Unselected Choice
- Action Message Text
- Caution Text
- Choice Entry Field
- Column Heading
- Descriptive Text
- Emphasized Text
- Error Emphasis
- Field Prompt
- Function Keys
- Informational Message Text
- List Entry Field
- List Item Description
- List Item
- Normal Entry Field
- Normal Text
- Panel ID
- Panel Information
- Panel Title
- Point and Shoot
- Pulldown Available Choice
- Pulldown Unavailable Choice
- Reference Phrase
- Scroll Information
- Selection Available Choice
- Selection Unavailable Choice
- Variable Output Information
- Warning Message Text

- Warning Message
- Workarea Separator Line

Figure 24 shows the options for the first two panel elements.

```

Modify ISPDFLTS and Other DM Settings
Command ==> _____ End of data
More: -

CUA Color Settings

Action Bar Selected Choice
Color          Intensity      Hilite          Reset Setting
6 1. Blue      0 0. Low        0 0. None       2 1. Yes
 2. Red        2. High        1. Blink       2. No
 3. Pink
 4. Green
 5. Turquoise
 6. Yellow
 7. White

Action Bar Separator Line
Color          Intensity      Hilite          Reset Setting
1 1. Blue      0 0. Low        0 0. None       2 1. Yes
 2. Red        2. High        1. Blink       2. No
 3. Pink
 4. Green
 5. Turquoise
 6. Yellow
 7. White

F1=HELP      F2=          F3=END        F4=DATASETS  F5=FIND      F6=CHANGE
F9=SWAP      F10=LEFT    F11=RIGHT     F12=SUBMIT

```

Figure 24. Modify ISPDFLTS and Other DM Settings panel (ISPPMOD5)

### System Profile (ISPSPROF) Settings panels

Selecting Option 6, **Log and List Defaults** on the Create/Modify ISPF Configuration panel (ISPPMOD) displays the Modify Log/List Configuration Settings panel (ISPPMOD6). Scroll down to display all the fields on this panel.

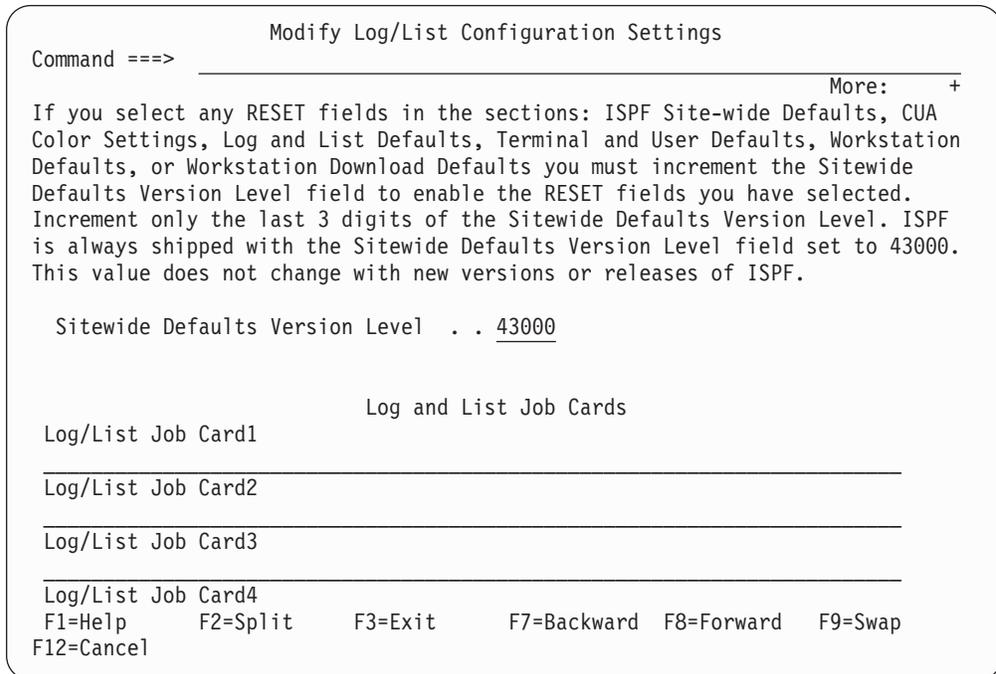


Figure 25. Modify Log/List Configuration Settings panel (ISPPMOD6)

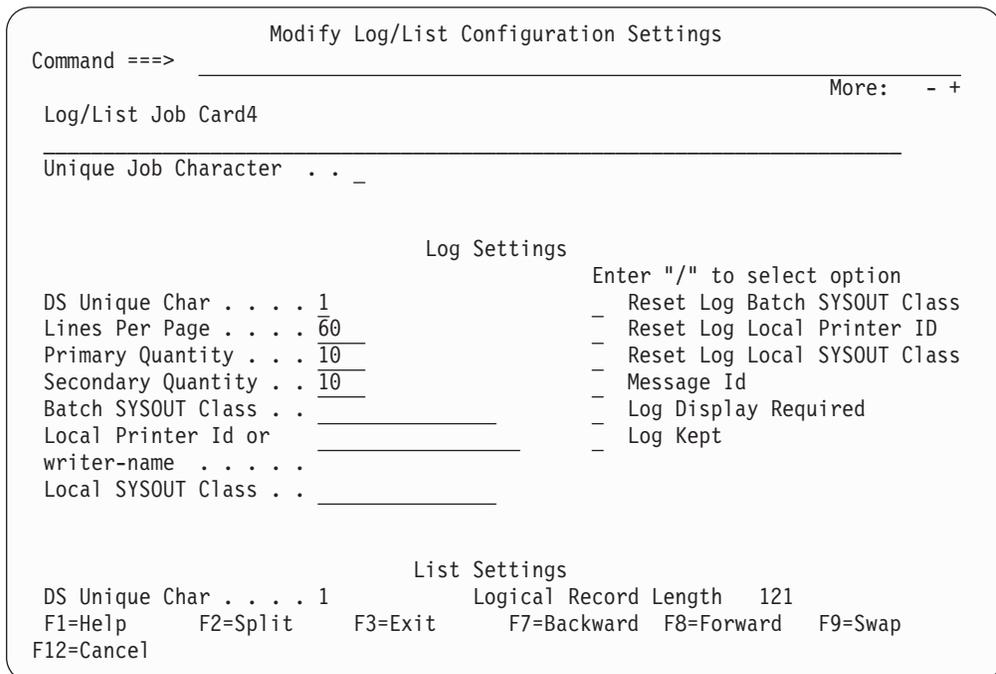


Figure 26. Modify Log/List Configuration Settings panel (ISPPMOD6)

```

                                Modify Log/List Configuration Settings
Command ==> _____ More: -
writer-name . . . . .
Local SYSOUT Class . . _____

                                List Settings
DS Unique Char . . . . . 1          Logical Record Length 121
Lines Per Page . . . . . 60         Primary Quantity . . . 100
Line Length . . . . . 120          Secondary Quantity . . 200
List Record Format                Batch SYSOUT Class . . _____
1 1. FBA                          Local Printer Id or
- 2. VBA                          writer-name . . . . . _____
                                Local SYSOUT Class . . _____

Enter "/" to select option
- Reset List Batch SYSOUT Class
- Reset List Local Printer ID
- Reset List Local SYSOUT Class
- List Display Required
- List Kept
- Reset List RECFM and LRECL
F1=Help   F2=Split   F3=Exit   F7=Backward F8=Forward F9=Swap
F12=Cancel

```

Figure 27. Modify Log/List Configuration Settings panel (ISPPMOD6)

Selecting Option 7, **Terminal and User Defaults** on the Create/Modify ISPF Configuration panel (ISPPMOD) displays the Modify Terminal and User Settings panel (ISPPMOD7). Scroll down to display all the fields on this panel.

```

                                Modify Terminal and User Settings
Command ==> _____ More: +
If you select any RESET fields in the sections: ISPF Site-wide Defaults, CUA
Color Settings, Log and List Defaults, Terminal and User Defaults, Workstation
Defaults, or Workstation Download Defaults you must increment the Sitewide
Defaults Version Level field to enable the RESET fields you have selected.
Increment only the last 3 digits of the Sitewide Defaults Version Level. ISPF
is always shipped with the Sitewide Defaults Version Level field set to 43000.
This value does not change with new versions or releases of ISPF.

Sitewide Defaults Version Level . . 43000

Settings Panel Defaults                Window Frame Color/Intensity
Color                                Intensity
Aspect Ratio . . . . . 0             6 1. Blue                2 0. Low
Input Field Pad Character  B         2. Red                    2. High
Delimiter . . . . . ;
Family Printer . . . . . : 2
Device Name . . . . . _____
                                3. Pink
                                4. Green
                                5. Turquoise
                                6. Yellow
                                7. White

Enter "/" to select option                Screen Format
- Reset Device Name                        2 1. DATA
                                2. STD
F1=Help   F2=Split   F3=Exit   F7=Backward F8=Forward F9=Swap
F12=Cancel

```

Figure 28. Modify Terminal and User Settings panel (ISPPMOD7)

```

Modify Terminal and User Settings
Command ===> _____ More: - +

3. MAX
4. PART

Terminal Type
3 1. 3277      2. 3277A     3. 3278      4. 3278A
5. 3290A     6. 3278T     7. 3278CF    8. 3277KN
9. 3278KN    10. 3278AR   11. 3278CY   12. 3278HN
13. 3278HO   14. 3278IS   15. 3278L2   16. BE163
17. BE190    18. 3278TH   19. 3278CU   20. DEU78
21. DEU78A   22. DEU78T   23. DEU90A   24. SW116
25. SW131    26. SW500    27. 3278GR

-----

Environ Defaults
Termtrac DD Name . . ISPSNAP
Enter "/" to select option
_ Reset Termtrac DD Name

Environ Enblump      Environ Termtrac
2 1. ON              2 1. ON
2. OFF              2. OFF
                   3. ERROR

Retrieve Defaults
Retrieve Minimum Length . . 1
Retrieve Cursor Position
1 1. End of command
1 2. Start of command

-----

F1=Help      F2=Split      F3=Exit      F7=Backward  F8=Forward  F9=Swap
F12=Cancel

```

Figure 29. Modify Terminal and User Settings panel (ISPPMOD7)

```

Modify Terminal and User Settings
Command ===> _____ More: - +

General Defaults
Character Set Load Module ISP3278
Enter "/" to select option
_ Reset Character Set Load Module

Function Keys Defaults
Number of Keys
1 1. 12 2. 24
FKA Setting
2 1. SHORT 2. LONG 3. OFF
Enter "/" to select option
/ Display Prefix

GUI Global Colors
Enter a GUI color code for each Host color
Host colors _____ GUI colors _____
Blue . . . . 1      0 Black      8 Dark Gray
Red . . . . 4      1 Dark Blue  9 Light Blue
Pink . . . . 13     2 Dark Green 10 Light Green
Green . . . . 2     3 Dark Cyan  11 Light Cyan
Turquoise 3      4 Dark Red   12 Light Red
Yellow . . . . 6     5 Dark Magenta 13 Light Magenta
White . . . . 0     6 Brown      14 Yellow
                   7 Light Gray  15 White

F1=Help      F2=Split      F3=Exit      F7=Backward  F8=Forward  F9=Swap
F12=Cancel

```

Figure 30. Modify Terminal and User Settings panel (ISPPMOD7)

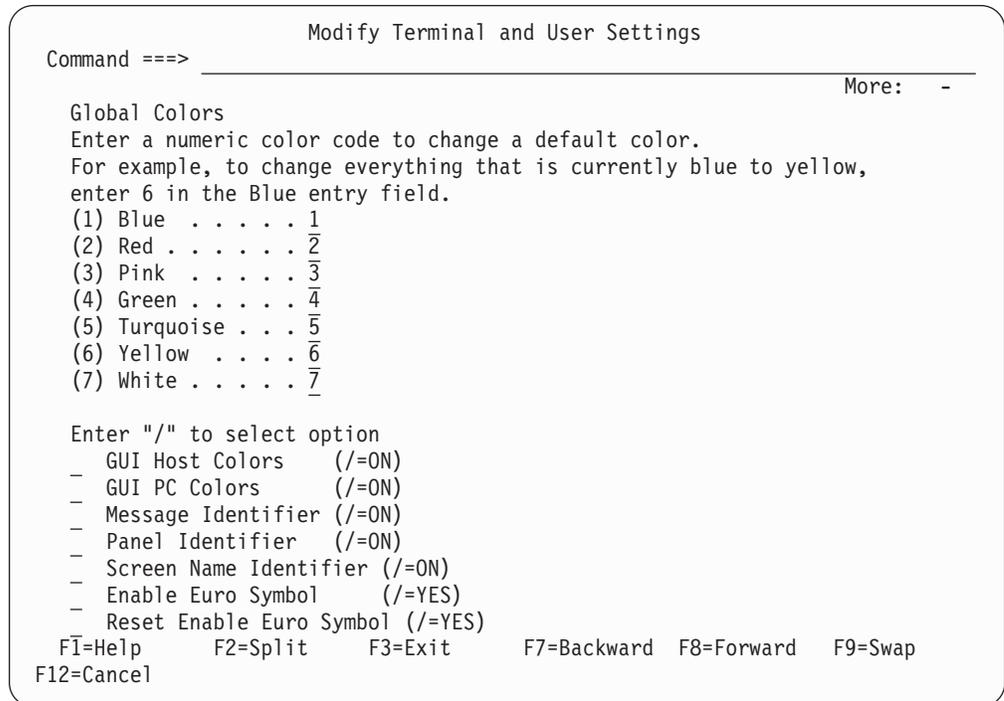


Figure 31. Modify Terminal and User Settings panel (ISPPMOD7)

Selecting Option 8, **Workstation Defaults** on the Create/Modify ISPF Configuration panel (ISPPMOD) displays the Workstation Defaults panel (ISPPMOD8). Scroll down to display all the fields on this panel.

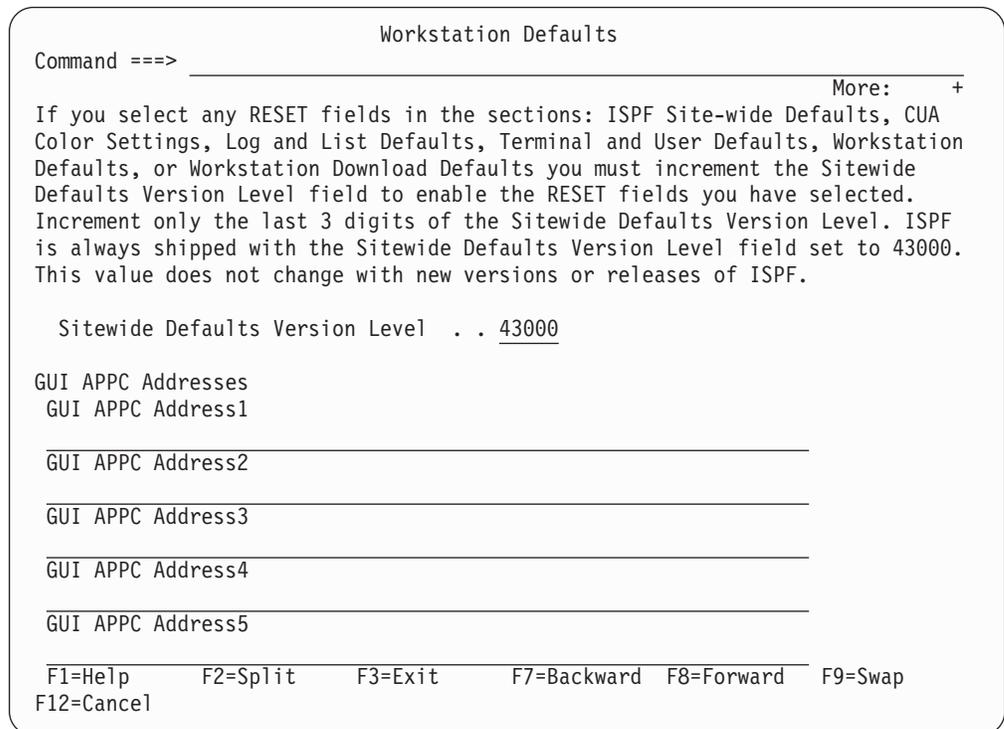


Figure 32. Workstation Defaults panel (ISPPMOD8)

```

Workstation Defaults
Command ==> _____ More: - +
GUI TCP Addresses
GUI TCP Address1
_____
GUI TCP Address2
_____
GUI TCP Address3
_____
GUI TCP Address4
_____
GUI TCP Address5
_____

GUI General Defaults
GUI Title

Host Code Page . . . . . 0000
Host Character Set . . . . . 0000

Enter "/" to select option
  Reset Host Code Page
  Reset Host Character Set
F1=Help   F2=Split   F3=Exit   F7=Backward F8=Forward F9=Swap
F12=Cancel

```

Figure 33. Workstation Defaults panel (ISPPMOD8)

```

Workstation Defaults
Command ==> _____ More: - +

/ GUI Display ENTER Key
/ Save GUI Values
/ GUI Accelerator Support
/ GUI Download Images
- GUI Make Path for Images
/ Continue 3270 After Loss of WS Connection

GUI Window Frame
1 1. Standard
- 2. Fixed
  3. Dialog

GUI Background Color
1 1. Dialog
- 2. Standard

GUI Network
1 1. TCP/IP
- 2. APPC
  3. ISPDTPRF file

GUI Switch
1 1. GUI Display
- 2. No GUI Display

GUI Close Window
1 1. Cancel
- 2. End
  3. Exit
  4. Return

F1=Help   F2=Split   F3=Exit   F7=Backward F8=Forward F9=Swap
F12=Cancel

```

Figure 34. Workstation Defaults panel (ISPPMOD8)

Selecting Option 9, **Workstation Download Defaults** on the Create/Modify ISPF Configuration panel (ISPPMOD) displays the **Workstation Download Defaults** panel (ISPPMOD9). Scroll down to display all the fields on this panel.

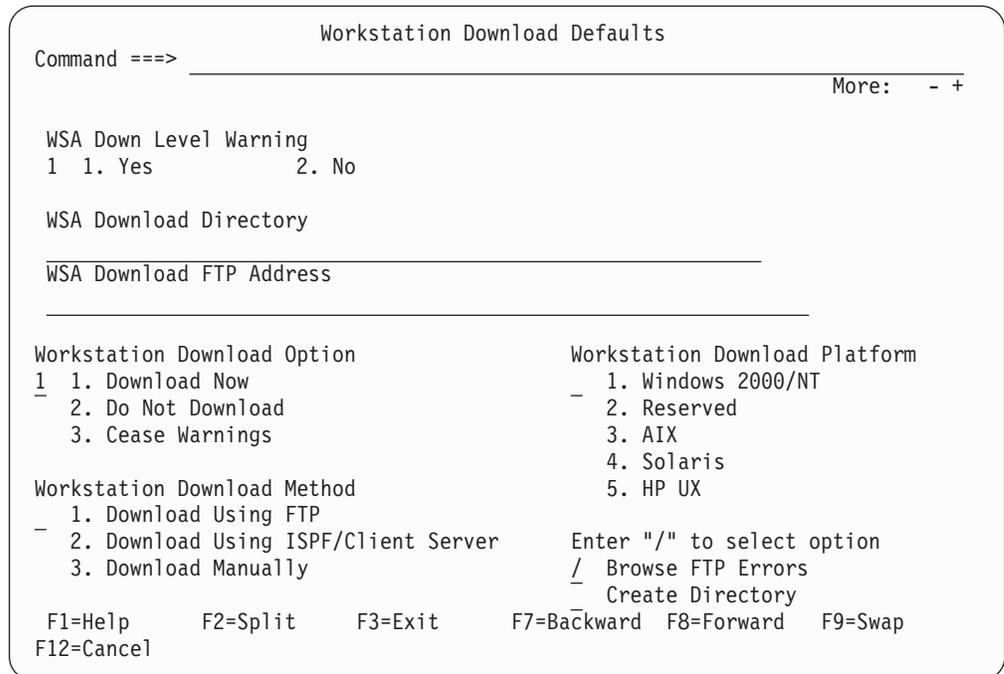
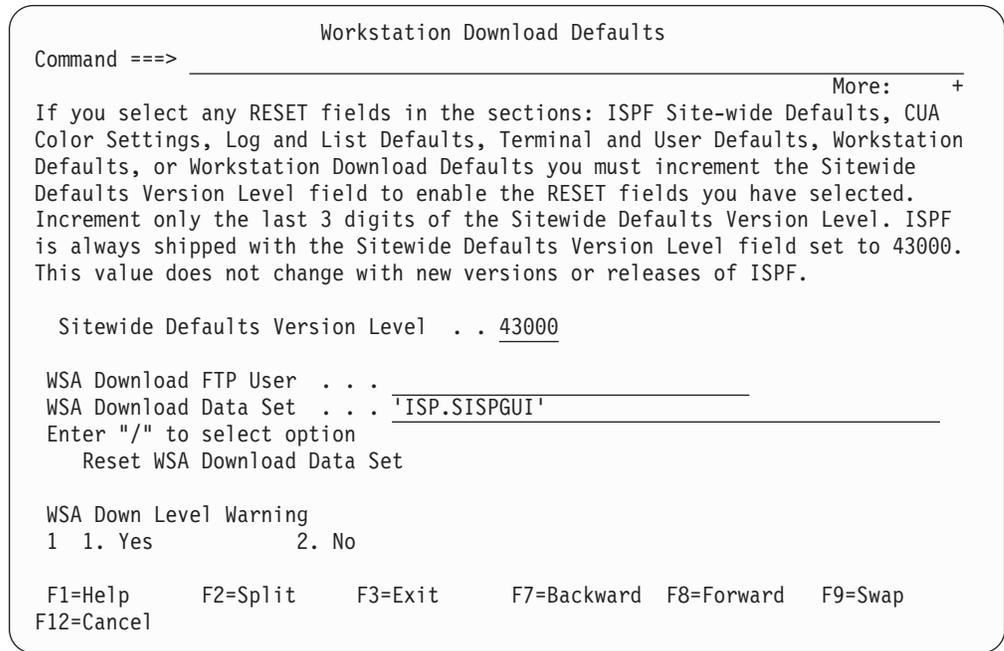


Figure 35. Workstation Download Defaults panel (ISPPMOD9)

## Edit Keyword File Configuration Table

If you choose Option 2, **Edit Keyword File Configuration Table** on the ISPF Configuration utility main menu panel, the Edit Keyword File panel appears, as shown in Figure 36 on page 32.

This option enables you to directly edit the keyword file and manually modify the configuration settings.

Comments can be added to the keyword file and the keywords can be rearranged, but such additions and changes are not preserved if you later use Option 1 **Create/Modify Settings and Regenerate Keyword File**. Only the changed values of keywords are carried forward. Full line comments are denoted by an asterisk (\*) in column one, or a forward slash and asterisk (/\*) in columns one and two. Comments on the same line as a keyword begin with a forward slash and asterisk, and end at the end of the line.

```

File Edit Edit_Settings Menu Utilities Compilers Test Help
-----
ISPCEDIT  LSACKV.CONFIG.KEYWORD.FILE(SYSTEM1) - 01.00      Columns 00001 00072
Command ==>>>

Modify, Add or Delete the keywords and values listed here. Enter END or EXIT
to save your changes, CANCEL to exit without saving your changes.

Note: Changes in the order of the keywords, or comments added to the file will
not be preserved if the file is regenerated using option 1.

000001 /* ISPF Configuration table definition.  Generated by REXX ISPCMOD  */
000002 /*   Created 10:40:04 on 9 Jan 2004                               */
000003 /*   by user LSACKV.                                             */
000004 /*   All values were included.                                   */
000005 /*                                                                 */
000006 /*-----*/
000007 /*                               PDF EXITS                          */
000008 /*-----*/
000009 DATA_SET_ALLOCATION_PROGRAM_EXIT          = NONE
000010 PRINT_UTILITY_PROGRAM_EXIT                = NONE
000011 PRINT_UTILITY_COMMAND_EXIT              = NONE
F1=Help      F2=Split      F3=Exit      F5=Rfind      F6=Rchange    F7=Up
F8=Down      F9=Swap       F10=Left     F11=Right     F12=Cancel

```

Figure 36. Edit Keyword File panel (ISPCEDIT)

If you make changes to the keyword file, when you end the edit session ISPF displays the confirmation window in Figure 37.

```

ISPPVERQ      Keyword File Verification
Command ==>>>

The Keyword file has been saved. You can run the ISPF
keyword verification to check the keywords and values in
the keyword file now, or exit without running the
verification. If you choose to exit without verifying the
keyword file you can run the verification later using the
Verify Keyword Table Contents option.

Instructions:
Press Enter to verify the keyword file.
Enter END or EXIT to exit without verification.

F1=Help      F2=Split      F3=Exit      F7=Backward
F8=Forward    F9=Swap       F12=Cancel

```

Figure 37. Keyword File Verification panel (ISPPVERQ)

On this Keyword File Verification window, you can indicate whether you want to run ISPF's keyword verification routine to check the keywords and values in your

keyword file. If you want to run the verification exec, press the Enter key. If you do not want to verify your changes, enter the END command, or EXIT.

## Verify Keyword Table Contents

If you choose Option 3, **Verify Keyword Table Contents** on the ISPF Configuration utility main menu panel, ISPF verifies that the keyword file is correct in these respects:

- each record is in the correct format
- each keyword specified is a recognized keyword
- each value is syntactically correct.

If any verification errors are found, a listing similar to the one shown in Figure 38 is displayed. The listing shows the line in error and the reason that ISPF flagged it as incorrect.

```
File Edit Edit_Settings Menu Utilities Compilers Test Help
-----
ISREDDE2  SYS04009.T120741.RA000.LSACKV.R0117181      Columns 00001 00072
Command ==>                                         Scroll ==> CSR
***** ***** Top of Data *****
000001 /* ISPF Configuration table verification report. */
000002 /*   Generated by ISPF REXX exec ISPCVERF          */
000003 /*   Created 12:07:41 on 9 Jan 2004 by LSACKV      */
000004 /*   Input keyword data set:                       */
000005 /*   ==> CONFIG.KEYWORD.FILE(SYSTEM1)                */
000006 /*                                                    */
000007
000008 Error line 9:  DATA_SET_ALLOCATION_PROGRAM_EXIT      = 1EXIT
000009   Value does not meet ISPF member naming conventions, first character
000010   cannot be numeric
000011
***** ***** Bottom of Data *****

F1=Help      F2=Split    F3=Exit     F5=Rfind    F6=Rchange  F7=Up
F8=Down      F9=Swap     F10=Left    F11=Right   F12=Cancel
```

Figure 38. Verification Failure Listing

## Build Configuration Table Load Module

If you choose Option 4, **Build Configuration Table Load Module** on the ISPF Configuration utility main menu panel, ISPF displays the Build Configuration Table Load Module panel, as shown in Figure 39 on page 34.

This option enables you to convert the keyword file you specify into a load module for ISPF to use to determine the session settings. You can create two separate load modules: the configuration table load module (default name ISPCFIGU), and the VSAM configuration load module (default name ISPCFIGV). The configuration table load module is always created, the VSAM configuration load module is only created if one of the VSAM restriction keywords is used in your file.

While you can create the load modules with any name initially (for example, if you want to create several different configurations for your installation), ISPF only

recognizes the default names of ISPCFIGU and ISPCFIGV. If you create load modules with other names and want to use them, you must rename them when they are moved into the ISPF execution data sets.

```

ISPPCONF                ISPF Configuration Utility

Build Configuration Table Load Module

Command ==> _____

Input Keyword File Data set
Data Set . . . CONFIG.KEYWORD.FILE
Member . . . . SYSTEM1

Output Configuration Table Load Module Data Set
Data Set . . . LOAD.LIBRARY

Optional fields (leave blank for ISPF to use defaults)
Object data set . . . _____
Configuration member _____ (Defaults to ISPCFIGU)
VSAM member . . . . . _____ (Defaults to ISPCFIGV)

F1=Help      F2=Split      F3=Exit      F7=Backward  F8=Forward
F9=Swap      F12=Cancel

2. Include defaults as comments
F1=Help      F2=Split      F3=Exit      F7=Backward  F8=Forward  F9=Swap
F12=Cancel

```

Figure 39. Build Configuration Table Load Module panel (ISPPBLD)

When the Build Configuration Table Load Module panel appears, the Input Keyword File Data Set field value is initialized with the name found on the ISPF Configuration utility main menu panel. Verification of the keyword file specified is automatically run before the keyword file is converted into a load module. If ISPF finds any errors during the verification process, they are logged to a temporary data set, and ISPF puts you into View mode on the messages data set when the verification is complete. The build *is not* done until the verification completes without error.

The other entry fields found on the Build Configuration Table Load Module panel are:

**Output Configuration Table Load Module Data Set**

The name of the pre-existing partitioned data set into which the generated configuration table load module is to be stored.

**Object Data Set**

Optional. The name of the pre-existing partitioned data set into which the generated configuration table object module is to be stored. If no object data set is specified, ISPF uses a temporary data set.

**Configuration Member**

Optional. The member name of the configuration table load module created. If no member name is specified, a value of ISPCFIGU is used.

**VSAM Member**

Optional. The member name of the VSAM configuration load module created. If no member name is specified, a value of ISPCFIGV is used.

## Convert Assembler Configuration Table to Keyword File

If you choose Option 5, **Convert Assembler Configuration Table to Keyword File** on the ISPF Configuration utility main menu panel, ISPF converts a configuration table assembler source file used in releases of ISPF before OS/390 Version 2 Release 8.0 into the new keyword format.

You specify the assembler source file in the Configuration Table Assembler Source Data Set field on the ISPF Configuration utility main menu panel. The keyword file that is generated is placed into the data set and member you specify in the Keyword File Data Set field.

The Output File Content for Keyword File field controls how much data is written to the keyword file during the conversion. Valid values for this field are:

- 1 Only those values that are different than the default values are included in the keyword file.
- 2 Those values different than the default values are included in the keyword file, and all default values are included as comment lines.
- 3 All values are included in the keyword file.

If the output keyword file specified in the Keyword File Data Set field already exists, the Confirm Conversion panel asks if you want to overwrite it. If you want to overwrite the file press the Enter key, otherwise enter END or CANCEL to exit.

## Build SMP/E USERMOD

If you choose Option 6, **Build SMP/E USERMOD** on the ISPF Configuration utility main menu panel, ISPF displays the Build SMP/E USERMOD panel (ISPPSMP), as shown in Figure 40.

```
Build SMP/E USERMOD
Command ==> _____

Keyword File Data set
Data Set . . . . . CONFIG.KEYWORD.FILE
Member . . . . . ISRTSO

SMP/E Data Set . . _____
SYSMOD Identifier for USERMOD . . . . . _____
FMID for USERMOD . . . . . _____

Target Library DDDEF names
SYSLIB for keyword file source . . . . . _____
SYSLIB for load modules . . . . . _____

Distribution Library DDDEF names
DISTLIB for keyword file source . . . . . _____
DISTLIB for load modules . . . . . _____

Prior USERMODs to supersede (SUP)
Prior USERMOD to supersede . . . . . _____
Prior USERMOD to supersede . . . . . _____
Prior USERMOD to supersede . . . . . _____
Prior USERMOD to supersede . . . . . _____
F1=Help      F2=Split      F3=Exit      F7=Backward  F8=Forward
F9=Swap      F12=Cancel
```

Figure 40. Build SMP/E USERMOD panel (ISPPSMP)

This option enables you to convert the keyword file you specify into load modules (a configuration table load module and a VSAM configuration load module), and to package the keyword file source code and the generated load modules in an

SMP/E USERMOD. The configuration table load module is always created. The VSAM configuration load module is only created if one of the VSAM restriction keywords is used in your keyword file. The default names of ISPCFIGU (for the configuration load module) and ISPCFIGV (for the VSAM configuration load module) are always used when the load modules are packaged in an SMP/E USERMOD.

When the Build SMP/E USERMOD panel appears, the Keyword File Data Set field is initialized with the name found on the ISPF Configuration utility main menu panel. All entry fields on the Build SMP/E USERMOD panel are required except the "Prior USERMOD to supersede" fields. The other entry fields on the Build SMP/E USERMOD panel are:

#### **SMP/E Data Set**

The name of the pre-existing partitioned data set into which the SMP/E USERMOD is to be stored. The USERMOD is stored in a member named the same as the SYSMOD ID used for the USERMOD. This data set must be record format FB with LRECL 80.

#### **SYSMOD Identifier for USERMOD**

A seven-character identifier for the SMP/E USERMOD. The identifier is used in the ++USERMOD statement and is used for the member name in the SMP/E Data set.

#### **FMID for USERMOD**

The FMID for the FMID operand of the SMP/E USERMOD. The FMID for ISPF z/OS Version 1 Release 11.0 is HIF5H02.

#### **Target Library DDDEF names**

##### **SYSLIB for keyword file source**

The DDDEF name to be used as the SYSLIB for the keyword source in the SMP/E USERMOD. The DDDEF must exist in the target zone that has ISPF installed. The data set in the DDDEF entry must have the same record format and LRECL as your keyword source data set.

##### **SYSLIB for load modules**

The DDDEF name to be used as the SYSLIB for the load modules in the SMP/E USERMOD. The DDDEF must exist in the target zone that has ISPF installed. The data set in the DDDEF entry must have record format U and LRECL 0 with a block size equal to or greater than 6144.

#### **Distribution Library DDDEF names**

##### **DISTLIB for keyword file source**

The DDDEF name to be used as the DISTLIB for the keyword source in the SMP/E USERMOD. The DDDEF must exist in the target zone and distribution zone that has ISPF installed. The data set in the DDDEF entry must have the same record format and LRECL as your keyword source data set.

##### **DISTLIB for load modules**

The DDDEF name to be used as the DISTLIB for the load modules in the SMP/E USERMOD. The DDDEF must exist in the target zone and distribution zone that has ISPF installed. The data set in the DDDEF entry must have record format U and LRECL 0 with a block size equal to or greater than 6144.

### **Prior USERMODs to supersede (SUP)**

Four optional fields for the seven-character name of a previous USERMOD to be superseded by this SMP/E USERMOD. The previous USERMODs (up to four) are included in the SUP operand of the SMP/E USERMOD. The four fields must be filled from top to bottom.

When you complete the required fields on the panel and press Enter, the keyword file specified is automatically verified before it is converted into load modules and an SMP/E USERMOD is built. If ISPF finds any errors during verification, the errors are logged to a temporary data set and ISPF puts you into View mode on the messages data set.

If the verification process completes without error the keyword file is packaged in an SMP/E USERMOD as **++DATA (ISPCFIGU)**. The configuration load module is generated and packaged as **++PROGRAM (ISPCFIGU)**.

The VSAM load module, if required, is generated and packaged as **++PROGRAM (ISPCFIGV)**.

The SMP/E USERMOD is stored in the data set specified in the SMP/E Data Set field on the Build SMP/E USERMOD panel (ISPPSMP), using a member name equal to the SYSMOD ID specified in the SYSMOD Identifier for USERMOD field.

### **Installing the USERMOD on your system**

To install the resulting SMP/E USERMOD on your system, perform these steps.

1. Create target and distribution libraries for the keyword source file. The target and distribution libraries for the keyword source must have the same record format and LRECL as the keyword file that you used to build the USERMOD. One or two tracks will be sufficient space.
2. Create target and distribution libraries for the load modules. The target and distribution load libraries must be record format U, LRECL 0, with a block size equal to or greater than 6144. One or two tracks will be sufficient space.
3. Add DDDEFs to your target zone for the target and distribution libraries using the DDDEF names you specified on the Build SMP/E USERMOD panel, with the names of the appropriate target or distribution libraries you created.
4. Add DDDEFs to your distribution zone for the distribution libraries using the DDDEF names you specified on the Build SMP/E USERMOD panel, with the names of the appropriate distribution libraries you created.
5. SMP/E RECEIVE and APPLY the USERMOD.
6. Place the load module target library in the normal MVS search order on your system or allocate it to ISPLLIB in your logon procedure.

#### **Notes:**

1. You only need to create the distribution libraries and distribution DDDEFs if you SMP/E ACCEPT the USERMOD. However, the Distribution Library DDDEF names fields on the panel must be filled in when building the USERMOD to make the MCS statements in the USERMOD complete.
2. You might need an SMPTLOAD DDDEF in your target and distribution zone. The temporary data set for the ISPCFIGU and ISPCFIGV load modules is created with DSORG(PO) UNIT(SYSALLDA) and will be either a PDS or a PDSE depending on the setup of your system. The temporary load data set is IEBCOPY unloaded to produce the ++PROGRAM elements in the USERMOD. See the *SMP/E Reference* for more information about the requirement for SMPTLOAD.

RELATED REFERENCES

Chapter 9, "ISPF Configuration Table keywords and values," on page 203

---

## Chapter 3. Customizing ISPF

---

### Allocating optional image ISPF library

The data set described in Table 1 is not a required ISPF library. You must allocate it only if an application in GUI mode uses images.

*Table 1. Image data set*

DDNAME	Description	RECFM	LRECL	BLKSIZE
ISPILIB	Image Input Library	FB	80	(See note)

**Note:** The block size can be established by the application. It must be a multiple of 80.

If you plan to use ISPF's image support, you must allocate the image input data set to ddname ISPILIB before using the images. You can accomplish this by allocating the image input data set before you invoke ISPF. Image files in the Graphics Interchange Format (GIF) should reside in this input data set. This data set must be a partitioned data set. It can be allocated with DISP=SHR.

**Note:** The LIBDEF service does *not* result in the allocation of ISPILIB.

ISPF ships sample image files in the sample library SISPSAMP. The ISPF panel ISR@PRIM uses three of the sample image files: ISPFGIFL, ISPFGIFS, and ISPEXIT. You can copy the sample image files to your own image input data set allocated to ddname ISPILIB.

---

### Customizing action bars

Many ISPF action bar items are common to multiple panels. These action bar definitions are coded in separate files that are embedded into the various panels. The embedded DTL source fields are written in Generalized Markup Language (GML), as part of the DTL panel source.

To customize one or more action bar items:

1. Update the appropriate embed file from Table 2. These embed files are distributed in the ISP.SISPGENU library.

*Table 2. Embedded DTL source files*

File name	DTL File for...
ISPDFIL1	File action bar choice
ISPFMENU	SCLM menu action bar choice
ISPDFUNC	Functions action bar choice
ISPDHUCM	Action bar choice
ISPDHYXM	Action bar choice
ISPFJOB	Jobcard action bar choice
ISPDLANG	Compilers action bar choice
ISPDMENU	Menu action bar choice
ISPDREFE	Reflist action bar choice
ISPDREFL	Reflist action bar choice
ISPDREFM	Refmode action bar choice

Table 2. Embedded DTL source files (continued)

File name	DTL File for...
ISPDSAVE	Save action bar choice
ISPFSCLM	SCLM action bar choice
ISPDTEST	Test action bar choice
ISPDUTIL	Utilities action bar choice

2. Reconvert the affected panels by specifying the appropriate list of members (Table 3) and running the ISPD TLC conversion utility.

Table 3. DTL list of panels

File name	DTL List for...
ISPFAB	All action bar GMLs
ISPFALL	All product GMLs
ISPFIL1	File action bar choice
ISPMENU1	SCLM menu action bar choice
ISPFUNC	Functions action bar choice
ISPHUCM	Action bar choice
ISPHYXM	Action bar choice
ISPJJOB	Jobcard action bar choice
ISPLANG	Compilers action bar choice
ISPMENU	Menu action bar choice
ISPREFE	Reflist action bar choice
ISPREFL	Reflist action bar choice
ISPREFM	Refmode action bar choice
ISPSAVEB	Save action bar choice
ISPSCLM	SCLM action bar choice
ISPTTEST	Test action bar choice
ISPUTIL	Utilities action bar choice

The lists of panels for different types of modifications are contained in ISP.SISPSAMP. ISPD TLC will convert all of the members in a list that are used as input member names.

## Invoking the ISPD TLC conversion utility

ISPD TLC can be run either interactively or in a batch job. Because of the number of panels to be converted, a batch job might be the best choice. The information that follows describes both interactive and batch conversions.

**Note:** After conversion, you might want to preprocess the panels to improve performance. For more information about preprocessing panels, see "Preprocess all ISPF panels" on page 69.

For more information about ISPD TLC invocation, refer to *z/OS ISPF Dialog Tag Language Guide and Reference*.

### Running interactively

To run interactively, invoke the conversion utility from any command line by entering:

```
ISPD TLC
```

Figure 41 on page 41 shows the first screen of the ISPD TLC invocation panel with the default data set names and with the required options selected.

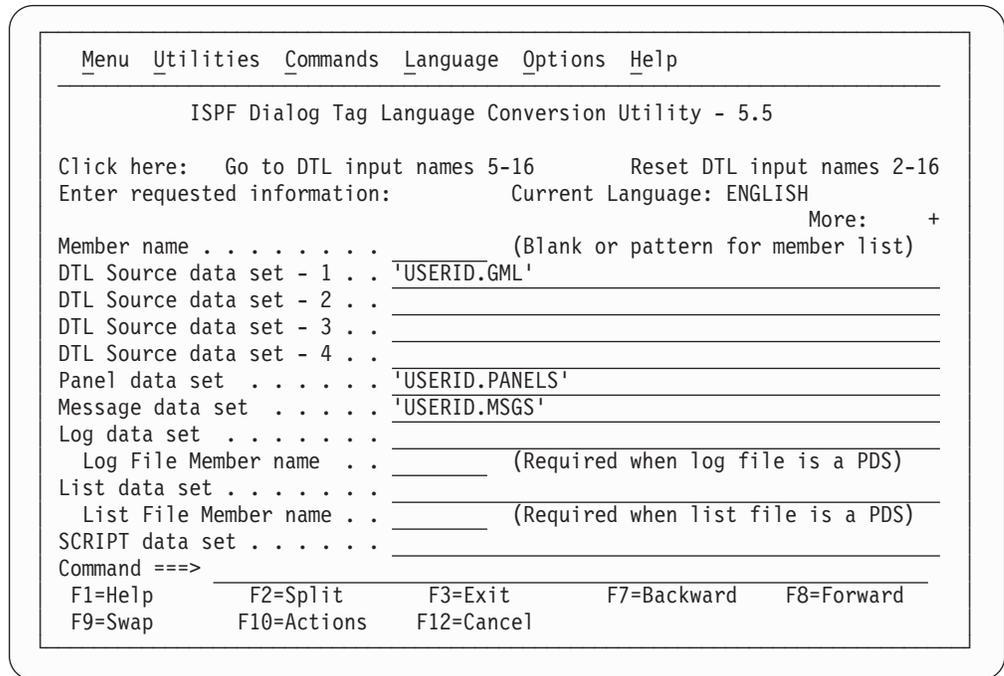


Figure 41. First ISPD TLC screen

The member name ISPMENU will cause all of the panels with the Menu action bar item to be converted.

When the interactive panel is displayed, change the data set names shown as required for your installation. Fill in the name of your input and output files where indicated. The first DTL source file should contain any local panel modifications. The second DTL source file contains the product panel source. The third DTL source file contains the conversion list member ISPMENU.

**Note:** The current language selection appears on all screens. A different language can be selected from the action bar.

After typing in your file names, scroll to the next screen (Figure 42 on page 42).

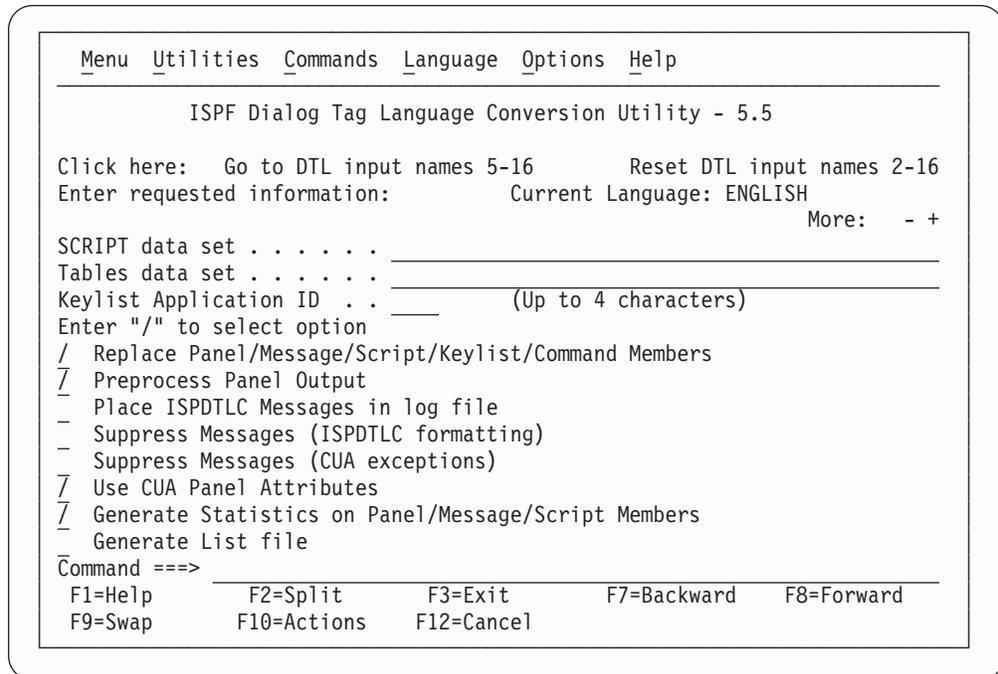


Figure 42. Second ISPD TLC screen

**Note:** To reduce the amount of screen output you can select options Suppress Messages (ISPD TLC formatting) and Suppress Messages (CUA exceptions).

Enter ISR as the Keylist Application ID and select the options as shown. Scroll to the next screen (Figure 43).

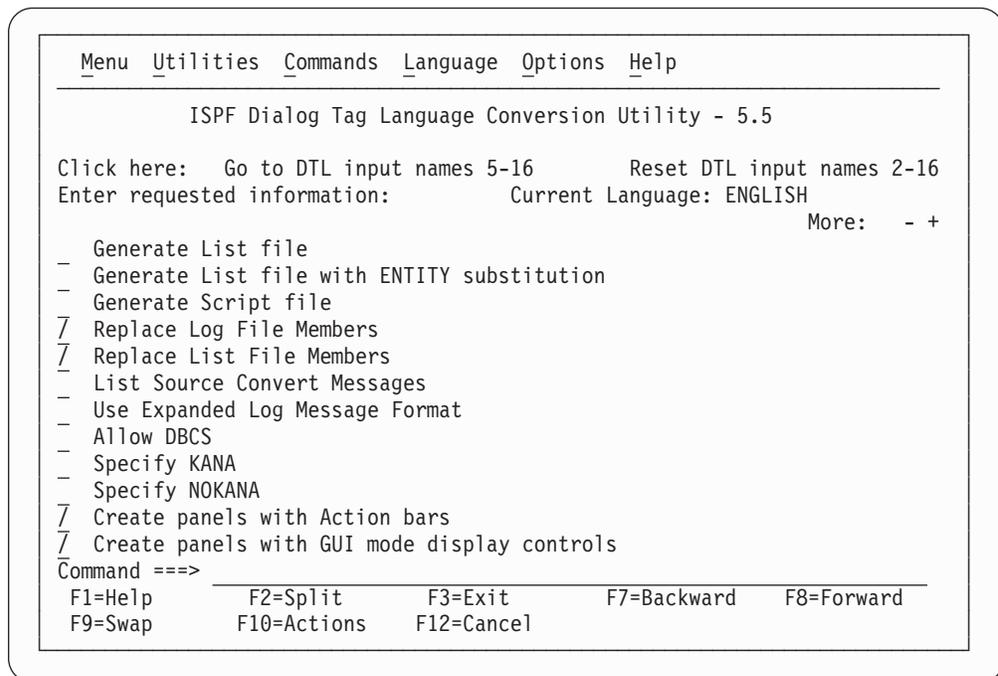


Figure 43. Third ISPD TLC screen

Scroll to the next screen (Figure 44 on page 43).

```

Menu Utilities Commands Language Options Help
-----
ISPFDial Tag Language Conversion Utility - 5.5

Click here:  Go to DTL input names 5-16      Reset DTL input names 2-16
Enter requested information:      Current Language: ENGLISH
                                          More:  - +

/ Create panels with GUI mode display controls
/ Add ISPDTLC version / timestamp to panels and messages
- Combine scrollable areas into panel )BODY section
- Display converted panels
- Display converted panels in a window
- Bypass data set name validation (after first cycle)
/ Enable graphic character display
- Use field names in place of Z variables
- Align DBCS prompt text with entry field
- Preserve leading ENTITY blanks when "space" is not specified
- Process multiple line comment blocks
- Scroll member list to last selected member
Command ==>
F1=Help      F2=Split      F3=Exit      F7=Backward  F8=Forward
F9=Swap      F10=Actions   F12=Cancel

```

Figure 44. Fourth ISPDTLC screen

Scroll to the last screen (Figure 45).

```

Menu Utilities Commands Language Options Help
-----
ISPCP01  ISPFDial Tag Language Conversion Utility - 5.5

Click here:  Go to DTL input names 5-16      Reset DTL input names 2-16
Enter requested information:      Current Language: ENGLISH
                                          More:  -

- Bypass data set name validation (after first cycle)
/ Enable graphic character display
- Use field names in place of Z variables
- Align DBCS prompt text with entry field
- Preserve leading ENTITY blanks when "space" is not specified
- Process multiple line comment blocks
- Scroll member list to last selected member
- Generate accessible character string
- Display additional DTL source data set list

Conversion status message interval . . . 1 (0 - 999)
DISPLAY(W) option check interval . . . 1 (1 - 99)
Command ==>
F1=Help      F2=Split      F3=Exit      F7=Backward  F8=Forward
F9=Swap      F10=Actions   F12=Cancel

```

Figure 45. Fifth ISPDTLC screen

**Note:** ISPF product panels require that the options “Preserve leading ENTITY blanks when “space” is not specified” and “Process multiple line comment blocks” be selected as shown in the figures here.

If a double byte language conversion is in process, the DBCS and KANA options might be required.

If you are recompiling the English version of the panels, a second conversion should be made for the uppercased English panels. You will need to enter a different output panel library name and select the UPPERENG language from the action bar. All of the other panel information and options should be the same as the first conversion.

## Running in batch mode

The batch conversion requires a profile data set that contains ddnames and data set names for the input and output files. The sample JCL shown in Figure 46 refers to the profile data set. The ISPDTLC invocation syntax specifies all of the required options for the conversion.

Change the data set names shown in both the profile and the JCL as required for your location.

```
//your jobcard here
//CONVERT EXEC PGM=IKJEFT01,DYNAMNBR=50
//SYSPRINT DD SYSOUT=*
//SYSPROC DD DISP=SHR,DSN=your.exec.dataset
//SYSEXEC DD DISP=SHR,DSN=ISP.SISPEXEC
//STEPLIB DD DISP=SHR,DSN=ISP.SISPLD
// DD DISP=SHR,DSN=ISP.SISPLPA
//ISPLLIB DD DISP=SHR,DSN=ISP.SISPLD
// DD DISP=SHR,DSN=ISP.SISPLPA
//ISPLIB DD DISP=SHR,DSN=ISP.SISPLD
//ISPLIB DD DISP=SHR,DSN=ISP.SISPLPA
//ISPMLIB DD DISP=SHR,DSN=ISP.SISPMENU
//ISPPLIB DD DISP=SHR,DSN=ISP.SISPPENU
//ISPTLIB DD DISP=SHR,DSN=your.profile.dataset
// DD DISP=SHR,DSN=ISP.SISPTENU
//ISPSLIB DD DISP=SHR,DSN=ISP.SISPSENU
//ISPTABL DD DISP=OLD,DSN=your.profile.dataset
//ISPPROF DD DISP=OLD,DSN=your.profile.dataset
//ISPLLOG DD DISP=OLD,DSN=your.log.dataset
//ISPLIST DD DISP=OLD,DSN=your.list.dataset
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
        PROFILE PREFIX(USERAA)
        ISPSTRT CMD(CONVACTB)
```

Figure 46. Sample JCL for Batch conversion

The CONVACTB EXEC (written in REXX) is:

```
'ISPDTLC ISPMENU (DISK KEYLAPPL=ISR MSGSUPP CUAATTR CUASUPP PLEB MCOMMENT
        NOSTATS NOACTBAR PROFILE=your.gml.dataset(profile)'
```

This EXEC should be placed on *your.exec.dataset*.

### Notes:

1. If you are recompiling the English version of the panels, you must run ISPDTLC two times. The second run is to create the uppercase English version of the panel. The UPPERENG language keyword is specified to create the uppercase panel version. A different profile is also used with a different panel library defined to DTLPAN.

```
'ISPDTLC ISPMENU (DISK KEYLAPPL=ISR MSGSUPP CUAATTR CUASUPP PLEB MCOMMENT
        UPPERENG NOSTATS NOACTBAR PROFILE=your.gml.dataset(profileu)'
```

2. If you are recompiling DBCS versions of the panels, you must add the DBCS option to the command syntax. Japanese panels require the KANA option as well.

```
'ISPDTLC ISPFALL (DISK KEYLAPPL=ISR MSGSUPP CUAATTR CUASUPP PLEB MCOMMENT
        JAPANESE DBCS KANA NOSTATS NOACTBAR PROFILE=your.gml.dataset(profile)'
```

*your.gml.dataset(profile)* should include:

<b>DTLGML</b>	<i>your.appl.gml.dataset</i>
<b>DTLGML</b>	ISP.SISPGENU
<b>DTLGML</b>	ISP.SISPGMLI
<b>DTLGML</b>	ISP.SISPSAMP
<b>DTLPAN</b>	<i>your.output.panel.dataset</i>
<b>DTLMSG</b>	<i>your.output.message.dataset</i>
<b>DTLLOG</b>	<i>your.log.dataset</i>

*your.gml.dataset(profileu)* for creating uppercased English panels should include:

<b>DTLGML</b>	<i>your.appl.gml.dataset</i>
<b>DTLGML</b>	ISP.SISPGENU
<b>DTLGML</b>	ISP.SISPGMLI
<b>DTLGML</b>	ISP.SISPSAMP
<b>DTLPAN</b>	<i>your.output.uppercase.panel.dataset</i>
<b>DTLMSG</b>	<i>your.output.message.dataset</i>
<b>DTLLOG</b>	<i>your.log.dataset</i>

Your updated GML files should be stored in *your.appl.gml.dataset*. The converted files will be in *your.output.panel.dataset* or *your.output.uppercase.panel.dataset*, and the messages generated will be in *your.log.dataset*.

The log file will contain the results of the conversion processing. Most of the messages generated by ISPD TLC are related to ISPF extensions to the Dialog Tag Language and have been suppressed. The expected message number from compiling the panel source will appear in a comment near the top of the DTL source file for that panel source.

---

## Customizing for profile sharing

Customizing for profile sharing shows how you can customize shared profiles within ISPF.

**Note:** Profile sharing is intended to be used by the same user logging concurrently onto multiple systems. The contents of the ISPF profile are often user-specific.

## Customizing the ISPF Configuration utility

You can customize the ISPF Configuration utility with these configuration options:

- PROFILE\_SHARING
- PROFILE\_ENQLOCK\_WAIT
- PROFILE\_ENQLOCK\_RETRY\_COUNT
- PROFILE\_ENQLOCK\_PROMPT
- PROFILE\_SYSPROF\_CONFLICT
- PROFILE\_ISPPROF\_CONFLICT
- PROFILE\_APPPROF\_CONFLICT
- PROFILE\_REFLIST\_CONFLICT
- PROFILE\_EDIT\_CONFLICT
- PROFILE\_BATCH\_CONFLICT
- PROFILE\_OTHER\_CONFLICT
- RESET\_PROFILE\_SHARING\_SETTINGS

For further details about configuration options, see “ISPF site-wide profile customizations” on page 217.

## Temporary data set names

The ISPF Configuration utility provides an option to specify an additional qualifier for ISPF temporary data sets, including Log, List and temporary control and work data sets. The use of this qualifier also includes trace data sets from ISPVCALL, ISPDPTRC and ISPFTRC. When Profile Sharing has been enabled and no value has been specified for the additional qualifier, a value of "ISP&SEQ" is used, where &SEQ is a system symbolic variable that has the same value as defined by the ISPF dialog variable ZSEQ.

## SYSIKJUA enqueue

To enable the ISPF multi-logon support in a sysplex, you must comment out or remove the RNLDEF statement specifying SYSIKJUA in the GRSNLxx member of the system parmlib concatenation:

```
/* RNLDEF RNL(INCL) TYPE(GENERIC) */
/* QNAME(SYSIKJUA) */
```

## Limitation of other products

These limitations apply with shared profiles:

### TSO/E Notifications

Job notifications cannot be sent to the system from which they were submitted.

### TSO/E Reconnect

Reconnect support ensures that a user's address space is retained for a specified reconnect interval after a TSO user loses connectivity to the TSO system. During this interval, if you try to reconnect, you are automatically reconnected to the original system, application, and address space.

### SDSF ULOG Console

In a sysplex, the responses for MVS and JES2 commands are only seen on the first system that activates an EMCS console. By default, SDSF creates an EMCS console name that is the same as your userid. You can change the console name used by SDSF using the SET CONSOLE (*console-name*) command.

**Note:** Other ISPF applications may experience unexpected results when they attempt to use the ISPF profile data set to store application specific information.

---

## TSO/ISPF client gateway

ISPF provides an interface or gateway for client applications to connect to the z/OS host and run TSO and ISPF commands. This section describes how the gateway works, how to install and customize the interface, and how client applications can use the interface.

## Gateway description

The gateway runs within z/OS UNIX and is invoked upon a request for a TSO or ISPF command from the client application. To run a requested command, the gateway establishes a TSO address space and, if the request is for an ISPF command, data sets are allocated and an ISPF session is also initialized. After completion of the command, the results are returned to the client.

If requested by the client, the gateway maintains the state of the TSO/ISPF session for any subsequent client function requests for the user. Otherwise, the interface terminates the TSO/ISPF session. The advantages of maintaining a user's TSO/ISPF session on z/OS are in bypassing the overhead of having to establish a new session for every TSO or ISPF command request and in adding an inherent ability to maintain state data between calls from the client.

The connection between the client and gateway may be by means of the HTTP protocol (web-based services), direct TCP/IP socket connections, or any other means through which data and commands can be exchanged between the client and z/OS host.

**Note:** ISPF commands invoked by means of the gateway run in an ISPF batch environment.

## Installing and customizing the gateway

This section describes installing and customizing the gateway.

### Installation considerations

Consider these points before you configure your system:

- A RACF OMVS segment (or equivalent) that specifies a valid non-zero uid, home directory, and shell command must be defined for each user of the interface.
- Set MAXPROCUSER in BPXPRMxx parmlib member to a minimum of 50. This can be checked and set dynamically (until the next IPL) with the following commands (as described in *z/OS MVS System Commands*, SA22-7627):

```
DISPLAY OMVS,0
SETOMVS MAXPROCUSER=50.
```

Setting a value that is too low can cause the interface to fail.

- Table 4 describes the load modules that comprise the gateway:

*Table 4. Load modules comprising the gateway*

Name	Library	Description
ISPZINT	ISP.SISPLPA	Gateway initialization - manages spawned TSO sessions, routes command requests to these sessions.
ISPZTSO	SYS1.LINKLIB	TSO initialization - attaches a TSO session to run a command. ISPZTSO must run APF-authorized.
ISPZCNT	ISP.SISPLOAD	ISPF initialization - allocates data sets and starts an ISPF session.
ISPZCMD	ISP.SISPLOAD	ISPF command interface - invokes an ISPF command and, if required, keeps the ISPF session active.
ISPZTMO	ISP.SISPLOAD	Reusable session time-out processor.

These load modules must be available to run under the server used to communicate between the client and z/OS host.

A PROGRAM class profile that identifies ISPZINT as being a controlled program should be defined using the command:

```
RDEFINE PROGRAM ISPZINT ADDMEM('ISP.SISPLPA'//NOPADCHK) UACC(READ)
```

(Refer to your security server's documentation for details.) Failure to define ISPZINT as being controlled might result in messages, such as CSV042I or ICH422I, being issued.

- Table 5 describes the gateway files installed into the z/OS UNIX file system:

*Table 5. Gateway files installed into the z/OS UNIX file system*

Name	Description
/usr/lpp/ispf/bin/ISPZINT	A stub file with the sticky bit set on to enable invocation of the gateway initialization load module ISPZINT.
/usr/lpp/ispf/bin/ISPZTSO	A stub file with the sticky bit set on to enable invocation of the TSO initialization load module ISPZTSO.
/usr/lpp/ispf/bin/ISPZXENV	A REXX routine called by the gateway XML API routine, ISPZXML, to set up environment variables.
/usr/lpp/ispf/bin/ISPZXML	A REXX routine that is the XML API for the gateway.

## Installing the gateway

To install the gateway, customize and run the JCL in the member ISPZINS1 in the ISPF samples data set ISP.SISPSAMP. Customize the JCL according to the instructions in the member.

This job performs these tasks:

- Creates CONFIG and WORKAREA directories in the z/OS UNIX file system at the directory you specify.
- Copies the sample ISPF configuration table from member ISPZISPC in the samples data set ISP.SISPSAMP to file ISPF.conf in the CONFIG directory. This file requires customization.

The recommended base directory for the configuration files is /etc/ispf. The part of the directory up to ISPF must exist before running this job.

You must have read and write access to the WORKAREA directory /var/ispf/WORKAREA. The WORKAREA is used for the transfer of files. Temporary directories of the format /var/ispf/WORKAREA/userid/\* are created during use of the interface.

**Note:** Some temporary session files might be created in the /tmp directory. Ensure all users have write access to the /tmp directory.

The interface removes any temporary files it creates in the WORKAREA directory. However, temporary output is sometimes left over, for example, if there is a communication error while processing. For this reason, it is recommended that you clear out the WORKAREA directory from time to time. To do this, use these commands in OMVS:

```
cd /var/ispf/WORKAREA
rm -r *
```

Where /var/ispf/WORKAREA depends on where you create the WORKAREA directory.

## Customizing the gateway

You must customize the ISPF configuration file `ISPF.conf` to host site requirements for ISPF data set allocation. This file is stored, by default, in directory `/etc/ispf`. The provided sample `ISPF.conf` has instructions to complete customization so your user site can:

- Include the minimum ISPF data set allocations for the gateway to operate. This means allocating the minimum ISPF data sets required to initialize an ISPF session. In the sample provided, these are specified as the `isp.sisp*` data sets. You might need to change these for your site-specified data set names.
- Add additional DDNAME file allocations or concatenate additional ISPF data sets.
- Launch a customer-defined allocation executable (`exec`) to provide further data set allocation by user ID. A sample `exec` is provided in member `ISPZISP2` in the ISPF samples data set `ISP.SISPSAMP`.

The allocations for each of the ISPF DDs must be specified on a single line with each data set separated by a comma. Comment lines can be added by beginning the line with an asterisk (\*). Here is a sample `ISPF.conf`:

```
* REQUIRED:
* Below is the minimum requirements for ISPF allocation.
* Change the default ISPF data set names below to match your
* host site.
* Add additional dsn concatenations on same line and separate
* by comma.
* Order of data sets listed is search order in concatenation.
*
sysproc=ISP.SISPCLIB
sysexec=SYS1.LOCAL.EXEC,ISP.SISPEXEC
ispmlib=ISP.SISPMENU
isptlib=ISP.SISPTEU
ispplib=ISP.SISPPENU
ispslib=ISP.SISPSLIB
ispllib=ISP.SISPLOAD,SYS1.LOCAL.LOAD
```

The `ISPF_timeout` option in the ISPF configuration file can be used to specify a time out value for reusable ("stateful") ISPF sessions. The time out value specifies the number of seconds a user's ISPF session can remain idle between service call requests. If the idle time exceeds the time out value, the session is terminated and the next service request for the user results in a new ISPF session being established. The default reusable ISPF session time out value is 900 seconds (15 minutes).

Here is an example of setting a time out value of 300 seconds (5 minutes):

```
ISPF_timeout = 300
```

## Customizing the HTTP server

If you plan to use the HTTP server to invoke the gateway, these changes must be made to the HTTP environment and configuration files:

- The HTTP environment file, `httpd.env`, must be updated to include directives that identify the workarea and configuration directories for the gateway. The `CGI_ISPWORK` directive is used to define the path for the `WORKAREA` directory used by the gateway. The `CGI_ISPCONF` directive is used to define the path for the `CONFIG` directory where the ISPF configuration file `ISPF.conf` is stored. Shown here are examples of the `CGI_ISPWORK` and `CGI_ISPCONF` directives specifying the default paths for the `WORKAREA` and `CONFIG` directories:

```
CGI_ISPWORK=/var/ispf
CGI_ISPCONF=/etc/ispf
```

- These exec directives must be added into the HTTP configuration file `httpd.conf`:

```
Exec          /ISPZINT      /usr/lpp/ispf/bin/ISPZINT
Exec          /ISPXML      /usr/lpp/ispf/bin/ISPXML
```

**Note:** The path specified for these directives must be the path of the directory where the gateway was installed.

If the gateway modules (see Table 4 on page 47) are not in the LINKLIST, then update the HTTP server STEPLIB DD to include the load library data sets containing these modules.

**Note:** To use the gateway, the client is required to supply a valid z/OS user ID and password.

For additional information about configuring IBM HTTP Web servers, review these IBM manuals:

- *HTTP Server Planning, Installing and Using*, (SC31-8690)
- *OS/390 e-business Infrastructure: IBM HTTP Server V5.1 for OS/390*, (SG24-5603-00)

## Customizing Remote Systems Explorer

If you plan to use IBM Rational Developer for System z's Remote Systems Explorer (RSE) to invoke the gateway, the RSE environment file must be customized. The environment file for your RSE connection is `rse.envvars` which, by default, is in `/etc/rdz`.

These environment variables used by the gateway should already have been defined in `rse.envvars` during base RSE customization:

```
_CMDSERV_BASE_HOME
_CMDSERV_CONF_HOME
_CMDSERV_WORK_HOME
```

These following changes are required to the `rse.envvars` file:

- If the gateway modules (see Table 4 on page 47) are not in the LINKLIST, then update the STEPLIB environment variable value to include the load library data sets containing these modules.

For additional information about installing and configuring RSE, review these IBM documents:

- *Program Directory for IBM Rational Developer for System z* (GI11-8298-00)
- *IBM Rational Developer for System z Host Configuration Guide* (SC31-6930-02)

## Configuration for other environments

If you plan to invoke the gateway by means other than through the HTTP server of RSE (for example, from the OMVS shell), configuration is required to ensure values are set for environment variables used by the gateway. This requires the environment set-up routine, `ISPZXENV`, to be modified to set values for the environment variables shown in Table 6 on page 51.

Table 6. Environment variables to be set by the environment set-up routine, *ISPZXENV*

Environment variable	Description
STEPLIB	Specifies the STEPLIB data sets from where MVS load modules are run. This environment variable must be set to the names of the load library data sets containing the gateway modules (see Table 4 on page 47) if these modules are not in the LINKLIST.
CGI_ISPCONF	Specifies the CONFIG directory path where the configuration files are stored.
CGI_ISPWORK	Specifies the WORKAREA directory path that is used for temporary files.

A sample of this REXX routine is provided in member *ISPZXENV* in the ISPF samples data set *ISP.SISPSAMP*. These lines in *ISPZXENV* may require modification:

```
STEPLIB = 'ISP.SISPLPA:ISP.SISLOAD'
```

The *STEPLIB* variable should specify the data sets containing the load modules for the TSO/ISPF client gateway (see Table 4 on page 47). If these load modules reside in the LINKLIST, then the *STEPLIB* variable should be set to a null value (that is, '').

```
CGI_ISPCONF = '/etc/ispf'
```

The *CGI\_ISPCONF* variable specifies the HOME directory path where the configuration files reside for the gateway. For more information on the CONFIG directory, see "Installing the gateway" on page 48.

```
CGI_ISPWORK = '/var/ispf'
```

The *CGI\_ISPWORK* variable specifies the pathname of the directory where work files for the gateway are stored. For more information on the work directory, see "Installing the gateway" on page 48.

Replace the version of *ISPZXENV* in *usr/lpp/ispf/bin* with your modified version of *ISPZXENV*.

## Using the gateway

This section describes how to provide TSO/ISPF service and command requests to the gateway and the format of the output returned by the gateway.

### Passing requests to the gateway

TSO/ISPF service and command requests are passed to the gateway in XML format. The XML schema shown here, which is supplied in member *ISPZXSDI* in the ISPF samples data set *ISP.SISPSAMP*, describes the format and can be used to validate the XML for a service request:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="ISPF-INPUT">
    <xs:complexType>
      <xs:all>

        <xs:element name="SERVICE-REQUEST">
          <xs:complexType>
            <xs:all>

              <xs:element name="service">
                <xs:simpleType>
```

```

        <xs:restriction base="xs:string">
          <!-- Specifies native TSO or ISPF service call -->
          <xs:enumeration value="ISPF"/>
          <xs:enumeration value="TSO"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>

    <xs:element name="session" minOccurs="0">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <!-- Default NONE : Session terminates after service call -->
          <xs:enumeration value="NONE"/>
          <!-- Reuseable ISPF session stays active between calls -->
          <xs:enumeration value="REUSE"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>

    <!-- Use existing ISPF profile in call -->
    <xs:element name="ispprof" type="xs:string" minOccurs="0"/>

    <!-- Free form TSO/ISPF command -->
    <xs:element name="command" type="xs:string"/>

    </xs:all>
  </xs:complexType>
</xs:element>

</xs:all>
</xs:complexType>
</xs:element>

</xs:schema>

```

This is an example of the XML to request a TSO LISTCAT command:

```

<?xml version="1.0"?>
<ISPF-INPUT>
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="ispf.xsd"
<SERVICE-REQUEST>
<service>TSO</service>
<session>NONE</session>
<command>LISTC ENT('SYS1.LINKLIB')</command>
</SERVICE-REQUEST>
</ISPF-INPUT>

```

This is an example of the XML to request to run REXX program DINFO in a "reusable" ISPF session which remains active to run subsequent commands for this user:

```

<?xml version="1.0"?>
<ISPF-INPUT>
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="ispf.xsd"
<SERVICE-REQUEST>
<service>ISPF</service>
<session>REUSE</session>
<command>TSO DINFO sys1.linklib</command>
<ispprof>USER.ISPPROF</ispprof>
</SERVICE-REQUEST>
</ISPF-INPUT>

```

This shows the REXX program DINFO:

```

/* REXX */
parse upper arg dsn .
address "ISPEXEC" "dsinfo dataset('"dsn"')"
if rc = 0 then do
  say 'Volume           = ' zdsvol
  say 'Primary allocation = ' strip(zds1ex) zdsspc
  say 'Secondary allocation = ' strip(zds2ex) zds2spc
end
else do
  say 'DSINFO rc = ' rc
  say 'ZERRMSG   = ' zerrmsg
end
exit rc

```

The Java program shown here, which is supplied in member ISPZXJAV in the ISPF samples data set ISP.SISPSAMP, shows an example of invoking the interface through a HTTP server and passing XML to request running the TSO PROFILE command:

```

package com.ibm.ispfcall;

import java.io.*;
import java.net.*;
import java.util.*;

public class XmlIspf {

    public static void main(String[] args) {

        try {

            BufferedReader in = new BufferedReader(new FileReader("C:\\logon.txt"));
            String logon = in.readLine();
            in.close();

            Date d = new Date() ;
            System.out.println("START Transfer DATE/TIME is :" + d.toString() );

            // URL details for CGI POST
            URL url = new URL("http", "SITE.COM", 1234, "/ISPZXML");
            HttpURLConnection con = (HttpURLConnection) url.openConnection();

            con.setUseCaches(false);
            con.setDoInput(true);
            con.setDoOutput(true);
            con.setRequestMethod("POST");
            con.setRequestProperty( "Authorization", "Basic "
                + Base64Encoder.encode( logon ));

            System.out.println("At url openConnection.. ");

            // POST CGI routines
            doPut(url, con, "POST", "");
            doGet(url, con);

            Date c = new Date() ;
            System.out.println("TOTAL Completion DATE/TIME is :" + c.toString() );

        }
        catch (IOException exception)
        {
            System.out.println("Error: " + exception);
        }
    }

    public static void doPut(URL url, HttpURLConnection con, String ISPFFUNC,
        String fileIn) throws IOException

```

```

{
  if (ISPF_FUNC.equals("POST"))
  {
    PrintWriter out = new PrintWriter(con.getOutputStream());
    // Below is a sample inline XML input for an ISPF service request
    // This could alternatively be read from an external file
    out.println( "<?xml version='1.0'?" );
    out.println( "<ISPF-INPUT>" );
    out.println( "xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'" );
    out.println( "xsi:noNamespaceSchemaLocation='ispf.xsd'" );
    out.println( "<SERVICE-REQUEST>" );
    out.println( "<service>ISPF</service>" );
    out.println( "<session>NONE</session>" );
    out.println( "<command>TSO PROFILE</command>" );
    out.println( "<ispprof>USERID.ISPPROF</ispprof>" );
    out.println( "</SERVICE-REQUEST>" );
    out.println( "</ISPF-INPUT>" );
    out.flush();
    out.close();
  }
}

public static void doGet(URL url, HttpURLConnection con) throws IOException
{
  BufferedReader in;
  try
  {
    System.out.println("About to accept response from Server");
    in = new BufferedReader(new InputStreamReader(con.getInputStream()));
    System.out.println("Response from Server received");
  }
  catch (FileNotFoundException exception)
  {
    InputStream err = ((HttpURLConnection)con).getErrorStream();
    if (err == null) throw exception ;
    in = new BufferedReader(new InputStreamReader(err));
  }

  String line;
  while ((line = in.readLine()) != null)
    System.out.println(line);

  in.close();
}

```

## Receiving output from the gateway

The output from a service request is returned by the gateway to the client in XML format. The XML schema shown here, which is supplied in member ISPZXSDO in the ISPF samples data set ISP.SISPSAMP, describes the format and can be used to process the XML returned for a service request:

```

<?xml version="1.0" encoding="ISO-8859-1" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="ISPF-OUTPUT">
    <xs:complexType>
      <xs:all>

        <xs:element name="SERVICE-REQUEST"/>

        <xs:element name="SERVICE-RESPONSE">
          <xs:complexType>
            <xs:all>
              <xs:element name="ISPF-COMMAND"/>
              <xs:element name="RETURN-CODE"/>
              <xs:element name="ISPF"/>
            </xs:all>
          </xs:complexType>
        </xs:element>
      </xs:all>
    </xs:complexType>
  </xs:element>

```

```

    </xs:all>
  </xs:complexType>
</xs:element>

  <xs:element name="OPERATIONS-LOG"/>

</xs:all>
</xs:complexType>
</xs:element>

</xs:schema>

```

The following shows the XML returned for the TSO LISTCAT command above:

```

<?xml version="1.0"?>
<ISPF-OUTPUT>
  <SERVICE-REQUEST>
    <service>TSO</service>
    <session>NONE</session>
    <command>LISTC ENT('SYS1.LINKLIB')</command>
  </SERVICE-REQUEST>
  <SERVICE-RESPONSE>
    <ISPF-COMMAND>

</ISPF-COMMAND>
<![CDATA[
  NONVSAM ----- SYS1.LINKLIB
          IN-CAT --- CATALOG.MASTER.SYSPLEXD
  READY
  END
]]>
</TSO>
</SERVICE-RESPONSE>
<OPERATIONS-LOG>
<![CDATA[
  Content-type: text/plain

  Entering ISPZINT (Service initialization)
  About to read from fileno(stdin) = 0
  Data read from STDIN is TSO LISTC ENT('SYS1.LINKLIB')
  EPOCH secs = 1202109526
  Local Date & time: Mon Feb  4 02:18:46 2008
  Hour: 2 Min: 18 Sec 46
  Function ID timestamp = ID008326
  Environment variables:
  0 QUERY_STRING=
  1 CONTENT_TYPE=application/x-www-form-urlencoded
  2 PATH=/bin:./usr/sbin:/usr/lpp/internet/bin:/usr/lpp/internet/sbin:
  /usr/lpp/products/java142/J1.4/bin/
  3 AUTH_TYPE=Basic
  4 DOCUMENT_URI=/ISPZXML
  5 SHELL=/bin/sh
  6 HTTPS=OFF
  7 HTTP_ACCEPT=text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2
  8 HTTP_USER_AGENT=Java1.3.1
  9 SERVER_PORT=1726
  10 RULE_FILE=//DD:CONF
  11 GATEWAY_INTERFACE=CGI/1.1
  12 PATH_INFO=
  13 CONTENT_LENGTH=274
  14 _CEE_RUNOPTS=ENVAR("_CEE_ENVFILE=//DD:ENV")
  15 _BPX_SPAWN_SCRIPT=YES
  16 REFERER_URL=
  17 _=./ISPZINT
  18 CLASSPATH=./usr/lpp/internet/server_root/CAServlet
  19 STEPLIB=CURRENT
  20 REQUEST_METHOD=POST

```

```

21 REMOTE_ADDR=9.190.236.239
22 LANG=C
23 LIBPATH=/bin:/usr/lpp/internet/bin:/usr/lpp/internet/sbin
24 REMOTE_USER=USERNME
25 SERVER_ADDR=192.168.123.11
26 FSCP=IBM-1047
27 PATH_TRANSLATED=
28 HTTP_CONNECTION=keep-alive
29 SERVER_TOKEN=1
30 HTTP_HOST=PTHISD1.AU.IBM.COM
31 _BPX_SHAREAS=YES
32 CGI_ISPCONF=/etc/ispf
33 SERVER_SOFTWARE=IBM HTTP Server/V5R3M0
34 REPORTBITS=77
35 DOCUMENT_ROOT=usr/lpp/ispf/bin
36 NETCP=ISO8859-1
37 CGI_ISPWORK=/var/ispf
38 COUNTERDIR=NULL
39 LC_ALL=en_US.IBM-1047
40 CGI_DTCONF=/etc/ispf
41 _BPX_USERID=USERNME
42 SERVER_PROTOCOL=HTTP/1.1
43 JAVA_HOME=/usr/lpp/products/java142/J1.4/
44 HTTPS_KEYSIZE=
45 TZ=EST5EDT
46 _CEE_ENVFILE=//DD:ENV
47 _BPX_BATCH_SPAWN=SPAWN
48 SCRIPT_NAME=/ISPZXML
49 NLSPATH=/usr/lib/nls/msg/%L/%N:/usr/lpp/internet/%L/%N:
/usr/lib/nls/msg/En_US.IBM-1047/%N
50 CGI_DTWORK=/var/ispf
51 DOCUMENT_NAME=usr/lpp/ispf/bin/ISPZXML
52 SERVER_NAME=PTHISD1.AU.IBM.COM
Number of environment variables is 53
Protocol = HTTP
FSCP = IBM-1047
NETCP = ISO8859-1
CGI_ISPCONF = /etc/ispf
CGI_ISPWORK = /var/ispf
CGI_TRANTABLE =
Server PATH = /bin:./usr/sbin:/usr/lpp/internet/bin:
/usr/lpp/internet/sbin:/usr/lpp/products/java142/J1.4/bin/
About to spawn task for ISPZTS0
Parameters passed to ISPZTS0 - LISTC ENT('SYS1.LINKLIB')
Return code from ISPZTS0 is 0
About to open /tmp/USERNME.ID008326.SYSTSPRT
*** XML-NOTE *** Reference tagged SERVICE-RESPONSE
OUT.84
]]>
</OPERATIONS-LOG>
<ISPF-OUTPUT>

```

Shown here is the XML returned from running the DINFO REXX program in a "reusable" ISPF session (see above):

```

<?xml version="1.0"?>
<ISPF-OUTPUT>
  <SERVICE-REQUEST>
    <service>ISPF</service>
    <session>REUSE</session>
    <command>TSO DINFO sys1.linklib</command>
    <ispprof>USERNME.TEST.ISPPROF</ispprof>
  </SERVICE-REQUEST>
  <SERVICE-RESPONSE>
    <ISPF-COMMAND>
      SELECT CMD(DINFO sys1.linklib) NEST
    </ISPF-COMMAND>

```

```

<RETURN-CODE>0</RETURN-CODE>
<ISPF>
<![CDATA[
  Volume           = $$SR86
  Primary allocation = 2398 BLOCK
  Secondday allocation = 0 BLOCK
]]>
</ISPF>
</SERVICE-RESPONSE>
<OPERATIONS-LOG>
<![CDATA[
Content-type: text/plain

Entering ISPZINT (Service initialization)
About to read from fileno(stdin) = 0
Data read from STDIN is ISPF TSO DINFO sys1.linklib&SESSION=SPAWN
&ISPPROF=USERNME.TEST.ISPPROF
EPOCH secs = 1202113932
Local Date & time: Mon Feb  4 03:32:12 2008
Hour: 3 Min: 32 Sec 12
Function ID timestamp = ID012732
Environment variables:
0 QUERY_STRING=
1 CONTENT_TYPE=application/x-www-form-urlencoded
2 PATH=/bin:./usr/sbin:/usr/lpp/internet/bin:/usr/lpp/internet/sbin:
  /usr/lpp/products/java142/J1.4/bin/
3 AUTH_TYPE=Basic
4 DOCUMENT_URI=/ISPZXML
5 SHELL=/bin/sh
6 HTTPS=OFF
7 HTTP_ACCEPT=text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2
8 HTTP_USER_AGENT=Java1.3.1
9 SERVER_PORT=1726
10 RULE_FILE=//DD:CONF
11 GATEWAY_INTERFACE=CGI/1.1
12 PATH_INFO=
13 CONTENT_LENGTH=313
14 _CEE_RUNOPTS=ENVAR("_CEE_ENVFILE=//DD:ENV")
15 _BPX_SPAWN_SCRIPT=YES
16 REFERER_URL=
17 =./ISPZINT
18 CLASSPATH=./usr/lpp/internet/server_root/CAServlet
19 STEPLIB=CURRENT
20 REQUEST_METHOD=POST
21 REMOTE_ADDR=9.190.236.239
22 LANG=C
23 LIBPATH=/bin:/usr/lpp/internet/bin:/usr/lpp/internet/sbin
24 REMOTE_USER=USERNME
25 SERVER_ADDR=192.168.123.11
26 FSCP=IBM-1047
27 PATH_TRANSLATED=
28 HTTP_CONNECTION=keep-alive
29 SERVER_TOKEN=1
30 HTTP_HOST=PTHISD1.AU.IBM.COM
31 _BPX_SHAREAS=YES
32 CGI_ISPCONF=/etc/ispf
33 SERVER_SOFTWARE=IBM HTTP Server/V5R3M0
34 REPORTBITS=77
35 DOCUMENT_ROOT=usr/lpp/ispf/bin
36 NETCP=ISO8859-1
37 CGI_ISPWORK=/var/ispf
38 COUNTERDIR=NULL
39 LC_ALL=en_US.IBM-1047
40 CGI_DTCONF=/etc/ispf
41 _BPX_USERID=USERNME
42 SERVER_PROTOCOL=HTTP/1.1
43 JAVA_HOME=/usr/lpp/products/java142/J1.4/

```

```

44 HTTPS_KEYSIZE=
45 TZ=EST5EDT
46 _CEE_ENVFILE=//DD:ENV
47 _BPX_BATCH_SPAWN=SPAWN
48 SCRIPT_NAME=/ISPZXML
49 NLSPATH=/usr/lib/nls/msg/%L/%N:/usr/lpp/internet/%L/%N:
/usr/lib/nls/msg/En_US.
IBM-1047/%N
50 CGI_DTWORK=/var/ispf
51 DOCUMENT_NAME=/usr/lpp/ispf/bin/ISPZXML
52 SERVER_NAME=PTHISD1.AU.IBM.COM
Number of environment variables is 53
Protocol = HTTP
FSCP = IBM-1047
NETCP = ISO8859-1
CGI_ISPCONF = /etc/ispf
CGI_ISPWORK = /var/ispf
CGI_TRANTABLE =
Server PATH = /bin.:/usr/sbin:/usr/lpp/internet/bin:/usr/lpp/internet/sbin:
/usr/lpp/products/java142/J1.4/bin/
ISPF standalone function invoked
&ISPPROF value = USERNME.TEST.ISPPROF
ISPF COMMAND = ISPF TSO DINFO sys1.linklib
ISPF PROFILE = USERNME.TEST.ISPPROF
Re-usable ISPF session = SPAWN
About to spawn task for ISPZTSO
Parameters passed to ISPZTSO - PROFILE
Return code from ISPZTSO is 0
About to process PROFILE data in /tmp/USERNME.ID012732.SYSTSPRT
About to malloc() 252 bytes for profdat
*** PROFILE data: 1IKJ56688I CHAR(0) LINE(0) PROMPT INTERCOM
NOPAUSE MSGID MODE
    WTPMSG NORECOVER PREFIX(USERNME) PLANGUAGE(ENU) SLANGUAG E(ENU)
    VARSTORAGE(LOW)
    IKJ56689I
    DEFAULT LINE/CHARACTER DELETE CHARACTERS IN EFFECT FOR THIS TERMINAL READY END
Temporary data set prefix set to : USERNME
About to call bpxwdyn to allocate VCMTEMP
Allocating data set USERNME.SCLMDT.VCMISPF.ID012732 to the VCMTEMP DD
1024 bytes of ISPF TSO DINFO sys1.linklib written to VCMTEMP
1024 bytes of /etc/SCLMDTIS;/var/ispf written to VCMTEMP
1024 bytes of /bin.:/usr/sbin:/usr/lpp/internet/bin:/usr/lpp/internet/sbin:
/usr/lpp/products/java142/J1.4/bin/~NONE written to VCMTEMP
Parameter to be passed to ISPZTSO CALL *(ISPZCNT)
'+ISPF ID012732 USERNME NONE NONE
NONE NONE NONE NONE NONE USERNME.TEST.ISPPROF ISPF TSO DINFO sys1.linklib'
Entering Spawn Processing: 03:32:12
PID of this process = 999
RESPfile is /var/ispf/WORKAREA/USERNME/ISPFPID.999
Group PID of this process = 50332642
About to issue system command: ps -u USERNME -o pid,pgid,jobname,xasid,comm
>/var/ispf/WORKAREA/USERNME/ISPFINT.pidlog
PID=6 PGID=50332642 JOBNAME=USERNME ASID=7A COMMAND=/bin/ps
PID=50332642 PGID=50332642 JOBNAME=USERNME ASID=90
COMMAND=/usr/lpp/ispf/bin/ISPZXML
PID=67109859 PGID=50332642 JOBNAME=USERNME ASID=8B COMMAND=/bin/sh
PID=16778213 PGID=50332642 JOBNAME=USERNME ASID=8B COMMAND=./ISPZXENV
PID=998 PGID=50332642 JOBNAME=USERNME ASID=80 COMMAND=/bin/sh
PID=999 PGID=50332642 JOBNAME=USERNME ASID=92 COMMAND=./ISPZINT
PID=16778216 PGID=50332642 JOBNAME=USERNME ASID=7A COMMAND=/bin/sh
No active ISPF session found - new TSO/ISPF session started
About to issue system command: rm /var/ispf/WORKAREA/USERNME/ISPFPID*
mkdir /var/ispf/WORKAREA/USERNME rc = -1
New SIGfile = /var/ispf/WORKAREA/USERNME/ISPFPID.signal
New CMDfile = /var/ispf/WORKAREA/USERNME/ISPFPID.cmd
RUN directory = /usr/lpp/ispf/bin/
About to spawn task for ISPZTSO

```

```

Parameters passed to ISPZTSO - CALL *(ISPZCNT) '+ISPF ID012732
USERNME NONE NONE
NONE NONE NONE NONE NONE USERNME.TEST.ISPPROF ISPF TSO DINFO sys1.linklib'
Return code from ISPZTSO is 0
Waiting on SIGNAL return 03:32:12
Read signal file: Output = COMPLETE
*** OUTPUT FROM ISPF SESSION ***
Entering ISPZCNT (ISPF Initialization)
Parameters +ISPF ID012732 USERNME NONE NONE NONE NONE NONE NONE NONE USERNME.
TEST.ISPPROF ISPF TSO DINFO SYS1.L
REC: sysproc=isp.sispplib,bzz.v1r1.sbzzc1ib,SYS2.CLIST.ISD1.USER,SCLMDW.V710.
SFEKPROC,WD4Z.V710.SFEKPROC
Allocation successful for SYSPROC
REC: sysexec=sclmdw.v3.test.exec,USERNME.sclmdw.exec,USERNME.exec
Allocation successful for SYSEXEC
REC: ispmlib=isp.sispmenu
Allocation successful for ISPMLIB
REC: isptlib=isp.sisptenu
Allocation successful for ISPTLIB
REC: isplib=isp.sisppenu
Allocation successful for ISPLIB
REC: ispslib=sclmdw.v3.dev.skels,bzz.v1r1.sbzzsenu,isp.sispsenu,isp.sispslib
Allocation successful for ISPSLIB
REC: isptrace=nullfile
Allocation successful for ISPTRACE
REC: ISPF_timeout = 300
IEBCOPY of ISPF profile RC = 00
NOTE: Data set allocations took 0.49 elapsed seconds
*** XML-NOTE *** Reference tagged SERVICE-RESPONSE
Current Process List:
      PID      PGID JOBNAME  ASID COMMAND
50331654  50332642 USERNME   7a /bin/sh
      10      50332642 USERNME   7a /bin/ps
50332642  50332642 USERNME   90 /usr/lpp/ispf/bin/ISPZXML
67109859  50332642 USERNME   8b /bin/sh
16778213  50332642 USERNME   8b ./ISPZXENV
      998      50332642 USERNME   80 /bin/sh
      999      50332642 USERNME   92 ./ISPZINT
50332648  50332642 USERNME4  89 ./ISPZTSO
]]>
</OPERATIONS-LOG>
<ISPF-OUTPUT>

```

## Return and reason codes from the gateway

The RETURN-CODE tag provides the value of the return code from the TSO or ISPF service. If there is an error in the gateway, a value of 8 or 12 is returned with the RETURN-CODE tag and REASON-CODE tags of this format are also provided to describe the error:

```

<REASON-CODE ID="ISPZ007">ISPF OR SCLM SERVICE HAS ENDED
ABNORMALLY</REASON-CODE>

```

Table 7 lists the possible reason codes when there is a return code of 8 from the gateway:

Table 7. Possible reason codes for a return code of 8 from the gateway

Reason Code	Description
ISPZ0001	Error in ALLOCATION/WRITE to <i>dsname</i> .
ISPZ0002	Processing terminates.
ISPZ0003	Following function has failed: MSG: <i>function</i>
ISPZ0009	Error reading the ISPF.conf configuration file.

Table 7. Possible reason codes for a return code of 8 from the gateway (continued)

Reason Code	Description
ISPZ0010	Ensure file exists in directory specified by the environment variable CGI_ISPCONF in the server configuration.
ISPZ0012	Error in ISPF data set allocation: See error message below
ISPZ0013	Error in allocating the following DD and data set names: MSG: DD= <i>ddname</i> MSG: <i>dsname1 dsname2... dsnamen</i>
ISPZ0014	Verify that the ISPF configuration file ISPF.conf on the host is correct.
ISPZ0224	*** Operation Cancelled ***
ISPZ0225	Possible previous request still active. Either cancel host session or retry once processing terminates.

Table 8 lists the possible reason codes when there is a return code of 12 from the gateway:

Table 8. Possible reason codes for a return code of 12 from the gateway

Reason Code	Description
ISPZ0007	ISPF or SCLM service has ended abnormally.
ISPZ0008	Review log for error details.

## Using the native API

This section documents the native API used by some IBM products to invoke the gateway. It is recommended, however, that applications use the XML structured API to invoke the gateway.

**Service request format:** Service call requests consist of a parameter string passed by means of STDIN to the interface module ISPZINT. The first word of the service request string identifies if the request is for a TSO or ISPF service:

TSO service-request

or

ISPF service-request

A TSO service request results in the TSO service running in a TSO address space without the overhead of establishing an ISPF environment. Shown here are some examples of TSO service requests:

TSO time

TSO alloc fi(temp1) da('hlq.filetemp') shr

An ISPF service request can be for a TSO or ISPF service. The service runs in an ISPF environment established in a TSO address space. The caller can request the ISPF environment be "reusable" (that is, remain active for running subsequent service requests) by including &SESSION=SPAWN in the parameter string. Also the caller can request an existing ISPF profile be dynamically allocated by including &ISPPROF= my.profile.dataset in the parameter string. Shown here are some examples of ISPF service requests:

ISPF tso profile

ISPF select cmd(%myexec)

ISPF tso listalc&SESSION=SPAWN&ISPPROF=HLQ.ISPPROF

ISPF tso ex 'my.dataset(exec1) '&SESSION=SPAWN

To terminate a "reusable" ISPF session, pass the parameter string shown here to ISPZINT:

```
ISPFUNC=SHUTDOWN
```

or

```
ISPFUNC=CANCEL
```

**Service response format:** Service response data is returned in a tagged format as shown in this section. However this is not valid XML format. Tagged data may be incorporated within system log messages and be stripped out at the client end by means of the encapsulating tags <ISPINFO> </ISPINFO>.

```
System logging data
:
:
<ISPINFO>
ISPF COMMAND : ISPF command issued
RC=X
MSG: xxxxxx
<Specific function request tags>
xxxxxx
xxxxxx
</End of specific function request tags>
</ISPINFO>
System logging data
:
:
```

**Service request and response examples:** Shown here is the request parameter string and response data returned for an ISPF service request:

#### Request

```
ISPF TSO PROFILE&SESSION=SPAWN&ISPPROF=&SYSUID..TEST.ISPPROF
```

#### Response

```
START Transfer DATE/TIME is :Tue Mar 18 12:25:59 WST 2008
At url openConnection
SCLMFunc returned from post is Connect
About to accept response from Server
Response from Server received
Entering ISPZINT (Service initialization)
About to read from fileno(stdin) = 0
Data read from STDIN is ISPF TSO PROFILE&SESSION=SPAWN&ISPPROF=
&SYSUID..TEST.ISPPROF
EPOCH secs = 1205814289
Local Date & time: Tue Mar 18 00:24:49 2008
Hour: 0 Min: 24 Sec 49
Function ID timestamp = ID001489
Environment variables:
0 _CEE_ENVFILE=/DD:ENV
1 # PATH=/bin:./usr/sbin:/usr/lpp/internet/bin:/usr/lpp/internet/sbin:
  /usr/lpp/products/java531-UK14829/J5.0/bin
2 PATH=/bin:./usr/sbin:/usr/lpp/internet/bin:/usr/lpp/internet/sbin:
  /usr/lpp/products/java142/J1.4/bin/
3 SHELL=/bin/sh
4 TZ=EST5EDT
5 LANG=C
6 LC_ALL=en_US.IBM-1047
7 NLSPATH=/usr/lib/nls/msg/%L/%N:/usr/lpp/internet/%L/%N:
  /usr/lib/nls/msg/En_US.
  IBM-1047/%N
8 LIBPATH=/bin:/usr/lpp/internet/bin:/usr/lpp/internet/sbin
9 # JAVA_HOME=/usr/lpp/products/java531-UK14829/J5.0/bin
10 JAVA_HOME=/usr/lpp/products/java142/J1.4/
11 CLASSPATH=./usr/lpp/internet/server_root/CAServlet
12 STEPLIB=CURRENT
13 _BPX_BATCH_SPAWN=SPAWN
```

```

14 _BPX_SHAREAS=YES
15 # _BPX_SHAREAS=YES
16 # _BPX_SPAWN_SCRIPT=NO
17 CGI_DTCONF=/etc/SCLMDTIS
18 CGI_DTWORK=/var/SCLMDTIS
19 CGI_ISPCONF=/etc/SCLMDTIS
20 CGI_ISPWORK=/var/SCLMDTIS
21 # CGI_TRANTABLE=DOHERTL.LSTRANS.FILE
22 COUNTERDIR=NULL
23 REPORTBITS=77
24 SERVER_SOFTWARE=IBM HTTP Server/V5R3M0
25 SERVER_NAME=PTHISD1.AU.IBM.COM
26 SERVER_PORT=1726
27 _BPX_SPAWN_SCRIPT=YES
28 _BPX_USERID=USERNME
29 RULE_FILE=//DD:CONF
30 SERVER_PROTOCOL=HTTP/1.1
31 REQUEST_METHOD=POST
32 GATEWAY_INTERFACE=CGI/1.1
33 PATH_INFO=
34 PATH_TRANSLATED=
35 QUERY_STRING=
36 SERVER_ADDR=192.168.123.11
37 SERVER_TOKEN=1
38 SCRIPT_NAME=/ISPZINT
39 REMOTE_ADDR=192.168.128.253
40 AUTH_TYPE=Basic
41 REMOTE_USER=USERNME
42 CONTENT_TYPE=application/x-www-form-urlencoded
43 CONTENT_LENGTH=61
44 REFERER_URL=
45 DOCUMENT_ROOT=usr/lpp/ispf/bin
46 DOCUMENT_URI=/ISPZINT
47 DOCUMENT_NAME=usr/lpp/ispf/bin/ISPZINT
48 FSCP=IBM-1047
49 NETCP=ISO8859-1
50 HTTPS_KEYSIZE=
51 HTTPS=OFF
52 HTTP_CONNECTION=keep-alive
53 HTTP_ACCEPT=text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2
54 HTTP_HOST=PTHISD1.AU.IBM.COM
55 HTTP_USER_AGENT=Java/1.5.0
56 HTTP_PRAGMA=no-cache
57 HTTP_CACHE_CONTROL=no-cache
58 _CEE_RUNOPTS=ENVAR("_CEE_ENVFILE=//DD:ENV")
Number of environment variables is 59
Connection Protocol = HTTP
Server Name = PTHISD1.AU.IBM.COM
Server Port = 1726
FSCP = IBM-1047
NETCP = ISO8859-1
CGI_ISPCONF = /etc/SCLMDTIS
CGI_ISPWORK = /var/SCLMDTIS
Server PATH = /bin:./usr/sbin:/usr/lpp/internet/bin:/usr/lpp/internet/sbin:
/usr/lpp/products/java142/J1.4/bin/
ISPF standalone function invoked
&ISPPROF value = &SYSUID..TEST.ISPPROF
ISPF COMMAND = ISPF TSO PROFILE
ISPF PROFILE = USERNME.TEST.ISPPROF
Re-usable ISPF session = SPAWN
About to spawn task for ISPZTSO
Parameters passed to ISPZTSO - PROFILE
Return code from ISPZTSO is 0
About to process PROFILE data in /tmp/USERNME.ID001489.ISPF.SYSTSPRT
About to malloc() 252 bytes for profdat
Temporary data set prefix set to : USERNME
About to call bpxwdyn to allocate VCMPTEMP

```

```

Allocating data set USERNME.ISPF.VCMISPF.ID001489 to the VCMTEMP DD
1024 bytes of ISPF TSO PROFILE written to VCMTEMP
1024 bytes of /etc/SCLMDTIS;/var/SCLMDTIS written to VCMTEMP
1024 bytes of /bin:./usr/sbin:/usr/lpp/internet/bin:/usr/lpp/internet/sbin:
/usr/lpp/products/java142/J1.4/bin/~/~ written to VCMTEMP
Parameter to be passed to ISPZTSO CALL *(ISPZCNT) '+ISPF ID001489 USERNME
USERNME.TEST.ISPPROF ISPF TSO PROFILE'
Entering Spawn Processing: 00:24:49
PID of this process = 67109485
RESPfile is /var/SCLMDTIS/WORKAREA/USERNME/ISPFPID.67109485
Group PID of this process = 67109485
No ISPZINT.pidlog detected so no existing ISPF session running
No active ISPF session found - new TSO/ISPF session started
About to issue system command: rm /var/SCLMDTIS/WORKAREA/USERNME/ISPFPID*
mkdir /var/SCLMDTIS/WORKAREA/USERNME rc = -1
New SIGfile = /var/SCLMDTIS/WORKAREA/USERNME/ISPFPID.signal
New CMDfile = /var/SCLMDTIS/WORKAREA/USERNME/ISPFPID.cmd
RUN directory = /usr/lpp/ispf/bin/
About to spawn task for ISPZTSO
Parameters passed to ISPZTSO - CALL *(ISPZCNT) '+ISPF ID001489 USERNME
USERNME.TEST.ISPPROF ISPF TSO PROFILE'
Return code from ISPZTSO is 0
Waiting on SIGNAL return 00:24:49
Read signal file: Output = COMPLETE
*** OUTPUT FROM ISPF SESSION ***
<ISPFINFO>
ISPF COMMAND : SELECT CMD(PROFILE) NEST
RC=0
<ISPF>
IKJ56688I CHAR(0) LINE(0) PROMPT INTERCOM NOPAUSE MSGID
MODE WTPMSG NORECOVER PREFIX(USERNME) PLANGUAGE(ENU)
SLANGUAGE(ENU) VARSTORAGE(LOW)
IKJ56689I DEFAULT LINE/CHARACTER DELETE CHARACTERS IN EFFECT FOR
THIS TERMINAL
</ISPF>
</ISPFINFO>
Current Process List:
      PID      PGID JOBNAME  ASID COMMAND
      94      67109485 USERNME   8a /bin/ps
50332025  67109485 USERNME3  98 ./ISPZTSO
16777835  16777835 USERNME   9b ISRUUDL
67109485  67109485 USERNME   95 /usr/lpp/ispf/bin/ISPZINT
50332289  67109485 USERNME   8a /bin/sh
RC from doGet then writefile :ok
TOTAL Completion DATE/TIME is :Tue Mar 18 12:26:02 WST 2008

```

Shown here is the request parameter string and response data returned for a TSO service request.

### Request

TSO LISTALC

### Response

```

START Transfer DATE/TIME is :Tue Mar 18 12:33:55 WST 2008
At url openConnection
SCLMFunc returned from post is Connect
About to accept response from Server
Response from Server received
Entering ISPZINT (Service initialization)
About to read from fileno(stdin) = 0
Data read from STDIN is TSO LISTALC
EPOCH secs = 1205814765
Local Date & time: Tue Mar 18 00:32:45 2008
Hour: 0 Min: 32 Sec 45
Function ID timestamp = ID001965
Environment variables:
0 _CEE_ENVFILE=//DD:ENV

```

```

1 # PATH=/bin:./usr/sbin:/usr/lpp/internet/bin:/usr/lpp/internet/sbin:
  /usr/lpp/products/java531-UK14829/J5.0/bin
2 PATH=/bin:./usr/sbin:/usr/lpp/internet/bin:/usr/lpp/internet/sbin:
  /usr/lpp/products/java142/J1.4/bin/
3 SHELL=/bin/sh
4 TZ=EST5EDT
5 LANG=C
6 LC_ALL=en_US.IBM-1047
7 NLSPATH=/usr/lib/nls/msg/%L/%N:/usr/lpp/internet/%L/%N:
  /usr/lib/nls/msg/En_US.
  IBM-1047/%N
8 LIBPATH=/bin:/usr/lpp/internet/bin:/usr/lpp/internet/sbin
9 # JAVA_HOME=/usr/lpp/products/java531-UK14829/J5.0/bin
10 JAVA_HOME=/usr/lpp/products/java142/J1.4/
11 CLASSPATH=./usr/lpp/internet/server_root/CAServlet
12 STEPLIB=CURRENT
13 _BPX_BATCH_SPAWN=SPAWN
14 _BPX_SHAREAS=YES
15 # _BPX_SHAREAS=YES
16 # _BPX_SPAWN_SCRIPT=NO
17 CGI_DTCONF=/etc/SCLMDTIS
18 CGI_DWORK=/var/SCLMDTIS
19 CGI_ISPCONF=/etc/SCLMDTIS
20 CGI_ISPWORK=/var/SCLMDTIS
21 # CGI_TRANTABLE=DOHERTL.LSTRANS.FILE
22 COUNTERDIR=NULL
23 REPORTBITS=77
24 SERVER_SOFTWARE=IBM HTTP Server/V5R3M0
25 SERVER_NAME=PTHISD1.AU.IBM.COM
26 SERVER_PORT=1726
27 _BPX_SPAWN_SCRIPT=YES
28 _BPX_USERID=USERNME
29 RULE_FILE=//DD:CONF
30 SERVER_PROTOCOL=HTTP/1.1
31 REQUEST_METHOD=POST
32 GATEWAY_INTERFACE=CGI/1.1
33 PATH_INFO=
34 PATH_TRANSLATED=
35 QUERY_STRING=
36 SERVER_ADDR=192.168.123.11
37 SERVER_TOKEN=1
38 SCRIPT_NAME=/ISPZINT
39 REMOTE_ADDR=192.168.128.253
40 AUTH_TYPE=Basic
41 REMOTE_USER=USERNME
42 CONTENT_TYPE=application/x-www-form-urlencoded
43 CONTENT_LENGTH=12
44 REFERER_URL=
45 DOCUMENT_ROOT=usr/lpp/ispf/bin
46 DOCUMENT_URI=/ISPZINT
47 DOCUMENT_NAME=usr/lpp/ispf/bin/ISPZINT
48 FSCP=IBM-1047
49 NETCP=ISO8859-1
50 HTTPS_KEYSIZE=
51 HTTPS_OFF=
52 HTTP_CONNECTION=keep-alive
53 HTTP_ACCEPT=text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2
54 HTTP_HOST=PTHISD1.AU.IBM.COM
55 HTTP_USER_AGENT=Java/1.5.0
56 HTTP_PRAGMA=no-cache
57 HTTP_CACHE_CONTROL=no-cache
58 _CEE_RUNOPTS=ENVAR("_CEE_ENVFILE=//DD:ENV")
Number of environment variables is 59
Connection Protocol = HTTP
Server Name = PTHISD1.AU.IBM.COM
Server Port = 1726
FSCP = IBM-1047

```

```
NETCP = ISO8859-1
CGI_ISPCONF = /etc/SCLMDTIS
CGI_ISPWORK = /var/SCLMDTIS
Server PATH = /bin:./usr/sbin:/usr/lpp/internet/bin:/usr/lpp/internet/sbin:
/usr/lpp/products/java142/J1.4/bin/
About to spawn task for ISPZTS0
Parameters passed to ISPZTS0 - LISTALC
Return code from ISPZTS0 is 0
About to open /tmp/USERNME.ID001965.ISPF.SYSTSPRT
<ISPINFO>
<TSO>
PDFTDEV.USERNME.LOAD
PDFTDEV.STG.LOAD
PDFTDEV.INT.LOAD
PDFTDEV.SVT.LOAD
USERNME.SCLMDW.LOAD
SCLMDW.V3TEST.LOAD
SCLMDW.V3BASE.LOAD
/tmp/USERNME.ID001965.ISPF.SYSTSPRT
</TSO>
</ISPINFO>
RC from doGet then writefile :ok
TOTAL Completion DATE/TIME is :Tue Mar 18 12:33:56 WST 2008
```



---

## Chapter 4. Improving ISPF performance

To improve ISPF performance consider using these methods, which are described in detail here:

- Use virtual I/O (VIO). See page 67.
- Remove or tailor edit functions. See page 67.
- Remove action bars from ISPF panels. See page 68.
- Remove scrollable areas from ISPF panels. See page 69.
- Preprocess ISPF panels. See page 69.
- Add load modules to LPALST. See page 71.
- Allocate execution data sets. See page 73.
- Disable generic searches. See page 74.
- Preallocate ISPF temporary data sets to VIO. See page 76.

Here are some other suggestions, which require no further description:

- Use a workstation editor to make many changes to a file
- Use Browse instead of View for large files
- Use a workstation compiler (whether in SCLM or not) for unit test compiles (such as C++SET/2 or COBOL Productivity Suite)
- Use an action bar to call other functions from Edit or Browse if you intend to return to the Edit or Browse session
- Use the IBM REXX compiler to compile ISPF dialogs written in REXX.

---

### Use virtual I/O

Unit name SYSALLDA is the default specified in the ISPF Configuration table (keyword name PDF\_DEFAULT\_UNIT). When defining SYSALLDA, or another unit if you modify the configuration table, use the VIO=YES option on the UNITNAME macro. PDF allocates its temporary data sets using VIO if it is available. See Chapter 2, “The ISPF Configuration Table,” on page 9 for information about modifying the configuration table.

---

### Remove or tailor Edit functions

Using the ISPF configuration table, you can tailor or remove functions that are likely to increase resource use. These include:

- Edit recovery data set attributes (RECOVERY ON)
- Edit in-storage change tracking (SETUNDO STORAGE)
- Edit enhanced highlighting (HILITE)

### Change the block size of the edit recovery data set

The performance of the edit recovery function can be enhanced by changing the block size of the edit recovery data set. This change is to the keyword EDIT\_RECOVERY\_BLOCK\_SIZE in the ISPF configuration table. Similar fields exist in the ISPF configuration table for other less frequently used data sets. See Chapter 2, “The ISPF Configuration Table,” on page 9 for more information about the ISPF configuration table.

## Disable SETUNDO STORAGE

The SETUNDO STORAGE function of the editor lets users undo changes made within edit sessions by keeping a record of those changes in storage. This provides better performance for individual users but can have a negative impact on overall system performance on systems where the editor is heavily used.

The function of saving the record of changes in storage can be completely disabled by setting the keyword UNDO\_STORAGE\_SIZE to zero in the ISPF configuration table. The UNDO function will still be available to users who are running with RECOVERY ON.

By default, existing user edit profiles that have never had SETUNDO explicitly changed, will have SETUNDO STORAGE enabled. Newly created profiles will have SETUNDO STORAGE turned off. If you want to force most of the existing, heavily used profiles to specify SETUNDO OFF, disable the function by changing the configuration table keyword UNDO\_STORAGE\_SIZE to zero and run the system like that for several days or weeks. When the function is disabled, edit profiles are changed to reflect SETUNDO OFF when they are used.

After several days or weeks, you might want to enable the function by again changing the configuration table. See Chapter 2, “The ISPF Configuration Table,” on page 9 for more information on the ISPF configuration table.

**Note:** If you normally place a copy of the edit profile table (normally ISREDIT) containing a ZDEFAULT profile in your ISPTLIB concatenation, you should ensure that it reflects the SETUNDO setting of your choice, because this will override the programmed defaults for newly created edit profiles.

## Disable Edit extended highlighting

Although edit extended highlighting provides powerful features (such as language sensitive color), it might require an unacceptable amount of CPU time system wide. To disable the feature, change the value of the keyword DEFAULT\_EDIT\_DISPLAY in the ISPF configuration table to a value of 0 or 1.

To completely disable extended highlighting, even for applications that use their own edit panels, set the value of keyword ALLOW\_EDIT\_HIGHLIGHTING in the ISPF configuration table to N0. See Chapter 2, “The ISPF Configuration Table,” on page 9 for more information about the ISPF configuration table.

---

## Remove action bars from ISPF panels

Some of the panels that are provided with the Dialog Tag Language (DTL) source can be recompiled to remove the action bars. This can result in some performance benefits. To recompile the DTL source:

1. Invoke ISPD TLC with the NOACTBAR option by specifying:

```
ISPD TLC (NOACTBAR
```

or by de-selecting the option “Create panels with Action bars” on the invocation panel.

2. Modify the convert EXEC to include the NOACTBAR parameter:

```
'ISPD TLC ISPFAB (DISK KEYLAPPL=ISR MSGSUPP CUAATTR CUASUPP PLEB MCOMMENT  
NOSTATS NOACTBAR PROFILE=your.gml.dataset(profile)';
```

3. If you are recompiling DBCS versions of the panels, you must add the DBCS option to the command syntax. Japanese panels require the KANA option as well.

```
'ISPDTLC ISPFALL (DISK KEYLAPPL=ISR MSGSUPP CUAATTR CUASUPP PLEB MCOMMENT  
JAPANESE DBCS KANA NOSTATS NOACTBAR PROFILE=your.gml.dataset(profile)' ;
```

For more information about invoking ISPDTLC, see “Invoking the ISPDTLC conversion utility” on page 40.

---

## Remove scrollable areas from ISPF panels

Some of the panels that are provided with the Dialog Tag Language (DTL) source can be recompiled to remove scrollable areas (where the resulting panel fits within a standard 24 line screen). This can result in some performance benefits. To recompile the DTL source:

1. Invoke ISPDTLC with the MERGESAREA option by specifying:

```
ISPDTLC (MERGESAREA
```

or by selecting the option “Combine scrollable areas into panel body” on the invocation panel.

2. Modify the convert EXEC to include the MERGESAREA parameter:

```
'ISPDTLC ISPFALL (DISK KEYLAPPL=ISR MSGSUPP CUAATTR CUASUPP PLEB MCOMMENT  
NOSTATS MERGESAREA PROFILE=your.gml.dataset(profile)' ;
```

3. If you are recompiling DBCS versions of the panels, you must add the DBCS option to the command syntax. Japanese panels require the KANA option as well.

```
'ISPDTLC ISPFALL (DISK KEYLAPPL=ISR MSGSUPP CUAATTR CUASUPP PLEB MCOMMENT  
JAPANESE DBCS KANA NOSTATS MERGESAREA PROFILE=your.gml.dataset(profile)' ;
```

**Note:** The NOACTBAR AND MERGESAREA options can be combined to provide both results.

For more information about invoking ISPDTLC, see “Invoking the ISPDTLC conversion utility” on page 40.

---

## Preprocess all ISPF panels

Preprocessed panels are displayed much faster than if they are not preprocessed. However, panels that must be sized at run time (dynamic panels) cannot be preprocessed. After preprocessing the ISPF panels, copy all panels into your execution data set, specifying NOREPLACE. This will copy only those members that could not be preprocessed.

**Note:** The preprocessed versions of the ISPF panels are not updated by PTFs from IBM. If a panel is updated by a PTF, you must preprocess it into your execution data set again.

## Preprocessed panel utility

The preprocessed panel utility is an ISPF dialog called ISPPREP. This utility converts panel definitions to a form ISPF can display more quickly. The converted panels can be defined to ISPF in place of normal panel definitions. This improves ISPF’s performance.

When you install ISPF, consider creating a panel library to contain the preprocessed versions of the ISPF panels. Be aware that a preprocessed panel is in

an encoded form and cannot be changed through the normal edit procedure. For a general discussion of ISPPREP and its use, refer to the *z/OS ISPF Dialog Developer's Guide and Reference*.

This example shows how to create preprocessed versions of the ISPF panels contained in the data set ISP.SISPPENU. The example assumes this data set is cataloged. However, this is not necessary and both the panel input and panel output data sets can be uncataloged.

Before continuing, you must allocate the data set that will contain the preprocessed panels. For this example, catalog the data set and name it ISP.PREPPLIB. Allocate the data set with the same record format, logical record length, and block size of the ISPF panel data set, ISP.SISPPENU. However, the entire ISPF panel data set in the preprocessed format will take up about 25% less space than the original data set, so allocate the new data set accordingly.

To convert the ISPF panel library, issue the ISPSTART command as follows:

```
ISPSTART PGM(ISPPREP) PARM(INPAN('ISP.SISPPENU'),  
                           OUTPAN('ISP.PREPPLIB'))
```

**Note:** This command creates all members in the output library with ISPF statistics included. Be sure that you have enough directory blocks to contain the members and their statistics, or use the NOSTAT parameter for ISPPREP. See the *z/OS ISPF Dialog Developer's Guide and Reference* for more information about ISPPREP.

The panel output data set name in the previous example is a suggested naming convention. Regardless of how you name the data set, be sure you modify your TSO LOGON procedure or the CLIST that allocates the ISPF data sets to allocate the new panel library. See the *z/OS ISPF User's Guide Vol I* for more information about allocating ISPF libraries.

The previous example shows the use of ISPPREP in foreground batch mode. You can also run ISPPREP as a background job, as an interactive dialog by selecting option 2 on the ISPF Primary Option Menu, or by entering the ISPPREP command on any Command line.

These restrictions apply to the panels you can convert to preprocessed panels. You cannot convert any panel that contains these items in the panel definition:

- A dialog variable specified with the WIDTH keyword in the )BODY header statement of a panel.
- A dialog variable that defines a model line in a table display panel definition.
- A dynamic or graphic area that has EXTEND(ON) specified for the attribute character.

If ISPPREP is passed a panel definition that does not meet these restrictions, a message is issued to the ISPF log data set that specifies the name of the panel that violated the restrictions. In this case, the panel definition is not converted to the preprocessed format. However, you can copy the original panel definitions into the new panel library to keep your panels grouped accordingly. In any case, be sure to check the ISPF log data set after invoking ISPPREP. The messages in the log data set help you identify any problems ISPPREP encounters during the conversion. For more information, refer to the *z/OS ISPF Dialog Developer's Guide and Reference*.

---

## Add load modules to the pageable link-pack area (LPA)

Once you have installed and verified ISPF, you can enhance its performance by adding the LPA-eligible load modules (in the SISPLPA library) to the LPA list in an LPALSTxx member of PARMLIB. Add those load modules not eligible for LPA (that is, those in the SISPLOAD library that are not re-entrant) to the link list in an LNKLSTxx member of PARMLIB. For information about adding data sets to the Link and LPA lists, see *z/OS MVS Initialization and Tuning Reference*. You can then remove these data sets from the STEPLIB in your TSO LOGON procedure. After adding SISPLPA to LPALST and SISPLOAD to LNKLST, specify CLPA as an initial program load (IPL) parameter to force the SISPLPA modules into the link pack area and to have SISPLOAD added as a system link library.

ISPF performance directly relates to the number of load modules that reside in the pageable LPA because these modules are not loaded into the user's private storage when they are called. For optimum performance, all of the eligible ISPF licensed program load modules should be in the LPA. However, to load all eligible modules to the LPA, your system would need a very large LPA.

Any module that is re-entrant (that is, has the RN attribute) is eligible for the LPA.

No modules are required in the LPA. Table 9 on page 72 lists the load modules recommended for inclusion in the LPA.

To conserve LPA space:

- Move some of the infrequently used modules to a system-link library. Keep the modules you most frequently use in the LPA.
- If you are not using SCLM, move all load modules that begin with FLM from the SISPLPA library to the SISPLOAD library so that they are in a system link library instead of the Link-Pack area.

## Moving load modules

To move load modules from one library to another, use SMP/E ++MOVE statements. SMP/E moves the load modules and all associated aliases and updates the SMP CSI. This ensures that the load modules are available to SMP/E for subsequent maintenance.

Sample usermods that contain SMP/E ++MOVE statements are included with ISPF to help you customize your system if you decide to put only part of the SISPLPA load modules into LPA.

Usermod ISPMOVE1 contains a **sample** ++MOVE statement to move a load module from SISPLPA to SISPLOAD. Usermod ISPMOVE2 contains a **sample** ++MOVE statement to move a load module from SISPLOAD to SISPLPA.

### Notes:

1. These sample usermods are not complete. They do not contain ++MOVE statements for the possible load modules that may be moved. They contain a sample ++MOVE statement to show you the format required. Before using them, you must edit them for applicability to your environment. Provide statements for those load modules that you want to move from one library to another. See the instructions in the samples for more information about changes you need to make.
2. Be aware that some PTFs may change the JCLIN for ISPF load modules. If the JCLIN is changed for a load module that you have moved, after application of

the PTF, the load module may reside back in the library you moved it from. You should carefully examine the output from the application of any PTF, and also the information in your SMP/E zone, and if necessary re-apply the usermod.

If you move the modules without using SMP/E ++MOVE statements, consider:

- You can use the ISPF Move and Copy utility (option 3.3) to move load modules between the SISpload and SISPLPA data sets. If you do this, you must also update the load module SYSLIB subentries in the SMP CSI to reflect the move, otherwise the modules will not be available to SMP/E for maintenance.
- Many of the load modules have aliases. You can determine the alias of any load module by viewing the data set list (from ISPF option 3.4) of SISPLPA and SISpload. When you use ISPF option 3.3 to move modules that have aliases, make sure that 'Process member aliases' is selected in the panel where you specify the "To" data set.

Five SCLM load modules are shipped as ++PROGRAMs. They are: FLMCQBLD, FLMCQCNT, FLMCQINT, FLMCQPRM, and FLMCQSTQ. You cannot use the usermods to move these load modules as SMP/E does not allow for moving a ++PROGRAM by a ++MOVE statement.

## ISPF load module descriptions

This section contains information describing ISPF load modules.

Table 9 lists the frequently used ISPF load modules that should be in the LPA and a brief description of each.

Any load module distributed in SISpload that does not have the re-entrant (RN) attribute is not eligible for the LPA and should not be moved to SISPLPA.

Load modules shipped with the attribute RMODE(24) reside below the 16M line.

**Note:** Load module ISPLINK is shipped with the attribute RMODE(24) to provide compatibility with dialogs that are AMODE(24) and use a LOAD and CALL interface to ISPLINK. However, programs that will reside above the 16MB line (RMODE(ANY)) and include ISPLINK in their load module can override the RMODE(24) at link-edit time. ISPLINK code can reside and execute above the 16MB line.

*Table 9. Minimum recommended load modules for the LPA*

Load Module	Description
FLM\$CPI	SCLM
FLMB	SCLM
FLMCPCS	SCLM
FLMDDL	SCLM
FLMIO24	SCLM
FLMP	SCLM
FLMRTLIB	SCLM
FLM\$LNK	SCLM
FLMS7C	SCLM
ISPICP	ISPF driver (alias name: ISPSTART)
ISPISM	Settings processor
ISPKEY	KEYLIST command processor
ISPCLU	KEYLIST Utility

Table 9. Minimum recommended load modules for the LPA (continued)

Load Module	Description
ISPLLP	Log/List processor
ISPMAIN	ISPF controller
ISPNLxxx	where xxx designates the languages you are using
ISPOMI	MSGID processor
ISPOPF	PFSHOW processor
ISPOPI	Panel ID processor
ISPOPT	Options processor
ISPOGUI	ISPF C/S processor
ISPSUBS	Common subroutines (alias name: ISPCIP)
ISPSUBX	Common subroutines
ISPTASK	Dialog driver
ISPTCM	TSO command table
ISPTUTOR	Tutorial processor (option T)
ISPYLxxx	where xxx designates the languages you are using
ISP32mmm	where mmm designates the terminal types in use
ISRBRO	Main Browse module
ISREDIT	Main Edit module
ISRNLxxx	where xxx designates the languages you are using
ISRPLEX	Edit extended highlight module
ISRPX	Edit extended highlight customization module
ISRRCL	Action bar router utility
ISRSEPRM	More SuperC
ISRSUBS	Main PDF subroutine module
ISRSUBX	Main below 16M PDF subroutine module
ISRSUPC	Superc
ISRUDA	PDF Utilities
ISRUMC	PDF Move Copy
ISR32mmm	where mmm designates the terminal types in use

Include FLMS7C to enhance the performance of SCLM batch jobs and tools or dialogs that use the FLMCMD interface to SCLM services. Include FLMS\$LNK for tools or dialogs using the FLMLNK interface to SCLM services. Include FLMLPCBL if you are using COBOL source; FLMLPFRT if you are using FORTRAN source; or FLMLPGEN if you are using Assembler, REXX, PLI, or text source.

## Allocate execution data sets

Consider these points when creating data sets that are used at execution time and when setting up ISPF applications.

- System-Determined Block size (SDB) should be used for the ISPxLIB data sets. To allocate a data set with SDB, code DCB=BLKSIZE=0 in JCL, or specify a block size of zero in ISPF Option 3.2 when allocating the data set. The exception to this is ISPLLIB data sets with a record format of U. These should be allocated using a block size consistent with your system conventions.
- If you do not use a system-determined block size, use a block size for the ISPxLIB data sets that is a half of a 3380/3390 track. Use a block size of 32760 for load module data sets including those allocated to ISPLLIB, if any.
- Use cached controllers for the ISPxLIB data sets.
- Use PDSEs for the ISPxLIB data sets. Pay careful attention to the parameters with which the PDSEs are defined. The direct MSR (millisecond response time)

parameter of the storage class affects PDSE performance. Low values of this parameter will improve PDSE performance at the expense of increased storage use.

- Make sure the ISPxLIB data sets are on lightly used volumes. You can spread them out over multiple volumes.
- Minimize the number of ISPLLIB data sets:
  - If ISPF modules are in the LPA and LNKLIST, do not include them in ISPLLIB. It is not necessary.
  - Put commonly used ISPF applications in the LPA and LNKLIST.
  - Use LIBDEFs where possible for infrequently used applications.

---

## Disable generic high-level qualifiers

ISPF allows a generic high-level qualifier when using option 3.4 to print or display a list of data set names. A generic high-level qualifier is one that contains a wildcard character. You can restrict the search limits (and improve performance) by disabling the use of wildcard characters in the first qualifier. You can do this in two ways:

- Through the `DISALLOW_WILDCARDS_IN_HLQ` setting in the ISPF Configuration table.
- By writing an exit that sets a return code to prevent the list from being generated when it finds a wildcard character. This method allows more flexibility, for example if you wish to allow a wildcard character at the end of a high-level qualifier but not at the start.

For more information about using the data set list filter exit, see “Data set list filter exit” on page 181.

Figure 47 on page 75 is an example of a data set list exit that disables wildcards. A copy of this exit is included in your `SISPSAMP` data set in member `ISRNOGEN`. Note that this example as written is nonreentrant, so it should not be put in the LPA. If nonreentrant exits are put in the LPA, abends can occur.

```

    TITLE ' ISRNOGEN:  PROC (PARM1,PARM2,PARM3,PARM4);'
* This program is an example of a data set list exit.  It checks
* for a generic high-level qualifier and sets the return code
* to prevent the list from being generated when there is a generic
* high-level qualifier.
ISRNOGEN CSECT ,
ISRNOGEN AMODE 31
ISRNOGEN RMODE ANY
@PROLOG STM @14,@12,12(@13)
        BALR @12,0
@PSTART DS 0H
        USING @PSTART,@12
        MVC  PARMADDR(16),0(@01)
* If the first parameter is one, check the data set name for
* a generic character (% or *)
CHECK1  L @04,PARMADDR
        CLC  PARM1(4,@04),ONE
        BNE  EXITRTN
* Check current Dsname
        L @05,PARMADDR+4
        LR @01,@05
        XR @02,@02
* Translate and test for *, %, blank, or period
        TRT  DSNAME(44,@05),TABLE
        SR @01,@05
        AR @05,@01
* Check the character that stopped the translate and test for an
* asterisk or a percent sign.  If either of these is the first
* character found, the high-level qualifier is generic.
        CLC  DSNAME(1,@05),ASTERISK
        BE  ERROR
        CLC  DSNAME(1,@05),PERCENT
        BE  ERROR
* If the high-level qualifier is not an asterisk or percent sign,
* set the return code to allow the list to be displayed.
NAMEOK  SLR @06,@06
        ST @06,EXITRC
EXITRTN DS 0H
        L @15,EXITRC
        L @14,12(,@13)
        LM @00,@12,20(@13)
        BR @14
* If the high-level qualifier is generic, set the return code to 8
* to prevent searching all catalogs.
ERROR   MVC  EXITRC(4),EIGHT
        B  EXITRTN
* Data for checkname /* */
@DATA   DS 0H
        DS 0F
PARMADDR DS 4F
        DS 0F
ONE     DC F'1'
TWO    DC F'2'
FOUR   DC F'4'
EIGHT  DC F'8'
        DS 0D
EXITRC  DS F
ASTERISK DC C'*'
PERCENT DC C'%'

```

Figure 47. Sample exit to disable generic high-level qualifiers (Part 1 of 2)

```

* Translate and test table with blank, period, asterisk, and
* percent signs
TABLE    DC    64X'00'
         DC    X'40'
         DC   10X'00'
         DC   X'4B'
         DC   16X'00'
         DC   X'5C'
         DC   15X'00'
         DC   X'6C'
         DC  147X'00'
@00      EQU   00          Equates for registers 0-15
@01      EQU   01
@02      EQU   02
@03      EQU   03
@04      EQU   04
@05      EQU   05
@06      EQU   06
@07      EQU   07
@08      EQU   08
@09      EQU   09
@10      EQU   10
@11      EQU   11
@12      EQU   12
@13      EQU   13
@14      EQU   14
@15      EQU   15
VOLUME   EQU   0
DSNAME   EQU   0
EXIT1    EQU   0
LEVEL    EQU   0
PARM1    EQU   0
PARM2    EQU   0
PARM3    EQU   0
PARM4    EQU   0
         DS    0D
@ENDDATA EQU   *
         END   ISRNOGEN

```

Figure 47. Sample exit to disable generic high-level qualifiers (Part 2 of 2)

---

## Preallocate ISPF temporary data sets to VIO

ISPF uses temporary data sets to generate JCL or utility control statements or to generate listings. To preallocate these data sets to VIO, include the DD statements in Figure 48 on page 77 in the TSO LOGON procedure.

Preallocation of these data sets to VIO is not mandatory; ISPF automatically allocates them to real data sets if required. However, preallocation is recommended, because it reduces overhead and eliminates potential problems from insufficient space.

```

//ISPCTL0 DD DISP=NEW,UNIT=VIO,SPACE=(CYL,(1,1)),
//          DCB=(LRECL=80,BLKSIZE=800,RECFM=FB)

//ISPCTL1 DD DISP=NEW,UNIT=VIO,SPACE=(CYL,(1,1)),
//          DCB=(LRECL=80,BLKSIZE=800,RECFM=FB)
:
//ISPCTLW DD DISP=NEW,UNIT=VIO,SPACE=(CYL,(1,1)),
//          DCB=(LRECL=80,BLKSIZE=800,RECFM=FB)

/* In this section of JCL, there is one DD for each screen
/* defined, based on the value of keyword MAXIMUM_NUMBER_OF_
/* SPLIT_SCREEN in the configuration table.
/* The DD name is in the form ISPCTLx, where x can be
/* 1-9, A-W. For example, if the keyword value = 8, only
/* ISPCTL1 to ISPCTL8 need to be coded.
/* ISPCTL0 is a special case, used only by Edit for the Submit
/* command.

//ISPWRK1 DD DISP=NEW,UNIT=VIO,SPACE=(CYL,(1,1)),
//          DCB=(LRECL=256,BLKSIZE=2560,RECFM=FB)

//ISPWRK2 DD DISP=NEW,UNIT=VIO,SPACE=(CYL,(1,1)),
//          DCB=(LRECL=256,BLKSIZE=2560,RECFM=FB)
:
//ISPWRKW DD DISP=NEW,UNIT=VIO,SPACE=(CYL,(1,1)),
//          DCB=(LRECL=256,BLKSIZE=2560,RECFM=FB)

/* In this section of JCL, there is one DD for each screen
/* defined, based on the value of keyword MAXIMUM_NUMBER_OF_
/* SPLIT_SCREEN in the configuration table.
/* The DD name is in the form ISPWRKx, where x can be
/* 1-9, A-W. For example, if the value of the keyword = 8,
/* only ISPWRK1 to ISPWRK8 need to be coded.

//ISPLST1 DD DISP=NEW,UNIT=VIO,SPACE=(CYL,(1,1)),
//          DCB=(LRECL=121,BLKSIZE=1210,RECFM=FBA)

//ISPLST2 DD DISP=NEW,UNIT=VIO,SPACE=(CYL,(1,1)),
//          DCB=(LRECL=121,BLKSIZE=1210,RECFM=FBA)
:
//ISPLSTW DD DISP=NEW,UNIT=VIO,SPACE=(CYL,(1,1)),
//          DCB=(LRECL=121,BLKSIZE=1210,RECFM=FBA)

/* In this section of JCL, there is one DD for each screen
/* defined, based on the value of keyword MAXIMUM_NUMBER_OF_
/* SPLIT_SCREEN in the configuration table.
/* The DD name is in the form ISPLSTx, where x can be
/* 1-9, A-W. For example, if the value of the keyword = 8,
/* only ISPLST1 to ISPLST8 need to be coded.

```

Figure 48. DD statements to preallocate data sets

**Notes:**

1. When allocating to VIO, make sure that enough auxiliary storage is dedicated to VIO so that system availability is not affected.
2. Use of the BUFNO parameter on allocation of ISPF libraries is not supported.
3. The ISPF temporary data set default names associated with the ISPCTLx are SPFTEMPx.CNTL, respectively, where x= value 0-9, A-W.
4. The ISPF temporary data set default names associated with the ISPWRKx are SPFTEMPx.WORK, respectively, where x= value 1-9, A-W.

5. The ISPF temporary data set default names associated with the ISPLSTx are SPFTEMPx.LIST, respectively, where x= value 1-9, A-W.
6. The ddnames ISPWRKx are used by ISPF for file tailoring services with ISPFILe allocated to a PDS. The ddnames ISPLSTx are used for generated listings.
7. The ddname ISPCTL0 is used with the edit SUBMIT command. The ddnames ISPCTLx are used with ISPF compress (both interactive and the LMCOMP service) and ISPF background (option 5), and for processing the file tailoring service FTOPEM TEMP.
8. ISPCTL1 is a required ddname when using SCLM. The data set should be allocated as an ISPF temporary data set for SCLM foreground processing and should be added to the FLMLIBS skeleton for batch processing.
9. ISPF does not support multivolume temporary data sets. If DFSMS™ is installed, temporary data sets dynamically allocated by ISPF must be assigned a data class with a volume count of one. This can be controlled in the ACS routines by testing the execution mode (&XMODE) for a value of TSO.
10. If you have dialogs that need to edit or browse temporary data sets, use the LMINIT service to associate a DATAID with ddname ZTEMPN and invoke edit or browse using the DATAID parameter. For more information, refer to the BROWSE and EDIT services in the *z/OS ISPF Services Guide*.

---

## Compress SCLM listings

The ISPF sample library ISP.SISPSAMP contains two sample members that demonstrate how to compress SCLM listings using the PACK option on the LMCOPY service. FLM03ASM is the language definition. FLM03LMC is a sample REXX exec. Comments explaining how to use these members are included in the code.

---

## Chapter 5. Customizing DM

Customizing DM describes procedures you can use to customize the DM component of ISPF to suit the particular needs of your installation:

- Changing the default name for the ISPF Client/Server component download data set
- Set up ISPF GUI for TSO Line Mode support
- SMF command accounting
- Preallocation of List/Log data sets to SYSOUT
- Specifying the maximum number of split screens
- Setting ISPF site-wide defaults
- Customizing command tables
- Creating ISPF terminal translation tables
- Creating ISPF code page translation tables
- Displaying square brackets used in C programs
- ISPEXEC processing
- ISPF-to-APL2 terminal type mappings
- Load APL2 workspace
- Tailoring ISPF defaults
- Customizing the ISPF TSO command table (ISPTCM)
- Alternate option 7.1 panels
- ISPF multicultural support

See also Chapter 6, “ISPF installation-wide exits,” on page 113.

---

### Changing the default name for the ISPF Client/Server component download data set

If the default data set names are not used for the ISPF data sets, use the ISPF Configuration utility to set the default data set name for the Client/Server Component Download. See Chapter 2, “The ISPF Configuration Table,” on page 9 for details.

---

### Set up ISPF GUI for TSO Line Mode support

When running ISPF in GUI mode, users connected by TCP/IP on a z/OS UNIX system, and all users connected by APPC, have the option to display all non-fullscreen TSO data in an ISPF/TSO GUI window. This window is scrollable and it contains an input field for entering required user responses. The data in the window can be selected and copied to a file of your choice. This enables you to minimize the emulator window from which you logged on without missing any messages that might appear in the emulator window.

To use the TSO line mode support for ISPF GUI, execute these steps:

- The Language Environment run-time library data sets, SCEERUN and SCEERUN2, must be in LPA or Linklist and must be APF authorized.
- These ISPF modules and aliases must be in LPA: ISPDTSI, ISPDWIN, ISPDTSK, ISPDINI, IKJDTINI, ISPDTER, IKJDTTER, ISPDTPUT, IKJDTPUT, ISPDGET, IKJDGET, ISPDTPC, and ISPQGUI.
- The ISPF subsystem initialization routine, ISPDTSI, must be invoked at system IPL. To do this, add this to your IEFSSNxx member in SYS1.PARMLIB.

```
SUBSYS SUBNAME(ISPF) INITRTN(ISPDTSSI)
```

One of these messages will appear on your console indicating whether the routine was successfully invoked:

```
ISPDT001 ISPF Subsystem Initialization: Completed successfully
ISPDT002 ISPF Subsystem Initialization: Error creating PC number - is
        ISPDTPC in LPA
ISPDT003 Subsystem name in IEFSSNxx is not ISPF
```

- These are required for TCP/IP communications:
  - z/OS Communications Server: IP
  - user ID with an OMVS segment defined
  - default group with an OMVS segment defined
  - z/OS UNIX home directory

See “Tailoring ISPF defaults” on page 106 for more information.

- Your logon procedure must invoke the alternate entry point of IKJEFT1I. The EXEC card must contain 'PGM=IKJEFT1I'.

---

## SMF command accounting

The MVS System Management Facility (SMF) collects and records a variety of system and job-related information. ISPF uses SMF to format information for a type 32 record. This type 32 record contains the names of the program functions and TSO commands being executed and the number of times each is used during the session. SMF also allows the installation to specify that the record is to include resources such as the total processor time under TCBs and SRBs and the total number of TGETs, TPUTs, and transactions associated with each name. The record is written when a TSO user logs off or when an SMF recording interval expires.

ISPF issues an SVC 109 (the extended service router, code X'19') to start and stop this functional accounting before and after the link to a program function, and before and after attaching a command. When processing command (CLIST) functions, the SVC 109 is issued before and after attaching the EXEC command processor. Therefore, the command name EXEC will be recorded, rather than the actual CLIST name.

The calls to SVC 109 can be nested depending on the nature of the program function or CLIST. The commands (and service units attributed to each command, if recorded) are accounted to the appropriate logical screen and are recorded as such. The command name ISPF SWAP is passed to SMF to indicate the user has swapped screens.

You must specify in module IEEMB846 the name of each module that is invoked for ISPF subfunctions or subcommands. You should also specify the command used to invoke ISPF. *z/OS MVS System Management Facilities (SMF)* provides details on including additional names. Without these additions to IEEMB846, ISPF will cause extensive counts of \*\*\*OTHER to be recorded in the SMF type 32 records.

Table 10 lists the ISPF options and their related module names plus any additional commands that can be implicitly or explicitly invoked by a particular option.

*Table 10. ISPF Options and related module names*

Option	Module Names	TSO Commands
0	ISPISM	

Table 10. ISPF Options and related module names (continued)

Option	Module Names	TSO Commands
1	ISRBRO	
2	ISREDIT	SUBMIT
3.1 and 3.2	ISRUDA	
3.3	ISRUMC	
3.4	ISRUDL	
3.5	ISRURS	
3.6	ISRUHC	SUBMIT, ICQCPC00
3.7	ISPWSD	
3.8	ISRUOLP	STATUS, OUTPUT, SUBMIT, ICQCPC00
3.9	ISPUCM	
3.11	ISRFMT	
3.12	ISRSSM	
3.13 and 3.15	ISRSEPRM	
3.14	ISRSFM	
3.16	ISRUTABL	
3.17	ISRUUDL	
4	ISRFPR	ASM, FORT, PLI, PLIC, LINK, TESTCOB, TESTFORT, CALL, SCRIPT, ALLOC, FREE, ICQCPC00
5	ISRJB1	SUBMIT
6	ISRPTC	All commands except those prohibited by ISPF in ISPTCM
7	ISPYXDR	
7.1	ISPYFI	
7.2	ISPYPI	
7.3	ISPYVI	
7.4	ISPYTI	
7.5	ISPYLI	
7.6	ISPYSI	
7.7.1	ISPYRFI	
7.7.2	ISPYRVI	
7.8	ISPYBI	
7.T	ISPTUTOR	
9	ISRALTDI	
10	FLMDDL	
10.1	FLMEB	
10.2	FLMED	
10.3	FLMUDU	
10.4	FLMBD	
10.5	FLMPD	
10.6	FLMPTC	
10.A	FLMA	

Table 10. ISPF Options and related module names (continued)

Option	Module Names	TSO Commands
11	ISRUDA	
X	ISPLLP	SUBMIT

Table 11 lists the ISPF commands with the related modules:

Table 11. ISPF commands

Command	Module Name
COLOR	ISPOPT
CUAATTR	ISPOPT
ENVIRON	ISPENV
EXHELP	ISPTUTOR
FKA	ISPOPF
HELP	ISPTUTOR
ISPFVAR	ISPISM
ISPLIBD	ISPLLS
ISPPREP	ISPPREP
ISRRLIST	ISRDSLST
ISRRROUTE	ISRRCL
KEYLIST	ISPKLU
KEYS	ISPOPT
KEYSHELP	ISPTUTOR
LIST	ISPLLP
LOG	ISPLLP
MSGID	ISPOMI
PANELID	ISPOPI
PFSHOW	ISPOPF
PSCOLOR	ISPOPT
REFACTD	ISRDSLST
REFACTL	ISRDSLST
REFADDD	ISRRSLST
REFADDL	ISRRSLST
REFLISTD	ISRDSLST
REFLISTL	ISRDSLST
REFOPEND	ISRDSLST
REFOPENL	ISRDSLST
SAREA	ISPSAM
SETTINGS	ISPISM
START	ISPSTRT
TSOCMD	ISRPTC
TUTOR	ISPTUTOR

Table 11. ISPF commands (continued)

Command	Module Name
ZKEYS	ISOPT

---

## Preallocation of List/Log data sets to SYSOUT

ISPF normally allocates the ISPF list and log data sets (sequential data sets) the first time a user requests printed output or takes action that generates log output. The user can control the printing and disposition of these data sets at ISPF termination and by issuing the ISPF LOG and LIST commands.

You can preallocate the list and log data sets directly to SYSOUT by including one or both of these DD statements in the TSO LOGON procedure:

```
//ISPLIST DD SYSOUT=A,  
//          DCB=(LRECL=121,BLKSIZE=1210,RECFM=FBA)  
  
//ISPLOG DD SYSOUT=A,  
//          DCB=(LRECL=125,BLKSIZE=129,RECFM=VA)
```

If you preallocate these data sets to SYSOUT, they are automatically printed when the user logs off TSO. Also, when the user exits from ISPF, the termination menu is bypassed. If the user reenters ISPF before logging off, any new output is added to the end of the SYSOUT data sets.

### Notes:

1. You cannot use ISPF option 7.5 to browse log data sets allocated to SYSOUT.
2. You can use the ISPF Log/List pop-up on the Settings panel to specify either the number of lines per page or to bypass logging altogether (by specifying zero primary pages). The rest of the information on these panels is ignored if the list and log data sets are allocated to SYSOUT.
3. The defaults for the list data set are LRECL=121, line length=120, RECFM=FBA, as they were in the previous licensed programs. However, you can use the Log/List pop-up on the Settings panel to change the characteristics of the list data set so screen images wider than 121 characters can be printed.
4. You cannot issue the LOG or LIST command to process a preallocated log or list data set.
5. The ISPF temporary data set default names associated with the ISPLOG ddname are SPFLOGx.LIST, where x=numeric value 0-9.
6. The ISPF temporary data set default names associated with the ISPLIST ddname are SPFx.LIST, where x=numeric value 0-9.

---

## Specifying the maximum number of split screens

ISPF can run up to 32 logical screens at one time. You can specify the maximum number of logical screens allowed for your installation by modifying the ISPF configuration table. See Chapter 2, "The ISPF Configuration Table," on page 9 for more information about modifying the ISPF configuration table. Set the value of keyword MAXIMUM\_NUMBER\_OF\_SPLIT\_SCREEN to any number from 4 to 32. The default value is 8. Be sure to consider your users' region size when you set the limit for the maximum number of screens.

---

## Setting ISPF site-wide defaults

You can set these site-wide defaults for ISPF by changing the ISPF configuration table:

- ISPF's Settings options
- CUA panel elements
- KEYLIST ON|OFF
- PFSHOW ON|OFF
- Log and List final disposition
- Default Primary panel
- SCROLL\_DEFAULT (CSR|DATA|HALF|MAX|PAGE)
- SCROLL\_MIN (minimum scroll value)
- SCROLL\_MAX (maximum scroll value)
- STATUS\_AREA\_DEFAULT (CAL|FUN|OFF|SES|UPS|USE)

For this list of defaults, except SCROLL and STATUS AREA, ISPF provides a force option which indicates the default set in the ISPF configuration table should be used, even though users may have a value set in their system or user profile. The SCROLL and STATUS AREA defaults are included in the ISPSPROF when the initial profile is built. The user can later override the site-wide default.

You can also set these site-wide defaults by changing the ISPF configuration table:

- Log, list, and temporary data set block size.
- Log and temporary data set logical record length.
- PRINTDS operands, DEST, or WRITER

This list of defaults cannot be overridden by the end user.

See Chapter 2, "The ISPF Configuration Table," on page 9 for more information about modifying the Configuration table.

---

## Customizing command tables

While running an application, you can use commands defined in eight different command tables. These command tables are:

- Application command table
- 3 User command tables
- 3 Site command tables
- System command table.

The user command tables and the site command tables are optional to use. They must be defined for your installation and present when ISPF is initialized. To make use of them, you must update the ISPF configuration table to include their application identification.

ISPF uses a specific order when searching the tables for commands you enter. However, you do have some control over the search order when using the optional site command tables. The site command tables can be searched either before or after the system command table. To define the search order relative to the site and system command tables, update the ISPF configuration table.

The keywords in the ISPF configuration table that determine the search order between the site command tables and the system command table, and whether or not user command tables and site command tables are defined, are:

### **APPLID\_FOR\_USER\_COMMAND\_TABLE**

The application ID for up to 3 user command tables. The default for each

is NONE (no user command tables). The user command tables are searched after the command table for the current application, that is, the command table for the current APPLID, and before the site-wide and default system command tables.

#### **APPLID\_FOR\_SITE\_COMMAND\_TABLE**

The application ID for up to 3 site command tables. The default for each is NONE (no site-wide command tables). The search order for the site command tables depends on the SITE\_COMMAND\_TABLE\_SEARCH\_ORDER\_SETTING.

#### **SITE\_COMMAND\_TABLE\_SEARCH\_ORDER**

Determines if the site-wide command tables are to be searched before or after the default ISPF command table. Valid values are BEFORE and AFTER. The default is BEFORE.

When you enter a command, the application command table is searched first. If the command is found, no further searching is necessary. If the command is not found in the application command table, up to 3 user command tables are searched. If the command is not found in the user command tables, up to 3 site command tables are searched or the system command table is searched (depending on the search order defined in the ISPF configuration table). Finally, if the command is still not found, the remaining command table(s), site or system, are searched. User command tables and site command tables are only searched if they have been defined in the ISPF configuration table and are present at ISPF initialization.

See Chapter 2, “The ISPF Configuration Table,” on page 9 for more information about modifying the ISPF configuration table.

## **Application command table**

Commands in the application command table are in effect only for the application you are running. Defining commands in the application command table lets you customize the set of commands you need for a particular application without redefining the system command table for each application.

ISPF provides a utility to create and modify the command tables. Enter ISPF option 3.9 from the ISPF Primary Option Menu. For more information about creating application command tables, refer to the *z/OS ISPF User's Guide Vol II* or the *z/OS ISPF Dialog Developer's Guide and Reference*.

## **User command tables**

Commands in up to 3 user command tables are in effect for all applications. These commands can override identically named commands in the site or system command tables, and are themselves overridden by identically named commands in the application command table.

You must specify the application ID of the user command tables in the ISPF configuration table, or the user command tables cannot be used. If you do not have a user command table that matches the application ID, the ID is ignored until the table is present and ISPF is reinitialized.

## **Site command tables**

Commands in up to 3 site command tables are in effect for all applications. An option in the ISPF configuration table enables a site to specify that up to 3 site command tables are searched before or after the system command table. If the site

command tables are searched first, then commands in the site command tables can override identically named commands in the system command table, and in turn can be overridden by identically named commands in the application or user command tables. If the site command tables are searched after the system command table, then their commands override no others, but can be overridden by identically named commands in either the application, user, or system command tables.

You must specify the application ID of up to 3 site command tables in the ISPF configuration table, or the site command tables cannot be used. If you do not have a site command table that matches the application ID, the ID is ignored until the table is present and ISPF is reinitialized.

## System command table

Commands in the system command table are in effect for all applications. However, these commands can be overridden by an identically named command in an application or user command table, or a site command table if it is defined to be searched first.

---

## Creating ISPF terminal translation tables

Creating ISPD terminal translation tables describes how to perform these tasks:

- Create a set of ISPF translation tables.
- Modify existing ISPF panels to use with the set of translation tables.

The ISP.SISPSAMP library includes sample assembler source programs, ISPOWNTT and ISPAPLTT. Use these as examples of what a completed module should look like. You can modify the sample module to suit your requirements and supply your own values for each of the translation tables.

ISPF uses these translation tables:

- 2-byte input translation table
- 2-byte output translation table
- Uppercase character translation table
- Lowercase character translation table
- Valid terminal output translation table
- Generic string master translation table
- Alphabetic character translation table
- Collating sequence translation table

The sample assembler modules include all of these translation tables. Each translation table consists of 32 consecutive DC instructions. Each DC instruction consists of eight hexadecimal values. You must supply the 256 hexadecimal values that make up each of the translation tables. The address of each table is at the start of the assembler module. Ignore addresses such as TBIP, which are set to zero.

The sample ISPOWNTT corresponds to an English 3278/3279 terminal except that the collating sequence translation table is not used in English.

**Note:** If the set of terminal translation tables has to support Katakana characters, you must perform these steps:

1. Change the DPRP pointer in the source from A(0) to A(TTDPR).
2. Rename table TTUPP to TTDPR.



Table 13. Valid terminal output translation table example

Table	Hexadecimal Code	Position
TTVAL	DC X'FFFFFFFFFFFFFFFF'	(X'00' to X'07')
	DC X'FFFFFFFFFFFFFFFF'	(X'08' to X'0F')
	DC X'FFFFFFFFFFFFFFFF'	(X'10' to X'17')
	...	
	DC X'FF00000000000000'	(X'78' to X'7F')
	DC X'FF00000000000000'	(X'80' to X'87')
	...	
	DC X'0000FFFFFFFFFFFF'	(X'E8' to X'EF')
	DC X'0000000000000000'	(X'F0' to X'F7')
	DC X'0000FFFFFFFFFFFF'	(X'F8' to X'FF')

## Generic string master translation table

The positions in the generic string master translation table (TTGSM in the example shown in Table 14) are filled in as follows:

- X'00' Blank character
- X'01' Invalid character
- X'02' Special character
- X'04' APL/TEXT special characters (only for APL and TEXT keyboards)
- X'08' APL/TEXT alphabetic characters (only for APL and TEXT keyboards)
- X'10' Lowercase alphabetic character
- X'20' Uppercase alphabetic character
- X'40' Numeric character
- X'80' User character subset

Table 14. Generic string master translation table example

Table	Hexadecimal Code	Position
TTGSM	DC X'0101010101010101'	(X'00' to X'07')
	DC X'0101010101010101'	(X'08' to X'0F')
	DC X'0101010101010101'	(X'10' to X'17')
	...	
	DC X'0102020202020202'	(X'78' to X'7F')
	DC X'0110101010101010'	(X'80' to X'87')
	...	
	DC X'2020010101010101'	(X'E8' to X'EF')
	DC X'4040404040404040'	(X'F0' to X'F7')
	DC X'4040010101010101'	(X'F8' to X'FF')

## Modifying the GSM to use the user character subset

The Generic String Master (GSM) translation table and its related tables can be modified to add an additional character subset to be used in picture string processing by the ISPF EDIT, FIND, and CHANGE commands.

These steps allow you to modify the GSM to use a character subset:

1. Choose which character is used to represent your subset. For example, Edit uses an @ to stand for alphabetic.
2. Modify the entry in the Generic String Special Character (GSS) table found in "Translation table for generic string special characters" on page 267 corresponding to the character you wish to use for a value of X'08'. This indicates where in the Generic String Character Code (GSC) table the mask for your character set is located. The GSC does not need to be changed as it is initially set for user character sets.
3. Modify the GSM entries of those characters you wish to include in your special character set so the high order bit is on.

## Alphabetic character translation tables

There are two alphabetic translation tables:

- TTALP, which includes the number sign (#), the dollar sign (\$), and the at sign (@) (Table 15).
- TTALB, which does not include the number sign (#), the dollar sign (\$), and the at sign (@) (Table 16).

**Note:** Valid alphabetic characters are represented with X'00'. For example, the hexadecimal position for an uppercase 'A' contains X'00'. Non-alphabetic characters are represented with X'FF'. For example, the hexadecimal position for a blank contains X'FF'.

Table 15. Sample alphabetic character translation table including #, \$, and @

Table	Hexadecimal Code	Position
TTALP	DC X'FFFFFFFFFFFFFFFF'	(X'00' to X'07')
	DC X'FFFFFFFFFFFFFFFF'	(X'08' to X'0F')
	DC X'FFFFFFFFFFFFFFFF'	(X'10' to X'17')
	...	
	DC X'FFFFFFFF0000FFFFFFFF'	(X'78' to X'7F')
	DC X'FF0000000000000000'	(X'80' to X'87')
	...	
	DC X'0000FFFFFFFFFFFFFFFF'	(X'E8' to X'EF')
	DC X'FFFFFFFFFFFFFFFF'	(X'F0' to X'F7')
	DC X'0000FFFFFFFFFFFFFFFF'	(X'F8' to X'FF')

Table 16. Sample alphabetic character translation table excluding #, \$, and @

Table	Hexadecimal Code	Position
TTALB	DC X'FFFFFFFFFFFFFFFF'	(X'00' to X'07')
	DC X'FFFFFFFFFFFFFFFF'	(X'08' to X'0F')
	DC X'FFFFFFFFFFFFFFFF'	(X'10' to X'17')
	...	
	DC X'FFFFFFFFFFFFFFFF'	(X'50' to X'57')
	DC X'FFFFFFFFFFFFFFFF'	(X'58' to X'5F')
	...	
	DC X'FFFFFFFFFFFFFFFF'	(X'78' to X'7F')
	DC X'FF0000000000000000'	(X'80' to X'87')
	DC X'0000FFFFFFFFFFFFFFFF'	(X'88' to X'8F')
	...	
	DC X'0000FFFFFFFFFFFFFFFF'	(X'E8' to X'EF')
	DC X'FFFFFFFFFFFFFFFF'	(X'F0' to X'F7')
	DC X'FFFFFFFFFFFFFFFF'	(X'F8' to X'FF')

## Collating sequence translation table

The collating sequence translation table (TTCOL in the example shown in Table 17 on page 90) contains the sort order of the 256 table entries, represented in hexadecimal (0-255). This table is used by TBSORT/TBADD services when the sort type is 'C', and by the ISPF Edit SORT command. See *z/OS ISPF Dialog Developer's Guide and Reference* for further information.

*Example 1:* If you want an 'A' to sort before a 'B', as shown in Table 17 on page 90, the hexadecimal position for an 'A' will contain a value that is less than the sort value found in the hexadecimal position for 'B'. Similarly, to sort a blank (X'40') before an 'A', the hexadecimal table position for a blank will contain a sort value less than that of the 'A'.

*Example 2:* If you want a blank (X'40') to sort last (not shown here), set the hexadecimal table position for a blank (X'40') to X'FF'.

Table 17 shows an example of how the table would look if you wanted to sort strictly on the basis of hexadecimal values (for example, 'a' before 'b', 'A' before 'A').

*Table 17. Collating sequence translation table example*

Table	Hexadecimal Code	Position
TTCOL	DC X'0001020304050607'	(X'00' to X'07')
	DC X'08090A0B0C0D0E0F'	(X'08' to X'0F')
	DC X'1011121314151617'	(X'10' to X'17')
	...	
	DC X'78797A7B7C7D7E7F'	(X'78' to X'7F')
	DC X'8081828384858687'	(X'80' to X'87')
	...	
	DC X'E8E9EAEBECEDEEEF'	(X'E8' to X'EF')
	DC X'F0F1F2F3F4F5F6F7'	(X'F0' to X'F7')
	DC X'F8F9FAFBFCFDFEFFF'	(X'F8' to X'FF')

If you want to sort strictly on the basis of hexadecimal codes (as ISPF does for English), set the pointer to the collating sequence table (COLP) to zero. In the case of a pure hexadecimal sort, ISPF does not require a table.

After you supply the values for these six tables, assemble and link-edit the module. After the load module is created, go to the next step.

## Specifying terminal types

Having created a new set of translation tables you must provide a way for users to select the appropriate terminal type to refer to it. The simplest way is for the user to select the option OTHER in the ISPF Settings panel (ISPISMMN) and then enter the name of the load module (see Settings (Option 0) in the *z/OS ISPF User's Guide Vol II*). The load module is a member of a partitioned data set that is either allocated to the ISPLLIB ddname, or exists in STEPLIB, JOBLIB, or LINKLIB.

Validity checks are conducted on the load module to help prevent a user from loading bad translation tables. This could cause ISPF to fail until the faulty specification is removed by manually editing or deleting the user system profile. These checks are performed:

- verify that the module name matches an 8-byte constant at the start of the module (as in the sample program ISPOWNNTT)
- verify that the uppercase translate table immediately follows the address pointer for the collating sequence translate table
- verify that the uppercase and lowercase translate tables are 256 bytes long

If the load of the new translation tables fails, ISPF reverts to the previous terminal type setting.

Older methods for specifying translation tables are still supported. See these topics for more information:

- "Changing the DTL source for ISPISMMN" on page 91
- "Changing the DTL source for ISPOPTxx panels" on page 93
- "Invoking ISPTTDEF" on page 94

## Changing the DTL source for ISPISMMN

To change the ISPISMMN Dialog Tag Language source, use the Edit option to update the DTL source file members ISPZMMCH and ISPZMMSO. The member ISPISMMN defines the basic panel. This panel does not have to be modified but it does have to be reconverted with ISPDTLC after the changes to the imbed members ISPZMMCH and ISPZMMSO are complete.

The last <SELFLD tag in DTL source member ISPISMMN (see Figure 49) defines the list of terminal types. Four columns of choices are specified. The number of choices in each column is calculated by the conversion utility. The choice definitions are found in file imbed ISPZMMCH.

```
<selfld type=single name=ztm pmtloc=before listtype=ddlist
      required=yes msg=ispo901 help=ispo901h autotab=no
      entwidth=2 selfmt=end selwidth=60 choicecols=4 choicedepth=*>
      &selfld_3_prompt;

&ispzmmch;          <!-- include CHOICE tags for terminal types -->

</selfld>
```

Figure 49. DTL source for terminal type selection - SELFLD tag

This example illustrates adding terminal type XXXX using module ISPOWNTT to panel ISPISMMN. Alternatively, you can replace an existing set of translation tables by typing over the terminal type and the name of the load module that the newly defined set replaces.

The modification to the DTL source member ISPZMMCH adds the terminal type selection to the panel display (see Figure 50) and creates panel logic to determine which selection number will match a certain terminal type.

Terminal Type	3	1. 3277	2. 3277A	3. 3278	4. 3278A
		5. 3290A	6. 3278T	7. 3278CF	8. 3277KN
		9. 3278KN	10. 3278AR	11. 3278CY	12. 3278HN
		13. 3278HO	14. 3278IS	15. 3278L2	16. BE163
		17. BE190	18. 3278TH	19. 3278CU	20. DEU78
		21. DEU78A	22. DEU78T	23. DEU90A	24. SW116
		25. SW131	26. SW500	27. 3278GR	28. 3278L1
		29. OTHER	30. XXXX		

Figure 50. Example: adding a terminal type to panel ISPISMMN

Modified DTL source for the English-language section of the member ISPZMMCH follows:

```

:
<condexec lang=english>
  <choice selchar=1  checkvar=zterm  match=3277>3277      <:-- 01 -->
  <choice selchar=5  checkvar=zterm  match=3290A>3290A     <:-- 02 -->
  <choice selchar=9   checkvar=zterm  match=3278KN>3278KN  <:-- 03 -->
  <choice selchar=13  checkvar=zterm  match=3278HO>3278HO  <:-- 04 -->
  <choice selchar=17  checkvar=zterm  match=BE190>BE190    <:-- 05 -->
  <choice selchar=21  checkvar=zterm  match=DEU78A>DEU78A  <:-- 06 -->
  <choice selchar=25  checkvar=zterm  match=SW131>SW131    <:-- 07 -->
  <choice selchar=2   checkvar=zterm  match=3277A>3277A    <:-- 08 -->
  <choice selchar=6   checkvar=zterm  match=3278T>3278T    <:-- 09 -->
  <choice selchar=10  checkvar=zterm  match=3278AR>3278AR  <:-- 10 -->
  <choice selchar=14  checkvar=zterm  match=3278IS>3278IS  <:-- 11 -->
  <choice selchar=18  checkvar=zterm  match=3278TH>3278TH  <:-- 12 -->
  <choice selchar=22  checkvar=zterm  match=DEU78T>DEU78T  <:-- 13 -->
  <choice selchar=26  checkvar=zterm  match=SW500>SW500    <:-- 14 -->
  <choice selchar=3   checkvar=zterm  match=3278>3278      <:-- 15 -->
  <choice selchar=7   checkvar=zterm  match=3278CF>3278CF  <:-- 16 -->
  <choice selchar=11  checkvar=zterm  match=3278CY>3278CY  <:-- 17 -->
  <choice selchar=15  checkvar=zterm  match=3278L2>3278L2  <:-- 18 -->
  <choice selchar=19  checkvar=zterm  match=3278CU>3278CU  <:-- 19 -->
  <choice selchar=23  checkvar=zterm  match=DEU90A>DEU90A  <:-- 20 -->
  <choice selchar=27  checkvar=zterm  match=3278GR>3278GR  <:-- 21 -->
  <choice selchar=4   checkvar=zterm  match=3278A>3278A    <:-- 22 -->
  <choice selchar=8   checkvar=zterm  match=3277KN>3277KN  <:-- 23 -->
  <choice selchar=12  checkvar=zterm  match=3278HN>3278HN  <:-- 24 -->
  <choice selchar=16  checkvar=zterm  match=BE163>BE163    <:-- 25 -->
  <choice selchar=20  checkvar=zterm  match=DEU78>DEU78    <:-- 26 -->
  <choice selchar=24  checkvar=zterm  match=SW116>SW116    <:-- 27 -->
  <choice selchar=28  checkvar=zterm  match=3278L1>3278L1  <:-- 28 -->
  <choice selchar=30  checkvar=zterm  match=XXXX>XXXX      <:-- 29 -->
</condexec>
:

```

Figure 51. DTL source for terminal type selection - CHOICE tags

The conversion utility formats CHOICE tags in a top-to-bottom, left-to-right order, placing the first seven CHOICE tags in column 1 (choice numbers 1, 5, 9, 13, 17, 21, and 25), CHOICE tags 8 through 14 in column 2 (selection numbers 2, 6, 10, 14, 18, 22, and 26), and so on. The number of entries in each column is based on 28 total lines of CHOICE and CHDIV tags divided by the specified number of choice columns, in our example, 4 (as defined in Figure 49 on page 91 by the CHOICECOLS keyword). These tags are arranged so that the choices appear in a left-to-right, top-to-bottom order).

By modifying the DTL source member ISPZMMSO, the new terminal type XXXX will be associated with its load module name ISPOWNNTT. Modified DTL source for the English-language section of the member ISPZMMSO follows:

```

/* set translate load module name based on terminal type */
:
:
&ZCHARLM = TRANS(&ZTERM
                3277 , ISP3277
                3277A , ISP3277A
</source>
<condexec lang=english
<source>
                3278 , ISP3278
                3278A , ISP3278A
                3290A , ISP3278A
                3278T , ISP3278T
</source>
</condexec>
<source>
                3278CF, ISP3278C
                3277KN, ISP3277K
                3278KN, ISP3278K
                3278AR, ISPAR78
                3278CY, ISPCY78
                3278HN, ISPHN78
                3278HO, ISPHO78
                3278IS, ISPIS78
                3278L2, ISPL278
                BE163, ISPB678
                BE190, ISPB978
                3278TH, ISPTH78
                3278CU, ISPCU78
                UE3278, ISP3278
                UE3278A, ISP3278A
                UE3290A, ISP3278A
                UE3278T, ISP3278T
                DEU78, ISPG78
                DEU78A, ISPG78A
                DEU90A, ISPG78A
                DEU78T, ISPG78T
                SW116 , ISPSW116
                SW131 , ISPSW131
                SW500 , ISPSW500
                3278L1, ISPL178
                3278GR, ISPGR78
                OTHER, ISPOTHR
                XXXX,  ISPOWNTT)
</source>

```

Figure 52. DTL source for valid terminal types and associated load module names

In Figure 52, the left entries (for example 3277, 3277A and 3278) are valid terminal types a user can specify. The right entries (for example ISP3277, ISP3277A and ISP3278) are the associated load module names.

The delivered ISPF terminal table names start with the prefix “ISP”. ISPF does not require that user-defined terminal table names begin with the prefix “ISP”; however, PDF terminal names require the “ISR” prefix. PDF searches for the load module beginning with the fourth position of the actual table name and prefixes it with “ISR”. See “Creating PDF translation tables” on page 129 for a discussion of PDF translation tables.

### Changing the DTL source for ISPOPTxx panels

If you still use the old options panels ISPOPT1 or ISPOPT1A, use ISPF Edit to update the DTL source file imbed member ISPZPTTT. ISPZPTTT is used by both ISPOPT1 and ISPOPT1A. It is a conversion-language-sensitive file that includes both the terminal type verification statement and the terminal type load module selection translation construct. ISPZPTTT is distributed in ISP.SISPGMLI.

After this member is updated, run ISPD TLC to convert all of the listed panels.

To add a new terminal type XXXX using load module ISPOWNTT to these panels, update ISPZPTTT as follows:

```
<source type=proc>
  VER (&ZTERM NB LIST
      3277,3277A,3278,3278A,3278T,3278CF,3277KN,3278KN,3290A,3278AR,
      BE163,BE190,3278CY,3278HN,3278HO,3278IS,3278L2,3278TH,3278CU,
      DEU78,DEU78A,DEU90A,SW116,SW131,SW500,3278L1,DEU78T,3278GR,
      UE3278,UE3278A,UE3290A,UE3278T,XXXX
      MSG=ISPO004)

  /* set translate load module name based on terminal type */
  &ZCHARLM = TRANS(&ZTERM
      3277 , ISP3277
      3277A , ISP3277A
      3278 , ISP3278
      3278A , ISP3278A
      3290A , ISP3278A
      3278T , ISP3278T
      3278CF, ISP3278C
      3277KN, ISP3277K
      3278KN, ISP3278K
      3278AR, ISPAR78
      3278CY, ISPCY78
      3278HN, ISPHN78
      3278HO, ISPHO78
      3278IS, ISPIS78
      3278L2, ISPL278
      BE163, ISPB678
      BE190, ISPB978
      3278TH, ISPTH78
      3278CU, ISPCU78
      UE3278, ISP3278
      UE3278A, ISP3278A
      UE3290A, ISP3278A
      UE3278T, ISP3278T
      DEU78, ISPGE78
      DEU78T, ISPGE78T
      DEU78A, ISPGE78A
      DEU90A, ISPGE78A
      SW116 , ISPSW116
      SW131 , ISPSW131
      SW500 , ISPSW500
      3278GR, ISPGR78
      3278L1, ISPL178
      OTHER, ISPOTHR
      XXXX, ISPOWNTT)
</source>
```

Figure 53. ISPZPTTT modified to add new terminal type XXXX

## Invoking ISPTTDEF

The ISPTTDEF program offers you an alternative approach to specifying the terminal type or the corresponding set of translation tables (the load module). You can invoke ISPTTDEF from a selection panel, command table, or dialog function. To invoke the program, enter:

```
SELECT PGM(ISPTTDEF) PARM(xxx)
```

where *xxx* is the terminal type (one of the types listed on the distributed panel ISPISMMN) or the name of the load module. When you specify a terminal type,

the ISPTTDEF program loads and uses the appropriate load module for that terminal type. If you specify a load module, the program attempts to load a module with that name.

## Creating ISPF code page translation tables

ISPF supports extended code pages that allow ISPF to display panels, messages, and variable application data correctly on terminals using any of the supported code pages. For example, ISPF can display a German panel on a French CECP (Country Extended Code Page) terminal, with all common characters displayed correctly. Any characters in the panel that do not exist in the terminal code page are displayed as periods.

The code page and character set are specified by CCSID (Coded Character Set Identifier) as defined by Character Data Representation Architecture (CDRA). ISPF supports these EXTENDED CODE PAGE CCSIDs for the TRANS service and also with the use of the CCSID keyword on panels and messages.

*Table 18. Extended CCSID1 Supported*

CCSID	Character Set	Code Page	Country/Language
00037	697	37	U.S.A. Canada Netherlands Portugal Brazil Australia New Zealand
00273	697	273	Austria Germany
00277	697	277	Denmark Norway
00278	697	278	Finland Sweden
00280	697	280	Italy
00284	697	284	Spain L.A. Spanish
00285	697	285	United Kingdom
00297	697	297	France
00420	235	420	Arabic
00424	941	424	Hebrew
00500	697	500	Switzerland Belgium
00838	1176	838	Thailand
00870	959	870	Latin-2
00871	697	871	Iceland
00875	923	875	Greece
00880	960	880	Cyrillic
01025	1150	1025	Cyrillic
01026	1126	1026	Turkey
01047	697	1047	Latin1

Table 18. Extended CCSID1 Supported (continued)

CCSID	Character Set	Code Page	Country/Language
01123	1326	1123	Ukraine

Table 19. Extended CCSID1 Supported (EURO)

CCSID	Character Set	Code Page	Country/Language
00924	1353	0924	Latin9
01140	695	1140	U.S.A. Canada Netherlands Portugal Brazil Australia New Zealand
01141	695	1141	Austria Germany
01142	695	1142	Denmark Norway
01143	695	1143	Finland Sweden
01144	695	1144	Italy
01145	695	1145	Spain L.A. Spanish
01146	695	1146	United Kingdom
01147	695	1147	France
01148	695	1148	Switzerland Belgium
01149	695	1149	Iceland
01153	1375	1153	Latin2
01154	1381	1154	Cyrillic
01155	1378	1155	Turkey
01158	1388	1158	Ukraine
01160	1395	1160	Thailand
04899	1356	0803	Hebrew
04971	1371	0875	Greece
12712	1357	0424	Hebrew
16804	1461	0420	Arabic

These Extended CCSIDs (shown in Table 20 on page 97) are also supported for panels and messages that specify an extended code page. These are the mixed SBCS/DBCS CCSIDs for these languages.

Japanese (Katakana) and Simplified Chinese EXTENDED CODE PAGES are not supported on any terminal but these EXTENDED CODE PAGES are supported for the TRANS service and with the CCSID keyword on panels and messages.

Table 20. Extended SBCS and DBCS CCSIDs Supported

CCSID	Character Set	Code Page	Country
00930	1172	290	Japanese (Katakana)
00939	1172	1027	Japanese (Latin)
00933	1173	833	Korean
00935	1174	836	Simplified Chinese
00937	1175	037	Traditional Chinese
01159	65535	1159	Traditional Chinese
01364	65535	0834	Korean
01371	65535	0835	Traditional Chinese
01388	65535	0837	Simplified Chinese
01390	65535	0300	Japanese
01399	65535	0300	Japanese
05123	65535	1027	Japanese
08482	65535	0290	Japanese

## Base code pages for terminals

ISPF provides direct translation between each BASE CODE PAGE and its EXTENDED CODE PAGE for panels or messages. It also provides direct translation between extended Japanese (Latin or English) and both base Japanese (English) and base Japanese (Katakana). All translation between the single-byte EXTENDED CODE PAGEs for the double-byte languages and the CECP code pages is through CCSID 00500.

ISPF supports the base code pages (including mixed SBCS/DBCS CCSIDs for the DBCS languages) shown in Table 21.

Table 21. Base CCSIDs Supported

CCSID	Character Set	Code Page	Country/Language
00803	1147	424	Hebrew (Old)
00931	101	037	Japan (English)
04369	265	273	Germany and Austria
04371	273	275	Brazil
04373	281	277	Denmark and Norway
04374	285	278	Finland and Sweden
04376	293	280	Italy
04380	309	284	L.A. (Spanish Speaking)
04381	313	285	U.K. English
04393	1129	297	France
04934	938	838	Thailand
04966	959	870	Latin-2
04976	960	880	Cyrillic
05029	933	833	Korean
05031	936	836	Simplified Chinese

Table 21. Base CCSIDs Supported (continued)

CCSID	Character Set	Code Page	Country/Language
05033	101	037	Traditional Chinese
08229	101	037	U.S. English and Netherlands
08476	650	284	Spain
09122	332	290	Japan (Katakana)
41460	904	500	Switzerland
45556	908	500	Switzerland

## ISPCCSID translation load modules

ISPCCSID translation load modules translate data from one CCSID to another. There is one translation load module for each of the supported CCSIDs. The name, or alias, of each CCSID translation load module is made up of a 5-digit CCSID, prefixed with "ISP". For example, load module ISP00111 supports translation of the CCSID 00111. Each CCSID translation load module must contain at least two translation tables. These translation tables convert data between the respective CCSID and CCSID 00500. In addition each CCSID load module can contain up to 256 pairs of optional "direct" translation tables. ISPF uses the direct translation tables when available. Otherwise, ISPF translates the characters through CCSID 00500. Translating through CCSID 00500 can result in valid characters being lost as CCSID 00500 does not have all possible code points defined.

You can add direct "To" and "From" translation tables for direct translation to prevent possible loss of characters through CCSID 00500 for character sets other than 697 or to augment the extended code page translation tables provided by ISPF. The direct translation CCSID must be one of the CCSIDs supported by ISPF (see "Extended code page translation tables provided by ISPF" on page 99) or added by the user.

Both "To" and "From" translation tables must be provided for direct translation tables as well as CCSID 00500 tables, even though there might be no translation needed. For example, to translate from a base CCSID to an extended CCSID for the same code page, all characters will translate to themselves.

## Adding translation tables for extended code page support

You can provide support for additional code pages by creating or modifying translation tables using the sample assembler module ISPEXCP in the ISP.SISPSAMP library.

Any translation tables that are added must be named ISP`nnnnnn`, where `nnnnnn` is the CCSID, and must be a CCSID defined in the Character Data Representation Architecture Registry. This CCSID must be different from any of the supported CCSIDs. The translation tables should include code points X'40' through X'FE'.

Table 22 on page 99 and Table 23 on page 99 show examples of the "To" and "From" translation tables needed to translate characters between CCSID 00500 and CCSID 00037.

Table 22. Table for translating from CCSID 00037 to CCSID 00500.

Table	Hexadecimal Code	Position
TO_500	DC X'4041424344454647'	(X'40' to X'47')
	DC X'4849B04B4C4D4EBB'	(X'48' to X'4F')
	DC X'5051525354555657'	(X'50' to X'57')
	DC X'58594F5B5C5D5EBA'	(X'58' to X'5F')
	DC X'78797A7B7C7D7E7F'	(X'78' to X'7F')
	DC X'8081828384858687'	(X'80' to X'87')
	DC X'E8E9EAEBECEDEEEF'	(X'E8' to X'EF')
	DC X'F0F1F2F3F4F5F6F7'	(X'F0' to X'F7')
	DC X'F8F9FAFBFCFDFE'	(X'F8' to X'FE')

Table 23. Table for translating from CCSID 00500 to CCSID 00037.

Table	Hexadecimal Code	Position
FROM_500	DC X'4041424344454647'	(X'40' to X'47')
	DC X'4849BA4B4C4D4E5A'	(X'48' to X'4F')
	DC X'5051525354555657'	(X'50' to X'57')
	DC X'5859BB5B5C5D5EB0'	(X'58' to X'5F')
	DC X'78797A7B7C7D7E7F'	(X'78' to X'7F')
	DC X'8081828384858687'	(X'80' to X'87')
	DC X'E8E9EAEBECEDEEEF'	(X'E8' to X'EF')
	DC X'F0F1F2F3F4F5F6F7'	(X'F0' to X'F7')
	DC X'F8F9FAFBFCFDFE'	(X'F8' to X'FE')

The source for these modules is provided in ISPEXCP, in the ISP.SISPSAMP library.

### Extended code page translation tables provided by ISPF

ISPF provides the translation tables shown in Table 24, which you can update. They are distributed in ISP.SISPSAMP.

Table 24. Translation tables provided with ISPF

Table name	CCSID	Description
ISPSTC1	00037	U.S.A, Canada, Netherlands, Portugal, Brazil, Australia, and New Zealand
ISPSTC2	00273	Austria and Germany
ISPSTC3	00277	Denmark and Norway
ISPSTC4	00278	Finland and Sweden
ISPSTC5	00280	Italy
ISPSTC6	00284	Spain and L.A. (Spanish-speaking)
ISPSTC7	00285	United Kingdom
ISPSTC8	00297	France
ISPSTC9	00500	Switzerland and Belgium
ISPSTC10	00939	Japan (Latin)
ISPSTC11	00930	Japan (Katakana)
ISPSTC12	00933	Korea
ISPSTC13	00935	Simplified Chinese
ISPSTC14	00937	Traditional Chinese

Table 24. Translation tables provided with ISPF (continued)

Table name	CCSID	Description
ISPSTC15	00870	Latin 2
ISPSTC16	00880	Cyrillic
ISPSTC17	01025	Cyrillic

### ISPCCSID translation load module generation macro

You can use the assembler macro, ISPCCSID, to generate custom ISPCCSID translation load modules. The macro also allows you to add “direct” translation tables to the ISPCCSID translation load modules ISPF supplies with the product. Calls to this macro must also be coded for the To\_500 and From\_500 tables and any “To” and “From” tables for direct translation. The load module must have either the name ISPxxxxx (where xxxxx is new CCSID) or an alias of ISPxxxxx. In both cases, the load module should be a CCSID defined in the Character Data Representation Architecture Registry.

Note that only the values for the hexadecimal digits X'40' through X'FE' are defined in a given translation table. These are the only code points that vary from CCSID to CCSID.

The first time you use the ISPCCSID macro, you must identify the CCSID of the ISPCCSID translation load module and provide the addresses of the “To” and “From” CCSID 00500 translation tables.

You can use the ISPCCSID macro again with the same ISPCCSID translation load module generation to identify the CCSID and translation table addresses of optional direct “To” and “From” translation tables.

The format of calls to the ISPCCSID assembler macro is:

```
ISPCCSID CCSID=nnnnn,TO=to-address,FROM=from-address
```

The required parameters of the ISPCCSID macro are:

*nnnnn*

This parameter is a 5-digit decimal (5 characters) number that specifies a CCSID. The *nnnnn* value on the first or only ISPCCSID macro definition is the CCSID associated with the ISPCCSID translation load module. The *nnnnn* value on other than the first ISPCCSID macro definition is the CCSID associated with direct “To” and “From” translation tables. If this parameter is not 5 digits, it causes an assembly error.

*to-address*

On the first or only ISPCCSID macro definition, this parameter specifies the address of the translation table that converts data from the CCSID associated with the respective ISPCCSID translation load module to CCSID 00500. On subsequent ISPCCSID macro definitions within the same ISPCCSID translation load module, this parameter specifies the address of the translation table that converts data from the CCSID associated with the respective ISPCCSID translation load module to the CCSID specified on this ISPCCSID macro definition.

*from-address*

On the first or only ISPCCSID macro definition, this parameter specifies the address of the translation table that converts data from CCSID 00500 to the CCSID associated with the respective ISPCCSID translation load module. On

subsequent ISPCCSID macro definitions within the same ISPCCSID translation load module, this parameter specifies the address of the translation table that converts data from the CCSID specified on this ISPCCSID macro definition to the CCSID associate with the respective ISPCCSID translation load module.

### ISPCCSID translation load module definition examples

Each ISPCCSID translation load module must be compiled separately using High Level Assembler (or a functional equivalent).

This example shows the ISPCCSID macro used with the Basic ISP00111 translation module.

```

                ISPCCSID CCSID=00111,TO=TRT0500,FROM=TRFR500
*
*
TRT0500  DC    XL191'...           00111 TO 00500
TRFR500  DC    XL191'...           00111 FROM 00500 (00500 TO 00111)
                END

```

This example shows the ISPCCSID macro used with the ISP00222 translation module with two direct CCSID entries.

```

                ISPCCSID CCSID=00222,TO=TRT0500,FROM=TRFR500
                ISPCCSID CCSID=00333,TO=TRT00333,FROM=TRF00333
                ISPCCSID CCSID=00444,TO=TRT00444,FROM=TRF00444
*
*
TRT0500  DC    XL191'...           00222 TO 00500
TRFR500  DC    XL191'...           00222 FROM 00500 (00500 TO 00222)
*
*
TRT00333 DC    XL191'...           00222 TO 00333

```

### Example of user-modifiable ISPF translation table

This is the module for CCSID 00037 (ISPSTC1). The existing tables can be modified, or more pairs of direct translation tables can be added. To add direct translation tables, add a new ISPCCSID macro call for the new direct translation tables, and add the new tables. The assembler program should be renamed to ISPTTC $nn$ , where  $nn$  is the last 1-digit or 2-digit number of the ISPSTC $nn$  name. For example, ISPSTC1 should be renamed ISPTTC1, and ISPSTC14 renamed ISPTTC14.

```

* THESE MACROS WILL GENERATE THE CCSID 00037 MODULE.
*
*
                ISPCCSID CCSID=00037,TO=TTC1T5H,FROM=TTC1F5H
                ISPCCSID CCSID=08229,TO=TTC1TB1,FROM=TTC1FB2
                ISPCCSID CCSID=04371,TO=TTC1TB2,FROM=TTC1FB2
*
*   TTC1T5H - CCSID 00037 TO CCSID 00500 Table
*
TTC1T5H  DS    0XL191
                DC X'401424344454647'           (X'40' TO X'47')
                DC X'4849B04B4C4D4EBB'         (X'48' TO X'4F')
                DC X'5051525354555657'         (X'50' TO X'57')
                DC X'58594F5B5C5D5E5A'         (X'58' TO X'5F')
                DC X'6061626364656667'         (X'60' TO X'67')
                DC X'68696A6B6C6D6E6F'         (X'68' TO X'6F')
                DC X'7071727374757677'         (X'70' TO X'77')
                DC X'78797A7B7C7D7E7F'         (X'78' TO X'7F')
                DC X'8081828384858687'         (X'80' TO X'87')
                DC X'88898A8B8C8D8E8F'         (X'88' TO X'8F')
                DC X'9091929394959697'         (X'90' TO X'97')
                DC X'98999A9B9C9D9E9F'         (X'98' TO X'9F')
                DC X'A0A1A2A3A4A5A6A7'         (X'A0' TO X'A7')

```

DC X'A8A9AAABACADAEAF'	(X'A8' TO X'AF')
DC X'5FB1B2B3B4B5B6B7'	(X'B0' TO X'B7')
DC X'B8B94A5ABCDBEBF'	(X'B8' TO X'BF')
DC X'C0C1C2C3C4C5C6C7'	(X'C0' TO X'C7')
DC X'C8C9CACBCCDCECF'	(X'C8' TO X'CF')
DC X'D0D1D2D3D4D5D6D7'	(X'D0' TO X'D7')
DC X'D8D9DADBDCDDDEDF'	(X'D8' TO X'DF')
DC X'E0E1E2E3E4E5E6E7'	(X'E0' TO X'E7')
DC X'E8E9EAEBECEDEEEF'	(X'E8' TO X'EF')
DC X'F0F1F2F3F4F5F6F7'	(X'F0' TO X'F7')
DC X'F8F9FAFBFCFDFE'	(X'F8' TO X'FE')

\*  
\* TTC1F5H - CCSID 00037 FROM CCSID 00500 Table  
\*

TTC1F5H DS 0XL191	
DC X'4041424344454647'	(X'40' TO X'47')
DC X'4849BA4B4C4D4E5A'	(X'48' TO X'4F')
DC X'5051525354555657'	(X'50' TO X'57')
DC X'5859BB5B5C5D5EB0'	(X'58' TO X'5F')
DC X'6061626364656667'	(X'60' TO X'67')
DC X'68696A6B6C6D6E6F'	(X'68' TO X'6F')
DC X'7071727374757677'	(X'70' TO X'77')
DC X'78797A7B7C7D7E7F'	(X'78' TO X'7F')
DC X'8081828384858687'	(X'80' TO X'87')
DC X'88898A8B8C8D8E8F'	(X'88' TO X'8F')
DC X'9091929394959697'	(X'90' TO X'97')
DC X'98999A9B9C9D9E9F'	(X'98' TO X'9F')
DC X'A0A1A2A3A4A5A6A7'	(X'A0' TO X'A7')
DC X'A8A9AAABACADAEAF'	(X'A8' TO X'AF')
DC X'4AB1B2B3B4B5B6B7'	(X'B0' TO X'B7')
DC X'B8B95F4FBCDBEBF'	(X'B8' TO X'BF')
DC X'C0C1C2C3C4C5C6C7'	(X'C0' TO X'C7')
DC X'C8C9CACBCCDCECF'	(X'C8' TO X'CF')
DC X'D0D1D2D3D4D5D6D7'	(X'D0' TO X'D7')
DC X'D8D9DADBDCDDDEDF'	(X'D8' TO X'DF')
DC X'E0E1E2E3E4E5E6E7'	(X'E0' TO X'E7')
DC X'E8E9EAEBECEDEEEF'	(X'E8' TO X'EF')
DC X'F0F1F2F3F4F5F6F7'	(X'F0' TO X'F7')
DC X'F8F9FAFBFCFDFE'	(X'F8' TO X'FE')

\*  
\* TTC1TB1 - CCSID 00037 TO CCSID 08229 Table  
\*

TTC1TB1 DS 0XL191	
DC X'404B4B4B4B4B4B4B'	(X'40' TO X'47')
DC X'484B4A4B4C4D4E4F'	(X'48' TO X'4F')
DC X'504B4B4B4B4B4B4B'	(X'50' TO X'57')
DC X'4B4B5A5B5C5D5E5F'	(X'58' TO X'5F')
DC X'60614B4B4B4B4B4B'	(X'60' TO X'67')
DC X'4B4B6A6B6C6D6E6F'	(X'68' TO X'6F')
DC X'4B4B4B4B4B4B4B4B'	(X'70' TO X'77')
DC X'4B797A7B7C7D7E7F'	(X'78' TO X'7F')
DC X'4B81828384858687'	(X'80' TO X'87')
DC X'88894B4B4B4B4B4B'	(X'88' TO X'8F')
DC X'4B91929394959697'	(X'90' TO X'97')
DC X'98994B4B4B4B4B4B'	(X'98' TO X'9F')
DC X'4BA1A2A3A4A5A6A7'	(X'A0' TO X'A7')
DC X'A8A94B4B4B4B4B4B'	(X'A8' TO X'AF')
DC X'4B4B4B4B4B4B4B4B'	(X'B0' TO X'B7')
DC X'4B4B4B4B4B4B4B4B'	(X'B8' TO X'BF')
DC X'C0C1C2C3C4C5C6C7'	(X'C0' TO X'C7')
DC X'C8C94B4B4B4B4B4B'	(X'C8' TO X'CF')
DC X'D0D1D2D3D4D5D6D7'	(X'D0' TO X'D7')
DC X'D8D94B4B4B4B4B4B'	(X'D8' TO X'DF')
DC X'E04BE2E3E4E5E6E7'	(X'E0' TO X'E7')
DC X'E8E94B4B4B4B4B4B'	(X'E8' TO X'EF')
DC X'F0F1F2F3F4F5F6F7'	(X'F0' TO X'F7')
DC X'F8F94B4B4B4B4B4B'	(X'F8' TO X'FE')

\*  
\* TTC1FB1 - CCSID 00037 FROM CCSID 08229 Table

```

*
TTC1FB1 DS 0XL191
DC X'4041424344454647' (X'40' TO X'47')
DC X'48494A4B4C4D4E4F' (X'48' TO X'4F')
DC X'5051525354555657' (X'50' TO X'57')
DC X'58595A5B5C5D5E5F' (X'58' TO X'5F')
DC X'6061626364656667' (X'60' TO X'67')
DC X'68696A6B6C6D6E6F' (X'68' TO X'6F')
DC X'7071727374757677' (X'70' TO X'77')
DC X'78797A7B7C7D7E7F' (X'78' TO X'7F')
DC X'8081828384858687' (X'80' TO X'87')
DC X'88898A8B8C8D8E8F' (X'88' TO X'8F')
DC X'9091929394959697' (X'90' TO X'97')
DC X'98999A9B9C9D9E9F' (X'98' TO X'9F')
DC X'A0A1A2A3A4A5A6A7' (X'A0' TO X'A7')
DC X'A8A9AAABACADAFAF' (X'A8' TO X'AF')
DC X'B0B1B2B3B4B5B6B7' (X'B0' TO X'B7')
DC X'B8B9BABBBBCDBEBF' (X'B8' TO X'BF')
DC X'C0C1C2C3C4C5C6C7' (X'C0' TO X'C7')
DC X'C8C9CACBCCDCECF' (X'C8' TO X'CF')
DC X'D0D1D2D3D4D5D6D7' (X'D0' TO X'D7')
DC X'D8D9DADBDCDDDEDF' (X'D8' TO X'DF')
DC X'E0E1E2E3E4E5E6E7' (X'E0' TO X'E7')
DC X'E8E9EAEBECEDEEEF' (X'E8' TO X'EF')
DC X'F0F1F2F3F4F5F6F7' (X'F0' TO X'F7')
DC X'F8F9FAFBFCFDFF' (X'F8' TO X'FE')

```

\*  
\* TTC1TB2 - CCSID 00037 TO CCSID 04371 Table

```

*
TTC1TB2 DS 0XL191
DC X'404B4B4B4B4B794B' (X'40' TO X'47')
DC X'4B4B4B4B4B4D4E4B' (X'48' TO X'4F')
DC X'50D04B4B4B4B4B4B' (X'50' TO X'57')
DC X'4B4B4B4B4B4B4B4B' (X'58' TO X'5F')
DC X'60614B4B4B4B7C4B' (X'60' TO X'67')
DC X'5B4B4B4B6C6D6E6F' (X'68' TO X'6F')
DC X'4B4A4B4B4B4B4B4B' (X'70' TO X'77')
DC X'4B4B7A4B4B7D7E7F' (X'78' TO X'7F')
DC X'4B81828384858687' (X'80' TO X'87')
DC X'88894B4B4B4B4B4B' (X'88' TO X'8F')
DC X'4B91929394959697' (X'90' TO X'97')
DC X'98994B4B4B4B4B4B' (X'98' TO X'9F')
DC X'4BA1A2A3A4A5A6A7' (X'A0' TO X'A7')
DC X'A8A94B4B4B4B4B4B' (X'A8' TO X'AF')
DC X'5F44B4BB4B4B4B4B' (X'B0' TO X'B7')
DC X'4B4B4B4B4B4B4B4B' (X'B8' TO X'BF')
DC X'4BC1C2C3C4C5C6C7' (X'C0' TO X'C7')
DC X'C8C94B4B4B4B4BC0' (X'C8' TO X'CF')
DC X'4BD1D2D3D4D5D6D7' (X'D0' TO X'D7')
DC X'D8D94B4B4B4B4B4B' (X'D8' TO X'DF')
DC X'E04BE2E3E4E5E6E7' (X'E0' TO X'E7')
DC X'E8E94B4B4B4B4B7B' (X'E8' TO X'EF')
DC X'F0F1F2F3F4F5F6F7' (X'F0' TO X'F7')
DC X'F8F94B4B4B4B4B' (X'F8' TO X'FE')

```

\*  
\* TTC1FB2 - CCSID 00037 FROM CCSID 04371 Table

```

*
TTC1FB2 DS 0XL191
DC X'4041424344454647' (X'40' TO X'47')
DC X'4849714B4C4D4E5A' (X'48' TO X'4F')
DC X'5051525354555657' (X'50' TO X'57')
DC X'58595B685C5D5EB0' (X'58' TO X'5F')
DC X'6061626364656667' (X'60' TO X'67')
DC X'6869486B6C6D6E6F' (X'68' TO X'6F')
DC X'7071727374757677' (X'70' TO X'77')

```

DC X'78467AEF667D7E7F'	(X'78' TO X'7F')
DC X'8081828384858687'	(X'80' TO X'87')
DC X'88898A8B8C8D8E8F'	(X'88' TO X'8F')
DC X'9091929394959697'	(X'90' TO X'97')
DC X'98999A9B9C9D9E9F'	(X'98' TO X'9F')
DC X'A0A1A2A3A4A5A6A7'	(X'A0' TO X'A7')
DC X'A8A9AAABACADAFAF'	(X'A8' TO X'AF')
DC X'B0B1B2B3B4B5B6B7'	(X'B0' TO X'B7')
DC X'B8B9BABBBCBDBEBF'	(X'B8' TO X'BF')
DC X'CFC1C2C3C4C5C6C7'	(X'C0' TO X'C7')
DC X'C8C9CACBCCDCECF'	(X'C8' TO X'CF')
DC X'51D1D2D3D4D5D6D7'	(X'D0' TO X'D7')
DC X'D8D9DADBDCDDDEDF'	(X'D8' TO X'DF')
DC X'E0E1E2E3E4E5E6E7'	(X'E0' TO X'E7')
DC X'E8E9EAEBECEDEEEF'	(X'E8' TO X'EF')
DC X'F0F1F2F3F4F5F6F7'	(X'F0' TO X'F7')
DC X'F8F9FAFBFCFDFF'	(X'F8' TO X'FE')
END	

---

## Displaying square brackets used in C programs

The standard non-APL terminals that ISPF supports do not have the left and right brackets used in a C program. Therefore, the translation tables provided with ISPF are defined so that these characters are not valid and are displayed as periods.

If you have a terminal that supports these characters, you can modify the translation tables TTVAl and TTGSM. To do so, simply indicate that these characters are valid.

Note that the C/370 compiler expects the brackets at code points AD and BD. On an APL or TEXT terminal, if you use the ISPF-supplied terminal type 3278A, the code points AD and BD are displayed correctly as left and right brackets.

---

## ISPEXEC processing

ISPEXEC is an external entry point in module ISPLINK. This is how ISPF supports the Call ISPEXEC interface in module dialogs. As a result of executing a CLIST that is not under ISPF, if the CLIST contains ISPEXEC dialog service statements, CLIST might try to invoke the ISPEXEC module as a command processor. The results of this change are as follows:

- If the ISPLINK module (or its alias entry points ISPLNK, ISPEXEC, ISPEX, or ISPQRY) is not invoked under ISPF, TSO issues an error message with a return code of 20.
- The ISPEXEC entry point can interfere with your installation's setup if someone creates a CLIST called ISPEXEC, or if a CLIST specifically checks for a return code of 12 (TSO issues return code 12 if you try to invoke ISPEXEC when not under ISPF). To eliminate this problem, you make this change:
  - Move the ISPLINK load module (and alias entry points ISPLNK, ISPEXEC, ISPEX and ISPQRY) to a library that is not defined in the search sequence for attaching commands under ISPF.
- The ISPLINK load module is usually link-edited with dialog functions coded in a programming language. Therefore, you should copy ISPLINK to an "automatic call" link-edit library. Modify existing CLISTs to recognize the ISPEXEC return code of 20 for invocation outside of an ISPF environment (testing for a return code not equal to 0 is recommended). If you choose, you can turn off (NOP) the error message issued from the ISPLINK (and ISPEXEC) module by applying this SUPERZAP:

NAME	ISPLINK	ISPLINK	
VER	00FC	0A5D	TPUT SVC
REP	00FC	0700	NOP INSTRUCTION

Remember that the location of the 0A5D instruction can change from 00FC as a result of maintenance.

---

## ISPF-to-APL2 terminal type mappings

ISPF-to-APL2 terminal type mappings provides information about how to add or change mappings, if you need to do so. See *z/OS ISPF Dialog Developer's Guide and Reference* for a description of the ISPF/APL2 terminal type dialog, ISPAPTT.

ISPAPTT consists of an 8-character header (with value ISPAPTT), and fifteen 20-character entries. Each 20-character entry contains three fields. All fields are left-justified and padded on the right with blanks.

- The first 13 entries each appear as follows:
  - A 4-character EBCDIC sequence number; for example, "0001," "0002"
  - An 8-character EBCDIC ISPF terminal type
  - An 8-character EBCDIC APL2 terminal type
- The 14th entry consists of:
  - A 4-character EBCDIC sequence number; for example, "0014"
  - An 8-character EBCDIC value of "BATCH" noting the ISPF terminal type used if executing in the background (dialog ISPAPTT looks for this value)
  - An 8-character EBCDIC value of "1" noting the APL2<sup>®</sup> terminal type to be used if executing in the background
- The last (15th) entry contains:
  - A 4-character EBCDIC value equal to "LAST"
  - An 8-character value composed of all hexadecimal Fs, indicating the end of the list (dialog ISPAPTT looks for this value)
  - An 8-character EBCDIC value of "3277" indicating the APL2 terminal type to be used if the ISPF terminal type is not found in the list

You can change any of these values by performing a zap of the module. Include the APL2 terminal type to use in the background, and the APL2 terminal type to use if the ISPF terminal type is not found in the list.

**Note:** Do not alter the ISPF terminal type "BATCH" in the 14th entry or the 8-character hexadecimal Fs in the 15th entry.

Several of the first 13 entries can be changed to allow new terminal types. These entries contain a 4-character EBCDIC sequence number followed by 16 characters (two 8-character areas) of binary zeros. To change these entries use a zap and enter the ISPF terminal type into the first 8-character area, and the corresponding APL2 terminal type into the second 8-character area.

---

## Load APL2 workspace

If you run APL2\* with ISPF, each ISPF/APL user must load an APL2 workspace from the ISPALIB library. This assumes APL2 is correctly installed and the VSAM cluster needed to hold existing workspaces is defined. The ISPF/APL user only has to perform this step once.

The workspace to use is in the ISP.SISPALIB data set. To place the workspace into the VSAM cluster with your existing APL2 workspaces:

1. Enter ap12
2. Specify )IN 'ISP.SISPALIB(ISPFWS)' to bring in the workspace information
3. Specify )WSID *nnnnnnnn* to name the workspace
4. Specify )SAVE to save the workspace and put it in the VSAM cluster
5. Specify )OFF HOLD to leave APL2

## Tailoring ISPF defaults

In earlier versions of ISPF (before OS/390 V2R8.0), default values were set in the ISRCONFIG table and the ISPDFLTA and ISPMTAIL macros. These values are now set in the ISPF Configuration table. See Chapter 2, "The ISPF Configuration Table," on page 9 for more information.

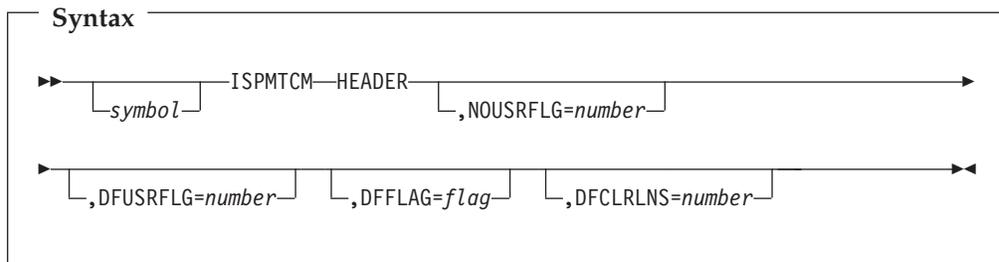
## Customizing the ISPF TSO command table (ISPTCM)

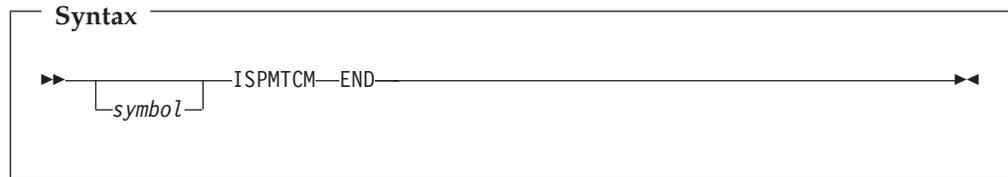
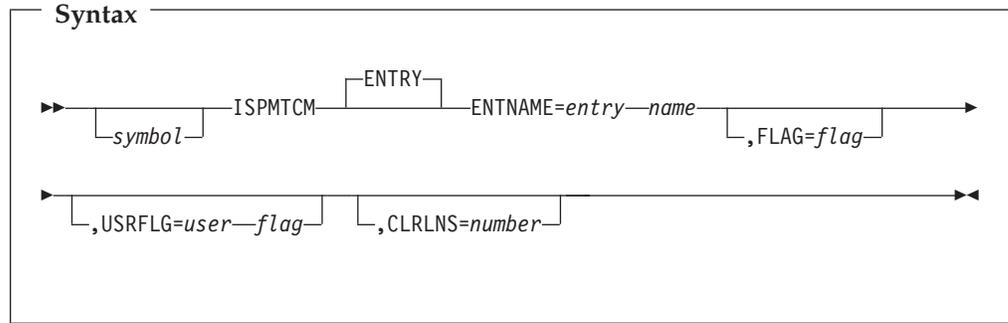
The ISPF TSO command table (ISPTCM) describes the TSO commands that are invoked under ISPF. When a TSO command is issued, ISPF searches ISPTCM. If the command is found, it uses the information in the table to process the command. If the command is not in ISPTCM, ISPF uses default values, which are in the table.

To change the list of TSO commands, their characteristics, or both, customize ISPTCM using the ISPMTCM macro. The assembler source of ISPTCM is member ISPTCMA in ISP.SISPSAMP. You must rename ISPTCMA to ISPTCM before assembling it. The macro ISPMTCM is in ISP.SISPMACS. ISP.SISPSAMP and ISP.SISPMACS are the default data set names for the ISPF data sets. Contact your system programmer for the names of the data sets containing ISPTCMA and ISPMTCM on your system. Customizing the ISPTCM involves modifying ISPTCMA, assembling it, and link-editing the ISPTCM module. When you use the ISPMTCM macro, remember that:

- If you modified ISPTCM in prior ISPF releases, you have to regenerate the table by using this macro.
- The first macro call must be HEADER, followed by one ENTRY macro call for each table entry desired, followed by a macro call of END.
- Use High Level Assembler to assemble ISPTCM.
- You can add up to four additional user-flag bytes for each entry in ISPTCM for your installation's use.
- The entry names in the IBM-supplied ISPTCM source, ISPTCMA, are arranged in alphabetical order. You must maintain alphabetical order for the entries within the ISPTCM.
- You can delete or modify any of the ISPF-provided entries in the table. ISPF allows you to have up to 1000 entries in ISPTCM.

The syntax of the macro is as follows:





<b>Operand</b>	<b>Description</b>
<b>HEADER</b>	<p>Parameter values are as follows:</p> <p><b>NOUSRFLG</b> The number of user flag bytes defined. The value can be a number between 0 and 4, inclusive. The default is 0.</p> <p><b>DFUSRFLG</b> The user flag that is to be used for an entry when the USRFLG operand is not specified. This must be a hexadecimal string, whose length in bytes must be NOUSRFLG. If you define a default user flag with this operand, the value for the NOUSRFLG operand must be greater than 0. If you do not define a default user flag and you indicated on the NOUSRFLG operand that a user flag exists, ISPF uses a string of binary zeros of whatever the length is in NOUSRFLG for the default user flag.</p> <p><b>DFFLAG</b> The value of the ISPF flag to be used for the default entry in ISPTCM. The default entry determines the characteristics of commands not found in ISPTCM. The value should be a 1-byte hexadecimal string. The default is 61.</p> <p><b>DFCLRLNS</b> The number of lines to clear from the bottom of the physical screen for line I/O when the CLRLNS operand is not specified. The value must be in the range from 0 to 99. The default is 3.</p>
<b>ENTRY</b>	<p>Parameter values are as follows:</p> <p><b>ENTNAME</b> A valid TSO command name. This operand is required for ENTRY calls. The alphabetic characters in ENTNAME must be in uppercase letters. Duplicate entry names cause an error message to be issued.</p>

## FLAG

The value of the ISPF flag byte for the current entry. The default is 02.

Flag Field	Flag Field Description
B'1.....'	Reserved.
B'.1.....'	Command requires function pool. Set this bit on for a command processor program that issues dialog services.
B'..1.....'	Command requires authorization check. Set this bit on for a command processor that must be invoked as an authorized command.
B'...1....'	Command is not to be logged. Set this bit on if the TSO command buffer should not be written to the ISPLOG data set.
B'....1...'	Command is not supported by ISPF. Set this bit on for commands that cannot be invoked under ISPF.
B'.....1..'	Command is command procedure (CLIST). Set this bit on if this is the name of a CLIST member.
B'.....1.'	Command is a command processor. Set this bit on if this is the name of a command processor program module.
B'.....1'	Command requires a BLDL to be issued. Set this bit on if a BLDL is to be issued to determine whether this is a command processor module or a CLIST.

## USRFLG

The value of the user flag for the current entry. Specify this parameter only if NOUSRFLG is not 0. If you do not specify a value and NOUSRFLG is greater than 0, ISPF uses the default user flag (DFUSRFLG). If specified, USRFLG must be a hexadecimal string with a length equal to the value of NOUSRFLG.

## CLRLNS

The minimum number of lines to clear if line mode is entered for this entry. The value should be an integer from 0 to 99. Specifying a value of 0 causes the entire physical screen to be erased. If you do not specify a value for CLRLNS, the DFCLRLNS value in the type HEADER call is used.

When the value for the number of lines to clear is nonzero, ISPF determines where to clear the screen according to:

1. ISPF calculates a value = (number of lines in the visible portion of the active logical screen - 1) - CLRLNS
2. The lesser of this calculated value and the number of lines in the panel displayed on the active logical screen (but not less than 0) is the number of the line after which the screen is cleared. Thus, ISPF will clear more than the

CLRLNS number of lines if it can do so without overlaying the displayed panel.

**Notes:**

1. If a CLRLNS value is larger than the visible portion of the active logical screen, ISPF erases the screen beginning at the top of the logical screen.
2. In split-screen mode, if line mode is entered when the top screen is active, the bottom screen will always be cleared. If the bottom screen is active, the bottom screen will be cleared (even if CLRLNS is greater than the physical screen size).
3. For DBCS devices, ISPF always erases the screen beginning at the top of the current logical screen, regardless of the CLRLNS value.
4. For 3290 devices, the entire physical screen is always cleared before going into line mode.

END                    Must be last macro call.

## Sample ISPTCM definition

An example of how to use the ISPMTCM macro to build ISPTCM is shown in Figure 54.

```
ISPMTCM  HEADER,DFCLRLNS=5,NOUSRFLG=2,DFFLAG=61
ISPMTCM  ENTRY,ENTNAME=ALLOCATE,CLRLNS=4
ISPMTCM  ENTNAME=MYCMDA,USRFLG=FFFF
ISPMTCM  ENTNAME=MYCMD1,CLRLNS=10,FLAG=14
ISPMTCM  END
```

Figure 54. Sample ISPTCM definition

## ISPTCM usage notes

- ISPTCM is a load module that contains a list of command names and their characteristics. ISPF processes each command in ISPTCM according to the FLAG field defined for its entry. If a command is not in ISPTCM, the DFFLAG parameter is used. The default value of DFFLAG is 61, which indicates to ISPF that commands not contained in ISPTCM require a function pool, an authorization check, and must be logged. Also, a BLDL should be done to locate the command. If the BLDL cannot locate the command, ISPF assumes it to be CLIST and attaches the EXEC command processor.  
You can alter DFFLAG to suit the needs of your installation. If you have not changed DFFLAG, or changed it so that it still calls for a BLDL, if a command processor is to run from the link pack area, its name *must* be in ISPTCM.
- Certain commands, such as, LOGON and ISPF are invalid under ISPF. Do not attempt to make these entries valid by changing the FLAG. The results are unpredictable.
- The USRFLG is an optional field for an entry in ISPTCM. If you define exit routines for TSO command start or TSO command end user exits, or both, you can define USRFLG according to your installation's needs. If you do, ISPF passes these flags to the exit routines. These flags do not affect ISPF execution. See Chapter 6, "ISPF installation-wide exits," on page 113 for a description of how the parameters are passed.

- One, and only one, of the last 3 bits of FLAG and DFFLAG must be 1. Otherwise, the results will be unpredictable.
- The presence or absence of a command in the TCM can affect the search sequence, as depicted in this table.

### Search sequence for attaching commands

ISPF attaches a command invoked through the SELECT command, ISPF option 6, or TSO command. The search sequence for locating the command is shown in Figure 55.

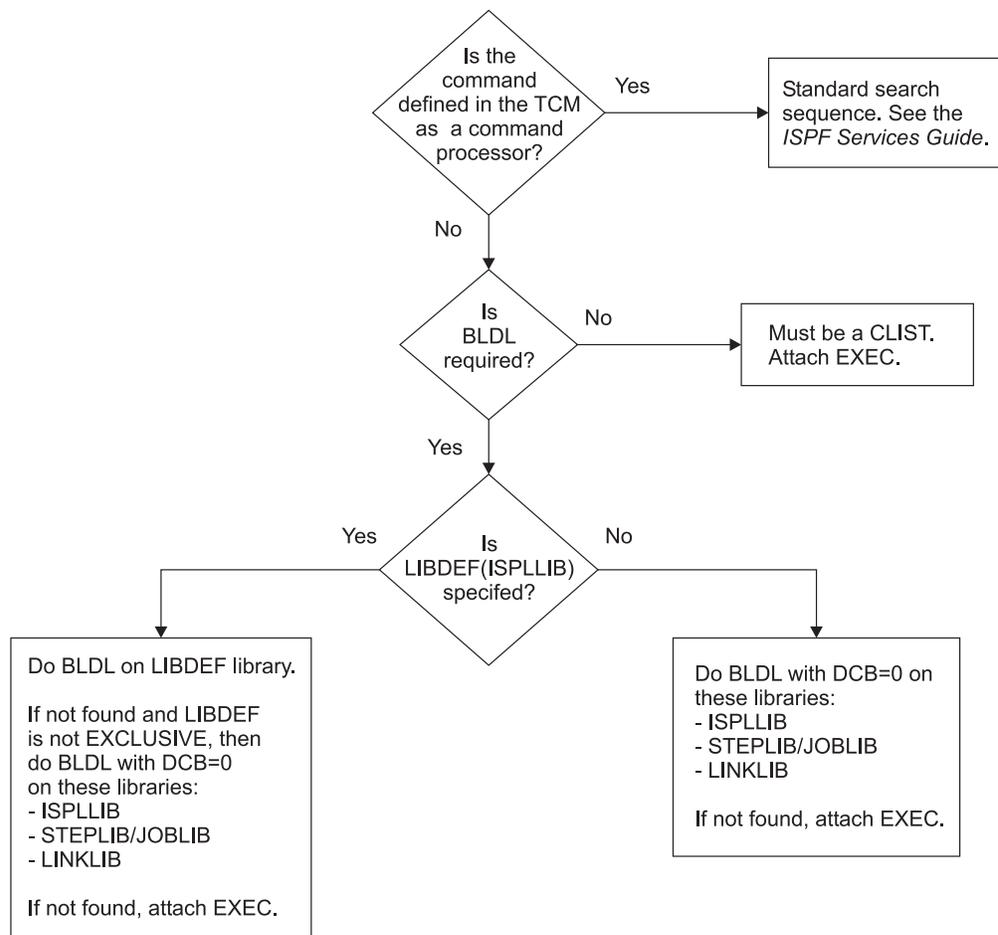


Figure 55. Search sequence for attaching commands

## Alternate option 7.1 panels

Panel ISPYFP, the normal panel displayed with option 7.1, requires extra scrolling to display some of the test entry fields and options. Because this can be inconvenient, ISPF provides alternate panels for ISPYFP. The alternate panels present all of the same entry fields and options in revised formats.

Panel ISPYFPA places most of the commonly used information within the first 24 panel lines. This format often eliminates the need for panel scrolling.

Panel ISPYFPB is similar to panel ISPYFPA, but it has a selection field that enables the user to select a function: panel, command, program, or request. Unlike panels ISPYFP or ISPYFPA, on panel ISPYFPB the panel, command, program, or request fields can all contain values.

All three panels are included in the product panel library. To select which panel to use, set the ISPF Configuration table keyword `USE_ALTERNATE_DIALOG_TEST_PANEL` to one of these options:

- 1 ISPYFP panel
- 2 ISPYFPA panel
- 3 ISPYFPB panel

---

## ISPF multicultural support

**Note:** The term "multicultural support" has replaced the previous term "National Language Support" (or "NLS").

To make ISPF available at your installation in a language other than or in addition to English, follow these instructions. You can install as many of the supported languages as needed by your installation.

1. Allocate and load libraries specific to the language. See the *z/OS Program Directory* for installation instructions.
2. Optionally, set up a default language for your installation.

With the standard version of ISPF, the default session language is English. You must invoke ISPF (ISPSTART) with a language keyword to get the session in a different language. If ISPF is used at your installation primarily in a non-English language, it is recommended that you change the default language. The benefits of doing this are:

- Not having to enter the language keyword when invoking ISPSTART
- Improved initialization time
- Possibly smaller LPA use
- Initialization error messages are issued in the default language.
- See "Changing the session language default value" for further details.

3. Set up the execution environment.

To run ISPF in any session language, perform these steps:

- Allocate panel, message, skeleton, table, and profile table libraries according to the language desired. The ddnames ISPPLIB, ISPMLIB, ISPSLIB, ISPTLIB do not change with the language used.
- Issue the ISPSTART command (with the desired language keyword if this language is different from the default language).

To run PDF in a language other than the default, issue the PDF command with the desired language keyword (for example, PDF GERMAN).

## Changing the session language default value

The Session Language Default value is a keyword, `DEFAULT_SESSION_LANGUAGE`, in the ISPF Configuration Table (see 234). For information about changing Configuration Table settings, see Chapter 2, "The ISPF Configuration Table," on page 9.



---

## Chapter 6. ISPF installation-wide exits

### Programming Interface information

ISPF provides 16 installation-wide exits that allow you to collect system-related information such as accounting, activity monitoring, authorization checking, and installation tailoring. These installation-wide exits occur at selected places during ISPF execution and pass control to your exit routines in the order you define them when you install the routines. You can define one or more exit routines to process at each user exit.

Table 25 summarizes the ISPF installation-wide exits. The exit ID is a unique number that identifies that installation-wide exit to ISPF.

*Table 25. ISPF installation-wide exits*

Exit ID	Installation Exit Name	Possible Uses for the Routine
1	ISPF initialization	Provides accounting and monitoring capabilities before ISPF initialization.
2	ISPF termination	Provides accounting and monitoring capabilities before ISPF termination.
3	SELECT service start	Provides monitoring information and lets you restrict access to applications selected through ISPF.
4	SELECT service end	Marks the end of a program, command, or menu invoked through any of the SELECT services.
5	TSO command start	Provides for monitoring and restricting commands invoked through ISPF; allows commands newly added to the system to be invoked without updating ISPTCM.
6	TSO command end	Provides for monitoring of TSO commands invoked through ISPF.
7	LIBDEF service	Provides for restrictions on the use of the LIBDEF service.
8	RESERVE	Allows use of your own method of serializing resources in addition to the RESERVE done by ISPF.
9	RELEASE	Provides for the release of any resources acquired at the RESERVE user exit.
10	Logical screen start	Allows for installation-wide exits to gather accounting and monitoring information for each logical screen.
11	Logical screen end	Gathers accounting and monitoring information for each logical screen.
12	ISPF/PDF service start	Monitors ISPF and PDF dialog services invoked through the ISPLINK or ISPEXEC interfaces.
13	ISPF/PDF service end	Marks the termination of ISPF or PDF dialog services invoked through the ISPLINK or ISPEXEC interfaces.
14	SWAP logical screens	Indicates a change of the active logical screen. Together with the logical screen start and end installation-wide exits, the routine can monitor resource use for each ISPF logical screen.
15	DISPLAY service start	Provides for tailoring of panels to be displayed.

Table 25. ISPF installation-wide exits (continued)

Exit ID	Installation Exit Name	Possible Uses for the Routine
16	Log, list, and temporary data set allocation	Controls data set naming conventions for log, list, and temporary data sets.

The ISPF installation-wide exits are linked together in the ISPEXITS load module. The main entry point of ISPEXITS is ISPXDT. ISPXDT defines which of the 16 installation-wide exits you plan to use and the names of the exit routines that receive control at each user exit. To assist you in building ISPXDT, ISPF provides seven assembler macros. These macros are described in “Exit macros” on page 249.

As you write exit routines for the ISPF installation-wide exits, remember that:

- Even though you can define multiple exit routines for a user exit, if an exit routine returns a return code of 12 or greater in register 15, control returns to ISPF without executing the remaining routines.
- You cannot activate or deactivate a user exit while an ISPF session is in progress. However, you can make changes to the ISPEXITS load module at any time, but remember that ISPF loads the module only once at session initialization. Therefore, the changes are not recognized until the next ISPF session.
- Because ISPF loads the ISPEXITS load module only once, consider making any exit routines that you write re-usable, and preferably reentrant. If you write exits that are *not* reentrant, they cannot be put in the Link Pack Area (LPA) library. Non-reentrant exit routines placed in the LPA can cause abend errors.
- You cannot invoke an ISPF service from within the exit routines.
- The installation-written exit routines receive control in an addressing mode of AMODE=31. The exit routines must support 31-bit addressing. ISPF does not restrict the residency mode of the installation-wide exits.
- Input-output exit routines can modify parameters. If there are multiple exit routines, successive routines receive these parameters as modified by the previous routine. The first exit routine receives parameters as described.
- You can define a data area to be used by each exit routine. These data areas can be shared by different exit routines.
- At initialization, ISPF gets the storage for the data area. It provides the length and the address of the data area in the parameter list when calling an exit routine that uses that data area. (ISPF obtains the storage from subpool 0 below the 16 MB line.) The first time ISPF calls the exit routine, the data area contains binary zeros. The information in the data area is retained between invocations of the exit routines and the storage is available for the entire ISPF session (independent of logical screen abends and restarts). It is released at ISPF termination.
- Data areas are on double-word boundaries.

---

## How to install the installation-wide exits

To install an ISPF installation-wide exit:

1. Indicate in installation tailoring that ISPF installation-wide exits are defined. When you install ISPF, you indicate that exit routines might exist by setting the ENABLE\_ISPF\_EXITS keyword in the ISPF Configuration table to YES. As a result, you can replace, add, or remove exit routines by simply reassembling ISPXDT and relinking ISPEXITS. Otherwise, ISPF uses the default value of NO and does not load the ISPEXITS module. In this case, you cannot invoke any

exit routines and you have to repeat part of the installation process if you want to change this value. See Chapter 2, “The ISPF Configuration Table,” on page 9 for more information.

2. Use the exit macros to assemble the exit definition table (ISPXDT). This process is discussed in detail in “Exit macros” on page 249. These macros reside in ISP.SISPMACS.
3. Link-edit the ISPEXITS load module. ISPXDT is the main entry point of this module. There is no need to include any other ISPF-supplied CSECTS in ISPEXITS. Figure 56 contains sample JCL to link-edit ISPEXITS.

```
//LKED EXEC PGM=IEWL,REGION=1024K,  
// PARM='XREF,LET,LIST,RENT,SIZE=(512K,128K)'  
//SYSPRINT DD SYSOUT=A  
//SYSUT1 DD UNIT=SYSDA,SPACE=(TRK,(10,5))  
//SYSLMOD DD DISP=SHR,DSN=ISP.SISPLoad  
/* All exit routines and ISPXDT CSECT must be in SYSLIB,  
/* but an INCLUDE SYSLIB statement is only required for  
/* ISPXDT and not for the exit routines.  
//SYSLIB DD DSN=ISP.LOCOBJ,DISP=SHR  
//SYSLIN DD *  
    ORDER ISPXDT  
    ENTRY ISPXDT  
    INCLUDE SYSLIB(ISPXDT)  
    NAME ISPEXITS(R)  
/*  
//
```

Figure 56. Sample JCL to Link-Edit ISPEXITS

4. ISPXDT does not contain any executable code. If your exit routines are reentrant, ISPEXITS can be copied into the LPA library. Otherwise, it must be copied into the system link library.

#### RELATED REFERENCES

“Exit macros” on page 249

---

## How to use the macros to define ISPXDT

To define ISPXDT using the exit macros:

1. Indicate the beginning of the exit entry definition section of ISPXDT by coding one ISPMXED macro with the START operand.

### Example

```
ISPMXED START  
  
:
```

2. Define the installation-wide exits you plan to use. Code the ISPMXLST macro, listing the numeric codes for each user exit in ascending order, enclosed within parentheses. This example shows that exit routines will be installed at ISPF installation-wide exits 3, 7, and 14.

**Example**

```
ISPMXED START
ISPMXLST (3,7,14)
:
:
```

If no operand is specified, a dummy ISPXDT is built.

3. For each user exit that you define on the ISPMXLST macro, use the ISPMXDEF, ISPMEPT, and ISPMXEND macros to define one or more exit routines. These three macros are a set and must be coded for each user exit you identify.
  - Code the ISPMXDEF macro with the exit ID as its operand.
  - Code the ISPMEPT macro for each exit routine and, if required, its data areas.
  - Code the ISPMXEND macro to end the exit routine definitions for this user exit.

In this example, three exit routines are defined at user exit 7. The first two share a common data area (DAREA015) and the third uses a different data area (DAREA027). ISPF calls entry points EXPT7EP1, EXPT7EP2, and EXPT7EP3 in that order unless one of the routines returns a return code of 12 or greater. For example, if entry point EXPT7EP2 returns a return code of 12, control returns to ISPF without executing entry point EXPT7EP3. Therefore, if you define more than one exit routine at a given user exit, define them in the correct order of priority.

**Example**

```
ISPMXED START
ISPMXLST (3,7,14)
:
:
ISPMXDEF 7
ISPMEPT EXPT7EP1,DAREA015
ISPMEPT EXPT7EP2,DAREA015
ISPMEPT EXPT7EP3,DAREA027
ISPMXEND
:
:
```

If you use the ISPMXDEF macro to define an exit ID that is not listed on the ISPMXLST macro, ISPF issues a warning message. You can still use ISPXDT, but the user exit is defined. As a result, you can easily disable a user exit by deleting its exit ID on the ISPMXLST macro. However, for every exit point listed on the ISPMXLST macro, there must be a corresponding ISPMXDEF macro.

4. Indicate the end of the exit entry definition section of ISPXDT by coding the END operand on the ISPMXED macro.

**Example**

```
ISPMXED START
ISPMXLST (3,7,14)
:
ISPMXED END
```

5. Define the data areas used by the exit routines in the exit data area definition section of ISPXDT. Indicate the start of this section by using the START operand on the ISPMXDD macro.

**Example**

```
:
ISPMXED END
ISPMXDD START
:
```

6. Use the ISPMDAD macro to define the name and size of each data area defined on an ISPMEPT macro. Every data area referenced on an ISPMEPT macro must be defined by an ISPMDAD macro.

This example defines DAREA015 to be 100 bytes long. (ISPF rounds the length to 104 in this case.)

**Example**

```
:
ISPMXDD START
ISPMDAD DAREA015,100
:
```

If you use the ISPMDAD macro to define a data area that is not coded on an ISPMEPT macro, a warning message is issued during assembly and storage for the area is not obtained during execution. If none of your exit routines require a data area, you need not code an ISPMDAD macro, but you must still code ISPMXDD START and ISPMXDD END.

7. Use the ISPMXDD macro to indicate the end of the exit data area definition section.

**Example**

```
:
ISPMXDD START
ISPMDAD DAREA015,100
:
ISPMXDD END
```

---

## Sample ISPXDT definition

Figure 57 shows a sample ISPXDT definition that defines exit routines for installation-wide exits 2 and 12.

```
<<ISPMXED<START
<*
<<ISPMXLST<(2,12)
<*
<<ISPMXDEF<2
<<ISPMEPT<MYEXT021,MYAREA01
<<ISPMEPT<MYEXT022,MYAREA02
<<ISPMEPT<MYEXT023,MYAREA02
<<ISPMXEND
<*
<<ISPMXDEF<12
<<ISPMEPT<MYEXT121,MYAREA03
<<ISPMEPT<MYEXT122,MYAREA01
<<ISPMEPT<MYEXT123
<<ISPMEPT<MYEXT124,MYAREA02
<<ISPMXEND
<*
<<ISPMXED<END
<*
<<ISPMXDD<START
<*
<<ISPMDAD<MYAREA01,1024
<<ISPMDAD<MYAREA02,2048
<<ISPMDAD<MYAREA03,256
<*
<<ISPMXDD<END
```

*Figure 57. Sample ISPXDT definition*

At user exit 2, exit routines with entry points MYEXT021, MYEXT022, and MYEXT023 are called, in that order.

At user exit 12, exit routines with entry points MYEXT121, MYEXT122, MYEXT123, and MYEXT124 are called, in that order.

MYAREA01 (1024 bytes) is used by exit routines at MYEXT021 and MYEXT122.

MYAREA02 (2048 bytes) is used by exit routines at MYEXT022, MYEXT023, and MYEXT124.

MYAREA03 (256 bytes) is used by the exit routine at MYEXT121.

Exit routine MYEXT123 does not require a data area.

---

## Exit parameter list

Each of the ISPF exit routines is passed a parameter list that contains parameters common to all the installation-wide exits and parameters that are specific to the user exit. The parameters are in the order shown here, followed by exit-specific parameters in the order shown under the individual user exit description. The common parameters are:

- |               |   |
|---------------|---|
| <b>exitid</b> | A fullword binary number that identifies the numeric code for the user exit.  |
| <b>userid</b> | An 8-character field that contains the TSO user ID (left-justified) of the current TSO session. If ISPF is running in batch and there is no |

	TSO user ID, the DSN prefix (as set by the TSO PROFILE PREFIX command) is placed in the Userid field. If there is no DSN prefix, the Userid field contains the characters 'BATCH '.
<b>screenid</b>	An 8-character field that identifies the active ISPF logical screen. Because ISPF supports up to 32 logical screens, the possible values are 0 through 9 and A through W.  At initialization and termination exit points, 0 indicates no active logical screens.  These identifiers are left-justified and the field is padded with blanks.
<b>ZENVIR</b>	32 characters of environmental information that is provided by the ZENVIR ISPF system variable. This variable is described in <i>z/OS ISPF Reference Summary</i> .
<b>datalen</b>	A fullword binary number that identifies the length of the exit data area, in bytes.
<b>dataptr</b>	The fullword address of the data area ISPF acquires for the exit routine. If you do not provide a data area name on the ISPMEPT macro, this address, as well as the data length, is 0. Note that the data area is always on a doubleword boundary and data length is a multiple of eight.

Standard OS linkage conventions are followed. For example: Register 1 (which points to a list of addresses; each address points to a different parameter) is used to pass the parameter list to the exit program. For more information about linkage conventions, refer to *z/OS MVS Programming: Assembler Services Guide*.

---

## Error processing

The installation-wide exit interface is a system programmer interface and, as such, no special error protection is provided for the exit routines. The ESTAE and ESTAI exit routines ISPF uses for abend recovery are used to recover from errors in the code for the user exit.

Abends within exit routines at these installation-wide exits cause ISPF to terminate:

- ISPF session initialization exit
- ISPF session termination exit
- Logical screen start exit
- Logical screen end exit
- SWAP exit

Abends within exit routines at any other user exit do not terminate ISPF, but do result in a logical screen restart, unless running in test mode or with ENBLDUMP on.

### RELATED REFERENCES

Chapter 10, "Exits," on page 249

|\_\_\_\_\_ End of Programming Interface information \_\_\_\_\_|



---

## Chapter 7. Customizing PDF

Customizing PDF describes procedures you can use to customize PDF.

Installation options modify the distributed release of PDF to suit your installation's particular needs. Most of the installation options are in the configuration table described in 9. You make some of these modifications by editing the panel descriptions in the ISPF panel library (ISPPPLIB data set).

All panels in the ISPPPLIB data set are in ISPF panel format. All messages in the ISPPMLIB data set are in ISPF message format. These formats are described in the *z/OS ISPF Dialog Developer's Guide and Reference*.

These topics describe how to modify the distributed release of PDF:

- Edit mode defaults
- Data set allocation defaults for the Outlist utility
- Using the Hardcopy utility with DBCS support
- SCLM Batch considerations
- Converting from LMF To SCLM
- Creating PDF translation tables
- PDF Foreground and Batch customizing
- Customizing Browse and Edit
- Customizing member list panels
- IBM Products option

See also Chapter 8, "PDF installation-wide exits," on page 181.

---

### Edit mode defaults

PDF saves several different edit modes in an edit profile. The user can specify the desired edit profile on the Edit Entry Panel. If the Profile field is left blank, the data set type is used as the profile name. For more information about edit profiles, refer to *z/OS ISPF Edit and Edit Macros*.

To preinitialize a set of edit profiles for first-time users, perform these steps:

1. Enter PDF.
2. Select the Edit option.
3. Set the edit profile with the defaults you chose.

For example, to set "COBOL FIXED 80" in your profile, edit a member of a partitioned data set that has a RECFM of F or FB, a LRECL of 80, and a type qualifier of COBOL (or enter COBOL as the profile name on the Edit Entry Panel).

ISPF provides two methods for initializing new edit profiles; you can set up a profile called ZDEFAULT in the ISPTLIB concatenation, or you can modify the edit profile defaults in the ISPF configuration table. The ISPF configuration table method is recommended because it is easier to maintain than the ZDEFAULT method. The ZDEFAULT method can still be used by individual users.

## Site-wide edit profile initialization

When no ZDEFAULT profile exists in the ISPTLIB concatenation and the user has no edit profile member in the ISPPROF concatenation, new edit profiles are created based on the settings in the ISPF configuration table. Using the configuration table, you can change any of the defaults for new edit profiles and you may override (force) settings for PACK, RECOVERY, WARN, SETUNDO, STATS, and IMACRO in existing profiles. When a setting is forced the editor WILL CHANGE the users' profiles, so be very careful if you override the IMACRO setting. IBM recommends that you use the site-wide initial macro instead of forcing the initial macro in each user's profile.

It is helpful to understand when the ZDEFAULT profile is used and where it exists in a user's concatenations. The ZDEFAULT profile exists as a row of the edit profile table named xxxEDIT where xxx is the application profile.

If ZDEFAULT exists in the edit profile table in the ISPTLIB concatenation, and the user has NO edit profile table in the ISPPROF allocation, the ZDEFAULT profile is copied from ISPTLIB into the user's edit profile when the user's edit profile is created. Therefore, many of your existing users may already have a ZDEFAULT profile in their edit profile. Individual users may delete their ZDEFAULT profiles using the PROFILE RESET command from within an edit session. Doing so will allow them to use the site-wide configuration for new profiles. You may also use a site-wide edit initial macro to issue a PROFILE RESET for all users. ISPF does not ship any edit profiles.

**Note:** If you use the force settings such as PACK OFF, edit macro commands which attempt to change forced settings will not get a failing return code, but the settings will not change.

## Creating a ZDEFAULT profile

Set up a special edit profile named ZDEFAULT (enter ZDEFAULT as the profile name on the Edit Entry Panel). The ZDEFAULT profile is the one used for the initial settings whenever a new edit profile is generated, regardless of the RECFM and LRECL values. For example, if you do not have an ASM profile and you edit an ASM data set, an ASM profile is generated using ZDEFAULT for the initial settings. If no ZDEFAULT profile exists, one is automatically generated with settings obtained from the ISPF Configuration Table.

The number of profiles you can establish also is described in the configuration table. See Chapter 2, "The ISPF Configuration Table," on page 9 for more details. When you finish, exit PDF. Your entire set of edit profiles is saved in your profile library (referenced by ddname ISPPROF) as the ISREDIT member.

The previous discussion assumes you are using Edit from the Edit Entry Panel, which is option 2 of the ISPF Primary Option Menu. If Edit is invoked from another dialog or if this dialog is altered, Edit might be invoked using a NEWAPPL value other than ISR. If this is the case, the table name begins with the NEWAPPL ID rather than ISR. Therefore, you must create a new set of defaults for the NEWAPPL ID. Copy this member to the table input library (referenced by ddname ISPTLIB). When a first-time user enters ISPF, there is no ISREDIT member in that user's profile library. As a result, Edit searches the table input library for member ISREDIT and uses it as the initial set of profiles for the new user. No ISREDIT member is distributed with the ISPF table input library.

## Action bars and extended color in Edit

Two ISPF configuration table keywords control whether action bars and highlighting are displayed in ISPF Edit. Disabling action bars provides more space to display data in the Edit panel. Disabling extended color support can improve system performance.

The keyword `ALLOW_EDIT_HIGHLIGHTING` can disable extended color support for all applications, including PDF itself and applications that use their own panels enabled for extended highlighting.

The `DEFAULT_EDIT_DISPLAY` keyword can be used to set the attributes of edit sessions invoked directly by PDF or by programs that invoke the edit service panel name other than `ISREDDE2`, `ISREDDE3`, `ISREDDE4`, or `ISREDDE5`. Using `DEFAULT_EDIT_DISPLAY`, you can make the edit session have:

- Neither action bars nor extended highlighting
- Extended highlighting with no action bars
- Action bars with no extended highlighting
- Both action bars and extended highlighting

`DEFAULT_EDIT_DISPLAY` can also be used to configure the editor to use the display method used in previous releases. This method does not support action bars or extended highlighting, but it performs much faster than the other methods. For more information, see “Disable Edit extended highlighting” on page 68.

If `ALLOW_EDIT_HIGHLIGHTING` is set to `NO`, the extended highlight support is disabled regardless of how `DEFAULT_EDIT_DISPLAY` is set.

See “Edit-related settings” on page 209 for more information about these keywords.

## Edit backup and recovery

Edit backup and recovery is controlled by two edit recovery tables. Table `ISREDRT` is used for PDF edit without the Edit Interface (EDIF) service. Table `ISREIRT` is used for PDF edit with the EDIF service. A copy of the tables is automatically saved in each user’s profile library. The number of entries (rows) in each table controls the number of recursion levels supported for backup and recovery. CLIST `ISREDRTI` or `ISREIRTI` for the EDIF service, builds the edit recovery table (`aaaaEDRT`, where `aaaa` is the ZAPPLID) the first time edit recovery is used for a given user and application ID.

The default table allows eight levels of recursion. After the table is built, you can execute CLIST `ISREDRTS` or `ISREIRTS` for the Edit Interface service to display it. To change the default size of the table, change the ‘SET N = ’ statement in CLIST `ISREDRTI` or `ISREIRTI` to any number from 1 to 99.

When a user enters Edit mode and recovery mode is on, or when a user attempts to turn on recovery mode, Edit automatically allocates a recovery data set if there is an unused entry in the edit recovery table. The edit recovery data set name is generated and placed in the table. The names generated are `ZPREFIX.ZUSER.AAAAxxx.BACKUP`, (where `AAAA` is the application ID and `xxx` is a 4-digit number between 0000 and 9999) if `ZPREFIX` and `ZUSER` are different, and `ZUSER.AAAAxxx.BACKUP` if `ZPREFIX` and `ZUSER` are the same.

The edit interface recovery table contains data set names with a last qualifier of `BACKUPI` instead of `BACKUP`. These data set names are passed to the data set

name change exit, if one exists, and the installation can change the names, if desired. Whenever the data set name change exit changes the recovery data set name, the PDF-generated recovery data set name is deleted and reused. The data set name change exit should check the recovery table and generate a unique data set name. The modified name is placed in the recovery table. If there is no unused entry, a message is displayed indicating recovery mode is not available. The user can continue editing with recovery mode off.

When you enter Edit, the edit profile controls the initial setting of recovery mode. When you terminate Edit, the system automatically deletes the recovery data set and frees the corresponding entry in the edit recovery table.

These restrictions apply to edit recovery data sets:

- they must be allocated as sequential data sets of record format U
- they cannot be striped, or striped and compressed data sets
- they cannot be multivolume data sets
- they should not be allocated on an SMS-managed volume whose management class has the PARTIAL\_RELEASE attribute

Edit recovery will not delete edit recovery data sets listed in the edit recovery table when the table has a disposition field set to 'K' associated with the recovery data set name. This is an obsolete facility for preallocating edit recovery files or enforcing naming conventions for edit recovery files. Instead, use the data set name change exit for this purpose.

---

## Data set allocation defaults for the Outlist utility

The Outlist utility (option 3.8) is available as either a CLIST (ISRUOL) or a program (ISRUOLP). The default used by ISPF is the program. To use the CLIST instead, perform these steps:

1. Modify panel ISRUTIL, changing:

```
      8, 'PGM(ISRUOLP)'  
to  
      8, 'CMD(ISRUOL)'
```

2. Modify the utilities action bar member (ISPDUTIL), changing:

```
<pdcc unavail=zut7 acc1=alt acc2=8>Outlist  
  <action run=ISRROUTE parm=U8>  
to  
  <pdcc unavail=zut7 acc1=alt acc2=8>Outlist  
  <action run=ISRUOL type=CMD>
```

3. Use Option 3.9 to create a command table entry called ISRUOL, specifying this as the action:

```
SELECT CMD(ISRUOL)
```

The Outlist utility invokes the TSO OUTPUT command to retrieve data from the SYSOUT queue. For the browse and print options of this utility, ISPF allocates a print data set and passes its name to the OUTPUT command by means of the PRINT parameter. The parameters in CLIST ISRUOL or the ISPF configuration table for program ISRUOLP determine the attributes of the print data set.

You can modify these print data set attributes to meet the needs of your installation. The default value for each attribute is shown in parentheses:

```
LRECL (133)  
BLKSIZE (3059)  
Primary space in tracks (200)  
Secondary space in tracks (100)
```

To change the attributes for CLIST ISRUOL, edit these lines as shown:

```
ATTR SPFUOL1 BLKSIZE(3059) LRECL(133)+
DSORG(PS) RECFM(FB &ZR)
ALLOC DA('&DSN') TRACKS /* */+
USING(SPFUOL1) RELEASE /* */+
SPACE(200 100) CATALOG /* */
```

To change the attributes for program ISRUOLP, modify the appropriate fields in the ISPF configuration table.

---

## Using the Hardcopy utility with DBCS support

When double-byte character set (DBCS) data and field-ruling information are in a data set, you cannot use the print program provided with Hardcopy utility (option 3.6). You must modify the ISRUHCP panel and the ISRUHCS1 skeleton to print the contents of the data set.

Use the ISRUHCP panel to specify print information to ISPF. Also, include the printout destination. Modify the panel as follows:

1. Add these input fields to the )BODY section:
  - A field to indicate a user-supplied print program is going to execute
  - Fields to provide information to the user-supplied program.
2. Add this logic to the )PROC section:
  - After all input checking is finished, if the user-supplied program is requested and a local terminal ID is specified, copy the terminal ID to a variable and clear out the original variable. Set a dummy SYSOUT class, if not set already.
  - VPUT all newly defined variables to the profile pool.

For information on modifying the )BODY and )PROC sections of a panel, see the *z/OS ISPF Dialog Developer's Guide and Reference*.

The ISRUHCS1 skeleton uses IEBGENER to print the data set. Make these changes to this skeleton:

1. If the user-supplied print program is requested, modify the SYSUT2 DD statement from SYSOUT to a temporary data set. Modify the DCB information also.
2. If the user-supplied program is requested, add a job step for initiating the user-supplied program that prints the temporary data set. You can get all information for the user-supplied program from the variables in the profile pool.

---

## SCLM Batch considerations

Before using the SCLM batch facility, modify the FLMLIBS skeleton to allow for batch submissions. FLMLIBS is found in the ISPF skeleton target data set ISP.SISPSLIB. It is the common imbed for the other SCLM skeletons used for batch submission. Modify the data set names in FLMLIBS to match your installation's naming conventions.

**Note:** In Figure 58 on page 126, the 'ISP' data set high-level qualifier represents your ISPF data sets. *xxx* corresponds to a national language as follows:

Language	xxx
US English	ENU (the default)
Uppercase English	ENP

Swiss German	DES
Japanese	JPN
German	DEU

```

)CM
)CM THIS DEFINES THE STEPLIB AND ISPF LIBRARIES
)CM TO BE USED DURING SCLM BATCH OPERATIONS
)CM
)CM BE SURE TO INCLUDE THE LOAD LIBRARIES CONTAINING ISPF.
//*
//*****
//* STEPLIB LIBRARIES
//*****
//*
//STEPLIB DD DSN=ISP.SISPLPA,DISP=SHR
//          DD DSN=ISP.SISPLOAD,DISP=SHR
//          DD DSN=CEE.SCEERUN,DISP=SHR
//          DD DSN=CEE.SCEERUN2,DISP=SHR
//*
//*****
//* ISPF LIBRARIES
//*****
//ISPLIB DD DSN=ISP.SISPMXXX,DISP=SHR ISPF MSGS
//
//ISPSLIB DD DSN=ISP.SISPSXXX,DISP=SHR ISPF SKELS
//          DD DSN=ISP.SISPSLIB,DISP=SHR ISPF SKELS
//
//ISPLLIB DD DSN=ISP.SISPPXXX,DISP=SHR ISPF PANELS
//
//ISPTLIB DD UNIT=&VIOUNIT;,DISP=(NEW,PASS),SPACE=(CYL,(1,1,5)),
//          DCB=(LRECL=80,BLKSIZE=19040,DSORG=PO,RECFM=FB),
//          DSN=&TABLESP          TEMPORARY TABLE LIBRARY
//          DD DSN=ISP.SISPTXXX,DISP=SHR ISPF TABLES
//
//ISPTABL DD UNIT=&VIOUNIT;,DISP=(NEW,PASS),SPACE=(CYL,(1,1,5)),
//          DCB=(LRECL=80,BLKSIZE=19040,DSORG=PO,RECFM=FB),
//          DSN=&TABLESP          TEMPORARY TABLE LIBRARY
//
//ISPPROF DD UNIT=&VIOUNIT;,DISP=(NEW,PASS),SPACE=(CYL,(1,1,5)),
//          DCB=(LRECL=80,BLKSIZE=19040,DSORG=PO,RECFM=FB),
//          DSN=&TABLESP          TEMPORARY TABLE LIBRARY
//
//ISPLLOG DD SYSOUT=*,
//          DCB=(LRECL=120,BLKSIZE=2400,DSORG=PS,RECFM=FB)
//
//ISPCTL1 DD DISP=NEW,UNIT=VIO,SPACE=(CYL,(1,1)),
//          DCB=(LRECL=80,BLKSIZE=800,RECFM=FB)  TEMPORARY FILE
//
//          TAILORING DATASET
//          OW01230
//SYSTEM DD SYSOUT=*
//
//*-----
//* TEMPORARY CLIST CONTAINING COMMAND TO BE EXECUTED
//*-----
//SYSPROC DD DSN=&&&CLIST&STEP,DISP=(OLD,DELETE)
//          DD DSN=ISP.SISPCLIB,DISP=SHR          CLIST LIBRARY OW01230
//
)CM
)CM 5647-A01 (C) COPYRIGHT IBM CORP 1989, 2005 */

```

Figure 58. Sample FLMLIBS skeleton

---

## Converting from LMF To SCLM

The LMF function was removed from ISPF in z/OS V1R2.0. Users of LMF are encouraged to migrate their existing LMF-controlled libraries to SCLM. There are several differences between LMF and SCLM, but the basic library manager function is similar, so migrating the hierarchy from LMF to SCLM is straightforward. SCLM has a full set of services, so most dialogs that use LMF services can be converted easily to SCLM. SCLM also includes a complete set of Configuration Management options.

### General conversion

Before doing the conversion, you should review the *z/OS ISPF Software Configuration and Library Manager Guide and Reference*, particularly these chapters:

- “Defining the Project Environment”, which describes the steps for setting up a project definition.
- “Converting Projects to SCLM”, which shows the steps for working with existing project data sets.

SCLM and LMF are similar in function, but there are some basic differences to consider when converting a project:

- In SCLM, all data sets in the hierarchy are controlled. This includes the data sets where the developers do their editing (called the *development level* in SCLM). These data sets are not controlled in the LMF environment.
- SCLM does not use a started task to update the controlled libraries. Users must have UPDATE authority to any level of the hierarchy that they must change, including the development level for editing, and any level into which they need to promote.
- SCLM uses a VSAM data set to hold its control information. Control information includes information about where members exist in the hierarchy, statistics, and dependency information.
- SCLM automates the BUILD step (compile, assemble, link-edit, and so on) and requires it to be done. This means SCLM must know what language is assigned to a part, and how the language is to be processed at BUILD time (called “SCLM Language Definition”).
- Output data, such as compiled object and load modules, can be automatically generated by SCLM. Review your processes to determine how to take advantage of SCLM’s synchronization of source modules and assigned outputs.

To convert from LMF to SCLM, follow the steps listed in “Defining the Project Environment” in *z/OS ISPF Software Configuration and Library Manager Guide and Reference*. This takes you through the creation of the SCLM project definition and control data sets needed for SCLM.

The steps involved in defining the project environment are:

1. Determine the project hierarchy. For an existing project the hierarchy has already been defined, but a review is recommended.
2. Identify the types of data to support. Again, this has already been done with an existing project.
3. Establish authorization codes. Authorization codes control the movement of data within the hierarchy. The purpose of this step is to assign authorization codes to the hierarchy. Authorization codes restrict the drawdown and promotion of members to certain groups within the hierarchy. The concept is

similar to LMF's concepts of authorized promoters and member access control, except that LMF uses userids and SCLM uses group names to restrict drawdown and promotion.

4. Allocate the PROJDEFS data sets. The PROJDEFS data sets are used to store the project definition data for an individual project. The project definition contains some of the same types of information as the LMF control file, such as the structure of the hierarchy.
5. Allocate the project partitioned data sets. Because you are using an existing project, most of these data sets already exist. Since all data sets are controlled in the SCLM environment, you will have to create development level data sets as part of the hierarchy for the developers to use when editing data.
6. Allocate and create the control data sets. Control data sets are used to track and control application programs within the hierarchy. There are 5 types of VSAM data sets that can be associated with a project:

#### **Primary Accounting**

The accounting data sets contain information about the software components in the project including statistics, dependency information and build maps (information about the last build of a member). At least one accounting data set is required for a project. The accounting data set contains some of the same types of information as the LMF control file, such as the member locking data.

#### **Secondary Accounting**

This is an optional backup of the information in the primary accounting data set.

#### **Export Accounting**

The export accounting data set contains accounting information that has been exported from the accounting data set. It is only required if the Import/Export functions of SCLM are used.

#### **Primary Audit Control**

The audit control data set contains audit information about changes to the software components in the project for groups that have auditing turned on. The audit data set contains the same types of information as the LMF activity log, as well as versions of members, if requested.

#### **Secondary Audit Control**

This is an optional backup of the information in the primary Audit Control data set.

7. Protect the project environment. This step should be reviewed because SCLM is structured slightly differently than LMF. Remember that developers must have UPDATE authority to the levels they need to change.
8. Create the project definition. The project definition is an assembler program, created using SCLM macros. Much of the definition is similar to the information that was entered in LMF's Library Controls (Option 8.1).
9. Assemble and link the project definition.

After the project has been defined, follow the steps in the *z/OS ISPF Software Configuration and Library Manager Guide and Reference* to put your existing project under SCLM control. In general, the steps are:

- Prepare the hierarchy. Delete all unnecessary data from the libraries. If any groups are to be non-key, make sure they don't contain any data.
- Create alternate project definitions.
- Create architecture definitions for the project.

- Register existing members with SCLM.

## Converting LMF dialogs to SCLM

There were eight ISPF services relating to LMF. Some of these have SCLM equivalents, some do not because of the differences between SCLM and LMF. The eight services are:

### **LMPROM**

Use the SCLM PROMOTE service.

### **EDIT with the LOCK parameter**

Use the SCLM EDIT service.

### **LMMFIND with the LOCK parameter.**

Use the SCLM LOCK service if locking is required. Use the DBUTIL service to find members in an SCLM project. ISPF services which use the LOCK keyword can also be modified to remove the keyword.

Alternatively, you can use the ISPF Configuration option FAIL\_ON\_LMF\_LOCK. If this option is set to NO, dialogs will continue to operate correctly without code modifications.

### **LMCOPY with the LOCK parameter**

Use the SCLM LOCK service, then use LMCOPY to copy the member.

### **LMHIER**

Use the SCLM NEXTGRP service. The NEXTGRP service returns the target group for a specified group. To simulate the data returned by LMHIER, loop on the NEXTGRP service until a return code greater than zero is returned. See SCLM example FLMNXTGP in ISPSISPSAMP for an example.

### **LMREVIEW**

There is no equivalent function in SCLM because the project control information is much different. Information about the contents of the hierarchy can be obtained using the SCLM DBUTIL service.

### **LMACT and LMDEACT**

SCLM does not follow the concept of active and inactive projects. Updates are done by updating the SCLM project definition, then reassembling and relinking to create an updated load module. This load module is accessed when you enter SCLM (or exit and reenter, if SCLM is active when the new module is built).

---

## Creating PDF translation tables

Creating PDF translation tables describes how to create a set of PDF translation tables.

Sample assembler modules are included in the ISPSISPSAMP sample library (members ISROWNTT and ISRAPLTT). Module ISROWNTT contains a complete set of translation tables for an English 3278/3279 terminal, and module ISRAPLTT contains a complete set of translation tables for the 3278/3276 APL terminals. Use these as an example of what a completed module should look like. You can modify the samples to suit your requirements, supplying the desired values for each of the translation tables.

PDF uses these translation tables:

- Valid data set name character translation table
- Invalid data set name character translation table

- Hexadecimal character translation table
- Numeric character translation table
- Alphanumeric character translation table
- Edit terminal output character translation table
- Generic string character translation table
- Generic string special character translation table
- Uppercase character translation table
- Lowercase character translation table.

The sample Assembler module includes all of these translation tables. Each translation table consists of 32 consecutive DC statements, where each DC statement consists of eight hexadecimal values. You are free to supply the desired 256 hexadecimal values that comprise each of the translation tables. The only exception to this is the generic string character translation table that consists of 32 consecutive DC statements, each consisting of one hexadecimal value. This table should not be modified. The address of each table is located at the start of the Assembler module. If a table is not used, the address for that table is set to 0 (for example: EDIP DC A(0)).

#### RELATED REFERENCES

Chapter 11, “PDF translation tables,” on page 265

---

## PDF Foreground and Batch customizing

With PDF you can customize the Foreground and Batch processing options. You can add and modify existing Foreground or Batch options, develop new primary options to provide an interface to the Foreground or Batch processing mechanisms, and modify the tutorial to reflect installation-developed modifications.

**Attention:** Do not try to customize PDF Foreground and Batch options unless you are a system programmer who is thoroughly familiar with ISPF.

Review the *z/OS ISPF Dialog Developer's Guide and Reference* for a description of panel and message definition formats, and also for specific requirements for selection panels (menus) and tutorial panels. The distributed panels provide examples of ISPF selection and tutorial panels. The ISPF Primary Option Menu is named ISR@PRIM and the first tutorial panel is named ISR00000.

The PDF Foreground and Batch options use selection and data entry panels in combination with CLISTs and file skeletons (Batch only). The specific requirements for foreground and batch panels, CLISTs, and skeletons are described in these sections.

### Foreground processing panels and CLISTs

The Foreground processing option uses ISPF dialog management services. Figure 59 on page 131 shows the Foreground selection panel definition. See the *z/OS ISPF Dialog Developer's Guide and Reference* for a general description of panel definition formats.

**Note:** In Figure 59, attribute characters have been replaced by blanks. Also, some of the ISPF-supplied CLISTs contain a specific library (such as SYS1.LINKLIB) on the program call. If the called program does not reside in that library, you might need to customize the CLIST.

```

)PANEL KEYLIST(ISRSAB,ISR)
)ATTR DEFAULT(...) FORMAT(MIX)                /* ISRFPA - ENGLISH - 5.6 */
0B TYPE(AB)
0D TYPE(PS)
04 TYPE(ABSL) GE(ON)
05 TYPE(PT)
09 TYPE(FP)
0A TYPE(NT)
11 TYPE(SAC)
22 TYPE(WASL) SKIP(ON) GE(ON)
10 TYPE(ET)
26 AREA(SCRL) EXTEND(ON)
27 TYPE(CEF) PADC(USER) CKBOX(ON)
28 TYPE(NEF) CAPS(ON) PADC(USER)
)ABC DESC('Menu') MNEM(1)
PDC DESC('Settings') UNAVAIL(ZPM1) MNEM(1) ACC(CTRL+S)
ACTION RUN(ISRRROUTE) PARM('SET')
PDC DESC('View') UNAVAIL(ZPM2) MNEM(1) ACC(CTRL+V)
ACTION RUN(ISRRROUTE) PARM('BR1')
PDC DESC('Edit') UNAVAIL(ZPM3) MNEM(1) ACC(CTRL+E)
ACTION RUN(ISRRROUTE) PARM('ED1')
PDC DESC('ISPF Command Shell') UNAVAIL(ZPM4) MNEM(6) ACC(CTRL+C)
ACTION RUN(ISRRROUTE) PARM('C1')
PDC DESC('Dialog Test...') UNAVAIL(ZPM5) MNEM(8) ACC(CTRL+T)
ACTION RUN(ISRRROUTE) PARM('DAL')
PDC DESC('Other IBM Products...') UNAVAIL(ZPM6) MNEM(1) ACC(CTRL+O)
ACTION RUN(ISRRROUTE) PARM('OIB')
PDC DESC('SCLM') UNAVAIL(ZPM7) MNEM(3) ACC(CTRL+L)
ACTION RUN(ISRRROUTE) PARM('SCL')
PDC DESC('ISPF Workplace') UNAVAIL(ZPM8) MNEM(6) ACC(CTRL+W)
ACTION RUN(ISRRROUTE) PARM('WRK')
PDC DESC('Status Area...') UNAVAIL(ZPMS) MNEM(8) ACC(CTRL+A)
ACTION RUN(ISRRROUTE) PARM('SAM')
PDC DESC('Exit') MNEM(2) PDSEP(ON) ACC(CTRL+X) ACTION RUN(EXIT)
)ABCINIT
.ZVARS=ISR@OPT
)ABC DESC('Utilities') MNEM(1)
PDC DESC('Library') UNAVAIL(ZUT1) MNEM(1) ACC(ALT+1)
ACTION RUN(ISRRROUTE) PARM('U1')
PDC DESC('Data set') UNAVAIL(ZUT2) MNEM(1) ACC(ALT+2)
ACTION RUN(ISRRROUTE) PARM('U2')
PDC DESC('Move/Copy') UNAVAIL(ZUT3) MNEM(1) ACC(ALT+3)
ACTION RUN(ISRRROUTE) PARM('U3')
PDC DESC('Data Set List') UNAVAIL(ZUT4) MNEM(2) ACC(ALT+4)
ACTION RUN(ISRRROUTE) PARM('U4')
PDC DESC('Reset Statistics') UNAVAIL(ZUT5) MNEM(5) ACC(ALT+5)
ACTION RUN(ISRRROUTE) PARM('U5')
PDC DESC('Hardcopy') UNAVAIL(ZUT6) MNEM(1) ACC(ALT+6)
ACTION RUN(ISRRROUTE) PARM('U6')
PDC DESC('Download...') UNAVAIL(ZUTDT) MNEM(2) ACC(ALT+7)
ACTION RUN(ISRRROUTE) PARM('UDT')
PDC DESC('Outlist') UNAVAIL(ZUT7) MNEM(2) ACC(ALT+8)
ACTION RUN(ISRRROUTE) PARM('U8')
PDC DESC('Commands...') UNAVAIL(ZUT8) MNEM(1) ACC(ALT+9)
ACTION RUN(ISRRROUTE) PARM('U9')
PDC DESC('Reserved') UNAVAIL(ZUT9) MNEM(6) ACTION RUN(ISRRROUTE) PARM('U10')
PDC DESC('Format') UNAVAIL(ZUT10) MNEM(1) ACC(ALT+F1)
ACTION RUN(ISRRROUTE) PARM('U11')
PDC DESC('SuperC') UNAVAIL(ZUT11) MNEM(1) PDSEP(ON) ACC(CTRL+F2)
ACTION RUN(ISRRROUTE) PARM('U12')

```

Figure 59. Foreground selection panel definition (ISRFPA) (Part 1 of 3)



```

)REINIT
REFRESH(ZFPKEDV)
)PROC
  &DSN      = ' '          /* INITIALIZE DATA SET NAME FIELD      */
  &ZORG      = ' '          /* INITIALIZE DATA SET ORGANIZATION VARIABLE */
  IF (&ZFPKEDV = ' ')
    &ZFPKED = 'NO'
  ELSE
    &ZFPKED = 'YES'
  &ZFPKED = TRUNC(&ZFPKED,1)
  VER (&ZFPKED,NB,LIST,Y,N) /* Y = EXPAND PACKED DATA */
  &ZFPKED = TRANS(TRUNC(&ZFPKED,1),Y,YES,N,NO)
  &ZFPAK = TRANS(TRUNC(&ZFPKED,1),Y,YES,N,NO)
  VPUT (ZFPAK,ZFPKED) PROFILE
&ZCMDWRK = &Z
IF (&ZCMD ^= &Z)
  &ZCMDWRK = TRUNC(&ZCMD, '.')
  &ZTRAIL=.TRAIL
  IF (&ZCMDWRK = &Z)
    .MSG = ISRU000
&ZSEL = TRANS (TRUNC (&ZCMD, '.'))
1, 'PGM(ISRFPR) PARM((ISRF01) 1) NEWPOOL'
2, 'PGM(ISRFPR) PARM((ISRF02) 2) NEWPOOL'
3, 'PGM(ISRFPR) PARM((ISRF03) 3) NEWPOOL'
5, 'PGM(ISRFPR) PARM((ISRF05) 5) NEWPOOL'
6, 'PGM(ISRFPR) PARM((ISRF06) 6) NEWPOOL'
7, 'PGM(ISRFPR) PARM((ISRF07) 7) NEWPOOL'
9, 'PGM(ISRFPR) PARM((ISRF09) 9) NEWPOOL'
10, 'PGM(ISRFPR) PARM((IGZTPIN2,ISRF10) 10) NEWPOOL'
10A, 'PGM(ISRFPR) PARM((ISRF10A) 10A) NEWPOOL'
11, 'PGM(ISRFPR) PARM((AFFFP11,DBBFP11,ISRF11) 11) NEWPOOL'
12, 'PGM(ISRFPR) PARM((ISRF12) 12) NEWPOOL'
13, 'PGM(ISRFPR) PARM((EDCFP13,ISRF13) 13) NEWPOOL'
14, 'PGM(ISRFPR) PARM((FANFP14,ISRF14) 14) NEWPOOL'
15, 'PGM(ISRALTDI) PARM(EVGFP15,,ISRF15,*) NOCHECK'
16, 'PGM(ISRFPR) PARM((EDCFP16,ISRF16) 16) NEWPOOL'
18, 'CMD(ISPD TLC (PANEL RETURN)) MODE(FSCR)'
19, 'PGM(ISRALTDI) PARM(CBC3PE4A,+,ISRF19,*,ISRF19A) NEWPOOL'
' ' ' '
*, '?'')
)PNTS
FIELD(ZPS01001) VAR(ZCMD) VAL(1)
FIELD(ZPS01002) VAR(ZCMD) VAL(11)
FIELD(ZPS01003) VAR(ZCMD) VAL(2)
FIELD(ZPS01004) VAR(ZCMD) VAL(12)
FIELD(ZPS01005) VAR(ZCMD) VAL(3)
FIELD(ZPS01006) VAR(ZCMD) VAL(13)
FIELD(ZPS01007) VAR(ZCMD) VAL(5)
FIELD(ZPS01008) VAR(ZCMD) VAL(14)
FIELD(ZPS01009) VAR(ZCMD) VAL(6)
FIELD(ZPS01010) VAR(ZCMD) VAL(15)
FIELD(ZPS01011) VAR(ZCMD) VAL(7)
FIELD(ZPS01012) VAR(ZCMD) VAL(16)
FIELD(ZPS01013) VAR(ZCMD) VAL(9)
FIELD(ZPS01014) VAR(ZCMD) VAL(18)
FIELD(ZPS01015) VAR(ZCMD) VAL(10)
FIELD(ZPS01016) VAR(ZCMD) VAL(19)
FIELD(ZPS01017) VAR(ZCMD) VAL(10A)
)END

/* 5694-A01 (C) COPYRIGHT IBM CORP 1980, 2004 */
/* ISPD TLC Release: 5.6. Level: PID */
/* z/OS 01.06.00. Created - Date: 19 Nov 2003, Time: 12:37 */

```

Figure 59. Foreground selection panel definition (ISRFPA) (Part 3 of 3)

Each Foreground option has an associated panel and CLIST. For example, option 1 has the option panel name ISRFP01 with the corresponding CLIST name ISRFC01. Figure 60 shows the ISRFP01 option panel definition.

**Note:** In Figure 60, attribute characters have been replaced by blanks.

```

)PANEL KEYLIST(ISRSAB,ISR)
)ATTR DEFAULT(...) FORMAT(MIX)          /* ISRFP01 - ENGLISH - 5.2 */
0B TYPE(AB)
04 TYPE(ABSL) GE(ON)
05 TYPE(PT)
09 TYPE(FP)
0A TYPE(NT)
0C TYPE(NT) SKIP(ON)
11 TYPE(SAC)
12 TYPE(CEF) PADC(USER)
19 TYPE(DT)
22 TYPE(WASL) SKIP(ON) GE(ON)
08 TYPE(CH)
26 TYPE(NEF) CAPS(ON) PADC(USER)
27 AREA(SCRL) EXTEND(ON)
28 TYPE(SAC) CSRGRP(99) RADIO(ON)
)ABC DESC('Menu') MNEM(1)
PDC DESC('Settings') UNAVAIL(ZPM1) MNEM(1) ACC(CTRL+S)
ACTION RUN(ISRROUTE) PARM('SET')
PDC DESC('View') UNAVAIL(ZPM2) MNEM(1) ACC(CTRL+V)
ACTION RUN(ISRROUTE) PARM('BR1')
PDC DESC('Edit') UNAVAIL(ZPM3) MNEM(1) ACC(CTRL+E)
ACTION RUN(ISRROUTE) PARM('ED1')
PDC DESC('ISPF Command Shell') UNAVAIL(ZPM4) MNEM(6) ACC(CTRL+C)
ACTION RUN(ISRROUTE) PARM('C1')
PDC DESC('Dialog Test...') UNAVAIL(ZPM5) MNEM(8) ACC(CTRL+T)
ACTION RUN(ISRROUTE) PARM('DAL')
PDC DESC('Other IBM Products...') UNAVAIL(ZPM6) MNEM(1) ACC(CTRL+O)
ACTION RUN(ISRROUTE) PARM('OIB')
PDC DESC('SCLM') UNAVAIL(ZPM7) MNEM(3) ACC(CTRL+L)
ACTION RUN(ISRROUTE) PARM('SCL')
PDC DESC('ISPF Workplace') UNAVAIL(ZPM8) MNEM(6) ACC(CTRL+W)
ACTION RUN(ISRROUTE) PARM('WRK')
PDC DESC('Status Area...') UNAVAIL(ZPMS) MNEM(8) ACC(CTRL+A)
ACTION RUN(ISRROUTE) PARM('SAM')
PDC DESC('Exit') MNEM(2) PDSEP(ON) ACC(CTRL+X) ACTION RUN(EXIT)
)ABCINIT
.ZVARS=ISR@OPT
)ABC DESC('RefList') MNEM(1)
PDC DESC('Current Data Set List &ZDSCURT') MNEM(1) ACC(CTRL+ALT+P)
ACTION RUN(ISRRLIST) PARM('PL1')
PDC DESC('Current Library List &ZDSCURLT') MNEM(2) ACC(CTRL+SHIFT+P)
ACTION RUN(ISRRLIST) PARM('LL1')
PDC DESC('List of Personal Data Set Lists') MNEM(1) PDSEP(ON) ACC(CTRL+ALT+O)
ACTION RUN(ISRRLIST) PARM('PL2')
PDC DESC('List of Personal Library Lists') MNEM(2) ACC(CTRL+SHIFT+O)
ACTION RUN(ISRRLIST) PARM('LL2')

```

Figure 60. Foreground Assembler H definition (ISRFP01) (Part 1 of 5)

```

)ABCINIT
.ZVARS=REFLIST
    VGET (ZCURTB ZCURLTB) PROFILE
    IF (&ZCURTB = &Z) &ZDSCURT = &Z
    ELSE &ZDSCURT= '(&ZCURTB)'
    IF (&ZCURLTB = &Z) &ZDSCURLT = &Z
    ELSE &ZDSCURLT= '(&ZCURLTB)'
)ABC DESC('Utilities') MNEM(1)
PDC DESC('Library') UNAVAIL(ZUT1) MNEM(1) ACC(ALT+1)
    ACTION RUN(ISRRROUTE) PARM('U1')
PDC DESC('Data set') UNAVAIL(ZUT2) MNEM(1) ACC(ALT+2)
    ACTION RUN(ISRRROUTE) PARM('U2')
PDC DESC('Move/Copy') UNAVAIL(ZUT3) MNEM(1) ACC(ALT+3)
    ACTION RUN(ISRRROUTE) PARM('U3')
PDC DESC('Data Set List') UNAVAIL(ZUT4) MNEM(2) ACC(ALT+4)
    ACTION RUN(ISRRROUTE) PARM('U4')
PDC DESC('Reset Statistics') UNAVAIL(ZUT5) MNEM(5) ACC(ALT+5)
    ACTION RUN(ISRRROUTE) PARM('U5')
PDC DESC('Hardcopy') UNAVAIL(ZUT6) MNEM(1) ACC(ALT+6)
    ACTION RUN(ISRRROUTE) PARM('U6')
PDC DESC('Download...') UNAVAIL(ZUTDT) MNEM(2) ACC(ALT+7)
    ACTION RUN(ISRRROUTE) PARM('UDT')
PDC DESC('Outlist') UNAVAIL(ZUT7) MNEM(2) ACC(ALT+8)
    ACTION RUN(ISRRROUTE) PARM('U8')
PDC DESC('Commands...') UNAVAIL(ZUT8) MNEM(1) ACC(ALT+9)
    ACTION RUN(ISRRROUTE) PARM('U9')
PDC DESC('Reserved') UNAVAIL(ZUT9) MNEM(6) ACTION RUN(ISRRROUTE) PARM('U10')
PDC DESC('Format') UNAVAIL(ZUT10) MNEM(1) ACC(ALT+F1)
    ACTION RUN(ISRRROUTE) PARM('U11')
PDC DESC('SuperC') UNAVAIL(ZUT11) MNEM(1) PDSEP(ON) ACC(CTRL+F2)
    ACTION RUN(ISRRROUTE) PARM('U12')
PDC DESC('SuperCE') UNAVAIL(ZUT12) MNEM(3) ACC(CTRL+F3)
    ACTION RUN(ISRRROUTE) PARM('U13')
PDC DESC('Search-For') UNAVAIL(ZUT13) MNEM(2) ACC(CTRL+F4)
    ACTION RUN(ISRRROUTE) PARM('U14')
PDC DESC('Search-ForE') UNAVAIL(ZUT14) MNEM(4) ACC(CTRL+F5)
    ACTION RUN(ISRRROUTE) PARM('U15')
)ABCINIT
.ZVARS=PDFUTIL
    &zut9 = '1'
)ABC DESC('Help') MNEM(1)
PDC DESC('Input Data Set') MNEM(1) ACTION RUN(TUTOR) PARM('ISR41001')
PDC DESC('Macro and copy libraries') MNEM(1) ACTION RUN(TUTOR) PARM('ISR41002')
PDC DESC('Object Data Set') MNEM(1) ACTION RUN(TUTOR) PARM('ISR41003')
PDC DESC('Listing Data Set') MNEM(1) ACTION RUN(TUTOR) PARM('ISR41004')
PDC DESC('Password protection') MNEM(1) ACTION RUN(TUTOR) PARM('ISR41005')
PDC DESC('Assembler selection') MNEM(1) ACTION RUN(TUTOR) PARM('ISR41006')
PDC DESC('Example') MNEM(1) ACTION RUN(TUTOR) PARM('ISR41007')
PDC DESC('Appendices') MNEM(5) ACTION RUN(TUTOR) PARM('ISR00004')
PDC DESC('Index') MNEM(3) ACTION RUN(TUTOR) PARM('ISR91000')
)ABCINIT
.ZVARS=FP1HELP

```

Figure 60. Foreground Assembler H definition (ISRFP01) (Part 2 of 5)

```

)BODY CMD(ZCMD)
.. Menu. RefList. Utilities. Help.
-----
.
.                               .Foreground Assembler.
.
.Command ==>.Z
.SAREA39
)AREA SAREA39
.ISPF Library:.
. . .Project . . .Z
. . .Group . . .Z      . . . .Z      . . . .Z      . . . .Z
. . .Type . . . .Z
. . .Member . . .Z      ..(Blank or pattern for member selection list).
.
.Other Partitioned or Sequential Data Set:.
. . .Data Set Name . . .Z
. . .
.List ID . . . .Z      . . .Assembler.
.Password . . .Z      . . .Z..1..High Level Assembler..2..Assembler H
.
.Assembler Options: (Options OBJECT and LIST generated automatically).
. . . ==>.Z
.
.Additional input libraries:.
. . . ==>.Z
. . . ==>.Z
. . . ==>.Z
)INIT
.ZVARS = '(ZCMD PRJ1 LIB1 LIB2 LIB3 LIB4 TYP1 MEM DSN LID PSWD ZASMOPT FHASM +
          FHAL1 FHAL2 FHAL3)'
.HELP = ISR41000
.ATTR(ZASMOPT)='CSRGRP(99) RADIO(ON)'
.ATTR(PSWD)='INTENS(NON)'
  &ZUT6 = 1
  &ZUT7 = 1
  &ZUT11 = 1
  &ZUT12 = 1
  &ZUT13 = 1
  &ZUT14 = 1
  &ZMLCSR = ' ' /* @M1A*/
.HELP = ISR41A00
&TYP1 = &ASMT /*ASSEMBLER TYPE VARIABLE NAME*/
IF (&ZORG = 'PS')
  IF (&LID = ' ') .CURSOR = LID
IF (&DSN ^= ' ')
  &MEM = ' ' /* @M1A*/
  IF (.CURSOR = ' ') .CURSOR = DSN
&LID = ' '
IF (&ZFTEMP = '')
  &ZFPACK = &ZFPACK
  VPUT (ZFPACK) SHARED
  &ZFTEMP = '0'
IF (&ZASMOPT ^= 1)
  IF (&ZASMOPT ^= 2)
    &ZASMOPT = 1

```

Figure 60. Foreground Assembler H definition (ISRFP01) (Part 3 of 5)

```

)REINIT
REFRESH(PRJ1 LIB1 LIB2 LIB3 LIB4 TYP1 MEM DSN FHAL1 FHAL2 FHAL3)
IF (&ZNXTMSG='ISRT') .CSRPOS = &ZCSRVP /* AUTOTYPE */
                                .CURSOR = &ZCSRVP /* AUTOTYPE */
ELSE &ZXZX = &Z /* AUTOTYPE */
&ZUT6 = 1
&ZUT7 = 1
&ZUT11 = 1
&ZUT12 = 1
&ZUT13 = 1
&ZUT14 = 1
IF (&ZMLCSR ^= ' ') /* @M1A*/
    .CURSOR = &ZMLCSR /* @M1A*/
)PROC
&ZCSRVP = .CURSOR /* AUTOTYPE */
&ZCSRVP = .CSRPOS /* AUTOTYPE */
&ZODSNLN = 0 /* AUTOTYPE */
&ZODSNMB = &Z /* AUTOTYPE */
&ZNAMES='ZCSRVP ZCSRVP PRJ1 LIB1 LIB2 LIB3 LIB4 TYP1 MEM '
&ZNAMES='&ZNAMES *.&ZODSNLN&ZODSNMB ZCMD'
IF (.CURSOR = DSN, FHAL1, FHAL2, FHAL3)
    &ZODSNLN = 56
    &ZODSNMB = &Z
    IF (.CURSOR = FHAL1) &ZODSNLN = 65
    IF (.CURSOR = FHAL1) &ZODSNMB = '%'
    IF (.CURSOR = FHAL2) &ZODSNLN = 65
    IF (.CURSOR = FHAL2) &ZODSNMB = '%'
    IF (.CURSOR = FHAL3) &ZODSNLN = 65
    IF (.CURSOR = FHAL3) &ZODSNMB = '%'
    &ZNAMES='ZCSRVP ZCSRVP * * * * * &ZCSRVP&ZODSNLN&ZODSNMB ZCMD'
PANEXIT((ZNAMES),LOAD,ISRAUTOT) /* AUTOTYPE */
IF (&ZNXTMSG='ISRT') EXIT /* AUTOTYPE */
VER(&ZASMOPT,NONBLANK)
VER(&ZASMOPT RANGE,1,2)
&ZUT6 = 0
&ZUT7 = 0
&ZUT11 = 0
&ZUT12 = 0
&ZUT13 = 0
&ZUT14 = 0
VGET (ZRDSN) SHARED /* REFERENCE LIST CODE */
IF (&ZRDSN ^= ' ') /* IF DATA SET SELECTED */
    &DSN = &ZRDSN /* PUT DSN VARIABLE */
    &ZRDSN = ' ' /* INTO PANEL */
    &ZRVOL = ' ' /* INTO PANEL */
VPUT (ZRDSN ZRVOL) SHARED /*
.CURSOR = DSN
.MSG = ISRDS003 /* MSG PENDING */

```

Figure 60. Foreground Assembler H definition (ISRFP01) (Part 4 of 5)

```

VGET (DSALSEL) SHARED /* */
IF (&DSALSEL ^= ' ') /* IF LIBRARY SELECTED */
  VGET (DSA1,DSA2,DSA3,DSA4,DSA5,DSA6,DSA7) SHARED
  &PRJ1 = &DSA1 /* PUT LIBRARY VARIABLES */
  &LIB1 = &DSA2 /* INTO PANEL */
  &LIB2 = &DSA3 /* . */
  &LIB3 = &DSA4 /* . */
  &LIB4 = &DSA5 /* . */
  &TYP1 = &DSA6 /* . */
  &MEM = &DSA7 /* . */
  &DSN = ' ' /* BLANK OUT DSN */
  &DSALSEL = ' ' /* CLEAR LIBRARY SELECTION */
VPUT (DSALSEL) SHARED /* */
.CURSOR = MEM /* MSG PENDING */
.MSG = ISRDS003 /* */

IF (&ZCMD ^= ' ') .MSG = ISPZ001 /* INVALID COMMAND */
IF (&DSN = ' ') /* IF NOT OTHER DATA SET, */
  VER (&PRJ1,NB) /* VERIFY LIBRARY FIELDS ARE */
  VER (&LIB1,NB) /* INPUT */
  VER (&TYP1,NB) /* */
IF (&DSN ^= ' ') /* @M1A*/
  VER(&DSN DSNAMEFM)
IF (&LID ^= '*') VER (&LID,NAME) /* LIST ID MUST BE VALID NAME */
VER (&PSWD,INCLUDE,ALPHA,NUM,MSG=ISR609) /* Check password @V9A*/
IF (&ZASMOPT = 1)
  &FHALEV = 'HLASM'
IF (&ZASMOPT = 2)
  &FHALEV = 'HASM'
IF (&FHAL1 ^= ' ') /* @M2A*/
  VER (&FHAL1,DSNAMEPQ) /* VERIFY ADDITIONAL LIB1@DSNQ*/
IF (&FHAL2 ^= ' ') /* @M2A*/
  VER (&FHAL2,DSNAMEPQ) /* VERIFY ADDITIONAL LIB2@DSNQ*/
IF (&FHAL3 ^= ' ') /* @L2A*/
  VER (&FHAL3,DSNAMEPQ) /* VERIFY ADDITIONAL LIB2@DSNQ*/
&ASMT = &TYP1 /* SAVE ASSEMBLER TYPE */
&ZSEL = 'CMD(%ISRFC01)' /* EXECUTE ASSEMBLE CLIST */
/* Begin @L3A */
&ZSYSDS1 = &FHAL1 /* Fill fields for ISRJFSYS */
&ZSYSDS2 = &FHAL2 /* to use as input. */
&ZSYSDS3 = &FHAL3 /* ZSYSDS? is a qualified */
&ZSYSCUR1 = 'FHAL1' /* dataset. */
&ZSYSCUR2 = 'FHAL2' /* ZSYSCUR? is were the cursor*/
&ZSYSCUR3 = 'FHAL3' /* is placed on a error. */
VPUT (ZSYSDS1 ZSYSDS2 ZSYSDS3 ZSYSCUR1 ZSYSCUR2 ZSYSCUR3) SHARED
/* End @L3A */
VPUT (PRJ1,LIB1,LIB2,LIB3,LIB4,ASMT,FHASM) PROFILE /* OY14824*/
VPUT (FHAL1,FHAL2,FHAL3,FHALEV,ZASMOPT) PROFILE
VPUT (PRJ1,LIB1,LIB2,LIB3,LIB4,ASMT,FHASM,DSN,LID) SHARED /* @L2C*/
&ZFBROWS = 'ISRBROB ' /*BROWSE LISTING , IF BLANK NO AUTO BROWSE */
&ZFPRT = 'ISRFPRT' /*PRINT PANEL NAME, IF BLANK NO AUTO PRINT PNL*/
VPUT (ZFBROWS,ZFPRT,FHAL1,FHAL2,FHAL3,FHALEV) SHARED /* OW10516*/
)END
/* 5694-A01 (C) COPYRIGHT IBM CORP 1980, 2001 */
/* ISPD TLC Release: 5.2. Level: PID */
/* z/OS 01.02.00. Created - Date: 10 Mar 2001, Time: 14:25 */
/* OW21977 - 960813 - OS/390 R2 ROLLUP APAR - OW19891, OW20382 */
/* OW10516 - 950113 - Add High Level Assembler. GT4045 - MOS */

```

Figure 60. Foreground Assembler H definition (ISRFP01) (Part 5 of 5)

Table 26 on page 139 lists the names of the ISPF-supplied panels and CLISTs for the Foreground processing option.

Table 26. ISPF-supplied panels and CLISTs for foreground processing option

Option	Description	Panel ID	CLIST ID
-	FOREGROUND SELECTION MENU	ISRFPA	--
1	ASSEMBLER	ISRF01	ISRFC01
2	VS COBOL II	ISRF02	ISRFC02
3	FORTRAN COMPILE	ISRF03	ISRFC03
5	PLI OPTIMIZER COMPILE	ISRF05	ISRFC05
6	VS PASCAL COMPILE	ISRF06	ISRFC06
7	BINDER/LINK EDIT	ISRF07	ISRFC07
9	SCRIPT/VS	ISRF09	ISRFC09
10	VS COBOL II DEBUG	--	--
10A	COBOL INTERACTIVE DEBUG	ISRF10A	ISRFC10A
11	FORTRAN INTERACTIVE DEBUG	ISRF11	ISRFC11
12	MEMBER PARTS LIST	ISRF12	ISRFC12
13	C/370 COMPILE	--	--
14	REXX/370 COMPILE	--	--
15	ADA/370 COMPILE	--	--
16	AD/CYCLE C/370 COMPILE	--	--
17	AD/CYCLE C/370 COBOL/370	--	--
18	ISPD TLC	ISPCP01	--
19	OS/390 C/C++	--	--

Figure 61 shows the overall flow of control for foreground processing.

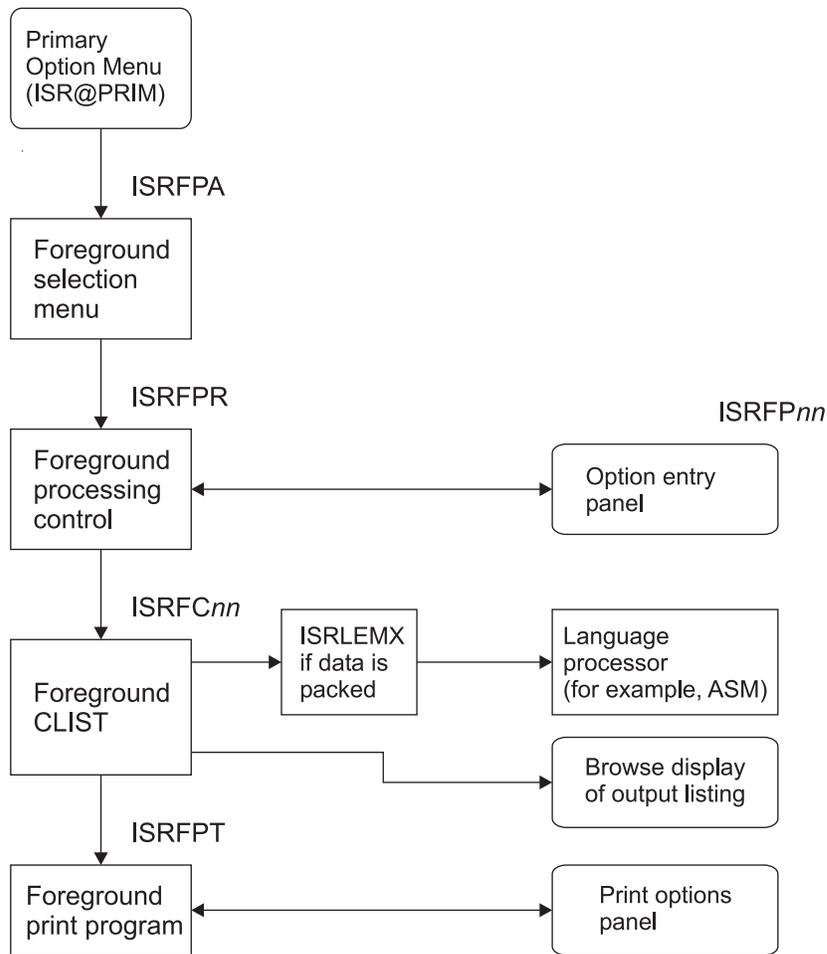


Figure 61. Foreground processing flow

ISRFPA is the Foreground Selection Panel. Each option on this menu translates to these selection keywords:

```
'PGM(ISRFPR) PARM((ISRFPnn) nn) NEWPOOL'
```

For all options, program ISRFPR receives control and is passed a parameter containing a list of panel names and the option number. The list of panel names is parsed and the first panel found in the ISPP LIB concatenation sequence is displayed.

**Attention:** To avoid possible conflicts with ISPF coding, do not use the numbers 12 or 19 for your options.

Each option panel (ISRFP01-ISRFP12) sets certain dialog variables that the ISRFPR program interprets. See “Required option entry panel variables” on page 141. One of these variables, ZSEL, is set to a string of selection keywords (following the rules for selection panels) that indicate which CLIST name (CMD) or load module name (PGM) receives control next.

**Note:** For foreground processing, ZSEL cannot be set to the PANEL keyword.

The designated CLIST or program does not receive control immediately, because ISRFPR uses the DISPLAY service to display the option panel, rather than the SELECT service. After the option panel is displayed, ISRFPR allocates the

user-designated libraries or data sets, displays a member list (if one is required), and scans the concatenated sequence of libraries (if specified) to find the designated member. Then ISRFPR invokes the SELECT service, passing as input the string of selection keywords specified in ZSEL. This causes the corresponding CLIST or program to receive control. (The distributed Foreground option uses CLISTs exclusively.)

The CLIST sets up and issues the appropriate TSO commands to invoke the language processor. The CLIST also initiates automatic browsing of the list data set and subsequent display of the Foreground Print Options panel. When the CLIST finishes, ISRFPR regains control and redisplay the option entry panel.

The PDF interface to VS COBOL II Interactive Debug uses the Debug Productivity Aid (DPA) integrated into VS COBOL II. If your installation does not have VS COBOL II Release 2 or if the panel for DPA (IGZTPIN2) is not in your ISPLLIB concatenation, PDF displays a panel stating DPA is not installed or could not be accessed.

The PDF FORTRAN Interactive Debug option supports both FORTRAN Interactive Debug Version 2 (5668-903) and FORTRAN Interactive Debug Version 1 (5734-F05). If FORTRAN Interactive Debug Version 2 is installed (panel AFFFFP11 is found in the ISPLLIB concatenation), PDF executes the Debug Dialog supplied by FORTRAN Interactive Debug Version 2. If FORTRAN Interactive Debug Version 1 is installed (panel DDBFP11), PDF executes that Debug Dialog. PDF looks for FORTRAN Interactive Debug Version 2 first, then Version 1, and finally its own Debug Dialog.

### Required option entry panel variables

These dialog variables are explicitly required and must be defined either in the )BODY, )PROC or )INIT section of the suboption panel displayed by ISRFPR. For example, if your application does not require libraries two through four, initialize LIB2, LIB3, and LIB4 to the system variable &Z (blank) in the )INIT section of the suboption panel. This would fill the requirement and allow you to leave these fields out the )BODY section of the suboption panel.

<b>PRJ1</b>	ISPF project name
<b>LIB1</b>	First ISPF library
<b>TYP1</b>	ISPF library type (initially set to nulls in ISRFPR)
<b>MEM</b>	ISPF member name (required if DSN is blank)
<b>DSN</b>	“Other” data set name
<b>ZSEL</b>	Selection string used by ISRFPR to select either a CLIST (CMD) or a program (PGM) with parameters.
<b>LIB2</b>	Second ISPF library
<b>LIB3</b>	Third ISPF library
<b>LIB4</b>	Fourth ISPF library

This field is optional:

<b>PSWD</b>	OS password
-------------	-------------

The ISRFPR program verifies and processes these variables and then stores these variables in the shared variable pool, so that they can be referenced by the selected CLIST or program.

<b>ZDSQMEM</b>	fully qualified input data set name (without quotes), with the member name in parentheses.
<b>ZDSQ</b>	fully qualified input data set name (without quotes), for the first input data set (without member name)
<b>ZDSQ2</b>	Same as ZDSQ for the second input data set (library)
<b>ZDSQ3</b>	Same as ZDSQ for the third input data set (library)
<b>ZDSQ4</b>	Same as ZDSQ for the fourth input data set (library)
<b>ZDS</b>	Same as ZDSQ without the last qualifier (if ZDSQ has only a one level qualifier, then ZDSQ = ZDS.)
<b>ZDS2</b>	Same as ZDS for the second input data set (library)
<b>ZDS3</b>	Same as ZDS for the third input data set (library)
<b>ZDS4</b>	Same as ZDS for the fourth input data set (library)
<b>ZMEM</b>	Input member name (blank for a sequential data set)
<b>ZORG</b>	Input data set organization: PO=partitioned PS=sequential

**Note:** If an asterisk (\*) is entered as the member name (from the option panel), ISRFPR does not process the member name. Instead, it sets the ZMEM variable to blank. The CLIST must handle this case.

### Other option entry panel variables

ISRFPR does not verify or process other variables from the option panel, such as list ID, compiler options, and additional input libraries. Instead, ISRFPR stores these variables in the shared variable pool (by including VPUT statements in the panel definition) so the CLIST can process them.

Several variables in the option panel are saved in the user profile (again by including VPUT statements in the panel definition) so that they are retained across sessions. In general, variables saved in the profile are also saved in the shared pool to prevent their being modified by another process in split-screen mode. Variables in the profile can be referenced by either screen, but those in the shared pool can be referenced only by the screen in which they are set.

An optional variable, ZSEL2, can be set in the panel. It causes ISRFPR to select a second CLIST or program after successful completion (return code = 0) of the first CLIST or program. ZSEL2 is used with the COBOL and FORTRAN Interactive Debug options.

These sections describe additional variables.

### Variables that control automatic Browse and Print

The process of automatically invoking the Browse function for a generated list data set, and subsequently displaying the Foreground Print Options panel is completely external to the ISRFPR program. The ZFBROWS and ZFPRINT variables control these functions. These variables reside in the shared variable pool. Select the distributed Foreground option on the ISRFPA Foreground Selection panel to store ZFBROWS and ZFPRINT in the shared variable pool. ZFBROWS is set to "ISRBROB" (use any nonblank name) and ZFPRINT is set to "ISRFPPRT" (the print panel name).

Use the BROWSE service to invoke the browse function from each CLIST. Use the SELECT service to invoke the print function from each CLIST. SELECT service invokes the ISRFPT program (load module). ISRFPT is distributed with PDF.

ZFPRINT is the variable that is set to the name of the panel to set the print options panel name the dialog developer calls.

The ZFLID variable in ISRFPPRT (Foreground Print Options panel) should be set to the data set name you are going to print. Use the VPUT service to put both the ZFPRINT and ZFLID variables in the shared variable pool. If you do not want to invoke automatic browse or print, set the corresponding variables to blank in either the individual option panels or CLISTs. If you want to set the variables from a panel, remember to use VPUT to store the variables in the shared variable pool so the CLISTs can access them.

When the language processor sends a return code greater than 12 to the ISPF-supplied CLISTs, CLIST bypasses automatic browse and display of the print panel. The exceptions to this are the PL/I Checkout compiler and the COBOL and FORTRAN Interactive Debug programs. The CLISTs for these programs do not display browse or print if the return code is a system return code (for example, "SOC1").

### **Variables that control option panel redisplay**

When an option is processed, normally the original option panel is redisplayed when the specified CLIST or program finishes. ISRFPR stores the original panel name in ZNEXTPN, a variable that controls the next panel to be displayed. You can change the value of ZNEXTPN in your CLIST or panel. For example, to display the linkage edit panel (ISRF07) after processing the FORTRAN panel (ISRF03), set ZNEXTPN to ISRF07 either in the )PROC section of the FORTRAN panel or in the FORTRAN CLIST (ISRFC03). Then place the variable in the shared pool. You can extend this type of panel linkage to any length you want.

### **Variables used by Foreground CLISTs**

In the distributed Foreground option, all option panels select a CLIST to set up and issue the TSO commands. (Only a CLIST or a program can be selected from the option panel.) To obtain variables set from the panel and by program ISRFPR, the CLIST references the shared variable pool. Based on these variables, the CLIST sets up and issues the TSO command required to invoke the language processor. The return code from the TSO command is saved in variable ZFPRFC, which is referenced in various error messages. After completion of the TSO command, the check for automatic browse and print is made. The CLIST issues any log messages. The CLIST also does some specific error checking and can override ISRFPR messages by setting variable ZFPRFC to the return code, issuing a VPUT of ZFPRFC, and invoking the SETMSG service.

The FORTRAN and COBOL interactive CLISTs differ from other CLISTs. These interactive programs attempt to read subsequent lines from the CLIST as input. Therefore, the CLIST is divided into two parts. All functions after the TSO command are in the second CLIST, which is referenced by the ZSEL2 variable described in "Required option entry panel variables" on page 141.

### **Foreground modifications**

Most of the Foreground processing logic resides in the option panels and associated CLISTs to make modifications as easy as possible. Before modifying existing options or adding new ones, you should study one or more of the

distributed panel/CLIST pairs to understand the relationships that exist among the panel, the CLIST, and the program ISRFPR. In particular, be aware of these items:

- Additional variables such as LANG and ZORG are used to pass information to log messages, error messages, and panels.
- These commands cannot be invoked under the Foreground option: SPF, PDF, ISPF, ISPSTART, LOGON, LOGOFF, TEST, or a CALL to an authorized program.
- If the CLIST returns a nonzero return code in the EXIT statement, ISRFPR does not attempt to process any second CLIST that was specified by ZSEL2.

**Steps to add a new Foreground primary option to PDF:** The steps required to add a new PDF primary option that uses the Foreground processing mechanism are listed here.

1. Add the new option (for example, '10') to the ISPF Primary Option Menu, panel ISR@PRIM. In the translated value for option 10, use the PANEL keyword to specify the name of the selection panel to be displayed next. For example:

```
)PROC
  &ZSEL = TRANS( TRUNC(&ZCMD, '.' )
                . . .
                10, 'PANEL(XYZ) '
```

2. Add new selection panel XYZ to the panel library. Use panel ISRFFPA as a model. For each option, use the PGM keyword to specify that program ISRFPR is to receive control, and use the PARM keyword to pass the name of the option panel. For example:

```
)PROC
  &ZSEL = TRANS( TRUNC(&ZCMD, '.' )
                . . .
                2, 'PGM(ISRFPR) PARM((FORNEW) 2) '
```

3. Proceed as specified in “Steps to add a new Foreground option to the Foreground Selection Panel,” starting at step 2.

**Steps to add a new Foreground option to the Foreground Selection Panel:** The steps required to add a new option to the PDF Foreground Selection Panel are listed here.

1. Add new option (for example, '99') to the Foreground Selection Panel, ISRFFPA. Use the PGM keyword to specify that program ISPFPR is to receive control, and use the PARM keyword to pass the name of the new option panel. For example:

```
)PROC
  &ZSEL = TRANS( TRUNC(&ZCMD, '.' )
                . . .
                99, 'PGM(ISRFPR) PARM((FORNEW) 99) '
                . . .
```

2. Add new option panel FORNEW to the panel library. Use one of the distributed option panels (for example, ISRFP01) as a model.
3. Develop a corresponding CLIST (referenced from panel FORNEW by the ZSEL variable). Use one of the distributed CLISTS (for example, ISRFC01) as a model. Add the CLIST to a library accessible to ddname SYSPROC.

## Batch processing panels, CLISTs, and skeletons

The Batch option uses ISPF dialog management services. Figure 62 and Figure 63 on page 149 show the Batch Selection and Batch JCL generation panel formats. See the *z/OS ISPF Dialog Developer's Guide and Reference* for a general description of panel definition formats.

**Note:** In Figure 62 and Figure 63 on page 149, attribute characters have been replaced by blanks.

```
)PANEL KEYLIST(ISRSAB,ISR)
)ATTR DEFAULT(...) FORMAT(MIX)          /* ISRJPA - ENGLISH - 5.6 */
0B TYPE(AB)
0D TYPE(PS)
04 TYPE(ABSL) GE(ON)
05 TYPE(PT)
09 TYPE(FP)
0A TYPE(NT)
0C TYPE(NT) SKIP(ON)
11 TYPE(SAC)
22 TYPE(WASL) SKIP(ON) GE(ON)
08 TYPE(CH)
10 TYPE(ET)
26 AREA(SCRL) EXTEND(ON)
27 TYPE(CEF) PADC(USER) CKBOX(ON)
28 TYPE(NEF) CAPS(ON) PADC(USER)
)ABC DESC('Menu') MNEM(1)
PDC DESC('Settings') UNAVAIL(ZPM1) MNEM(1) ACC(CTRL+S)
ACTION RUN(ISRRROUTE) PARM('SET')
PDC DESC('View') UNAVAIL(ZPM2) MNEM(1) ACC(CTRL+V)
ACTION RUN(ISRRROUTE) PARM('BR1')
PDC DESC('Edit') UNAVAIL(ZPM3) MNEM(1) ACC(CTRL+E)
ACTION RUN(ISRRROUTE) PARM('ED1')
PDC DESC('ISPF Command Shell') UNAVAIL(ZPM4) MNEM(6) ACC(CTRL+C)
ACTION RUN(ISRRROUTE) PARM('C1')
PDC DESC('Dialog Test...') UNAVAIL(ZPM5) MNEM(8) ACC(CTRL+T)
ACTION RUN(ISRRROUTE) PARM('DAL')
PDC DESC('Other IBM Products...') UNAVAIL(ZPM6) MNEM(1) ACC(CTRL+O)
ACTION RUN(ISRRROUTE) PARM('OIB')
PDC DESC('SCLM') UNAVAIL(ZPM7) MNEM(3) ACC(CTRL+L)
ACTION RUN(ISRRROUTE) PARM('SCL')
PDC DESC('ISPF Workplace') UNAVAIL(ZPM8) MNEM(6) ACC(CTRL+W)
ACTION RUN(ISRRROUTE) PARM('WRK')
PDC DESC('Status Area...') UNAVAIL(ZPMS) MNEM(8) ACC(CTRL+A)
ACTION RUN(ISRRROUTE) PARM('SAM')
PDC DESC('Exit') MNEM(2) PDSEP(ON) ACC(CTRL+X) ACTION RUN(EXIT)
)ABCINIT
.ZVARS=ISR@OPT
)ABC DESC('Utilities') MNEM(1)
PDC DESC('Library') UNAVAIL(ZUT1) MNEM(1) ACC(ALT+1)
ACTION RUN(ISRRROUTE) PARM('U1')
PDC DESC('Data set') UNAVAIL(ZUT2) MNEM(1) ACC(ALT+2)
ACTION RUN(ISRRROUTE) PARM('U2')
PDC DESC('Move/Copy') UNAVAIL(ZUT3) MNEM(1) ACC(ALT+3)
ACTION RUN(ISRRROUTE) PARM('U3')
PDC DESC('Data Set List') UNAVAIL(ZUT4) MNEM(2) ACC(ALT+4)
ACTION RUN(ISRRROUTE) PARM('U4')
PDC DESC('Reset Statistics') UNAVAIL(ZUT5) MNEM(5) ACC(ALT+5)
ACTION RUN(ISRRROUTE) PARM('U5')
PDC DESC('Hardcopy') UNAVAIL(ZUT6) MNEM(1) ACC(ALT+6)
ACTION RUN(ISRRROUTE) PARM('U6')
```

Figure 62. Batch selection panel definition (ISRJPA) (Part 1 of 4)



```

)INIT
.ZVARS = '(ZCMD ZDSCKOV ZBPKEDV BJC1 BJC2 BJC3 BJC4)'
&ZWINTTL = ' '
.HELP = ISR50000
&ZDSCKOV = ' '
&ZBPKEDV = ' '
&ZUT6 = 1
&ZUT7 = 1
&ZUT11 = 1
&ZUT12 = 1
&ZUT13 = 1
&ZUT14 = 1
IF (&ZDSCKO = ' ')
  &ZDSCKO = Y
&ZDSCKO = TRANS(TRUNC(&ZDSCKO,1),N,NO,*,YES)
&ZDSCKOV = TRANS(&ZDSCKO YES, '/' NO, ' ')
IF (&ZBPKED = &Z)
  &ZBPKED = N
&ZBPKED = TRANS(TRUNC(&ZBPKED,1),Y,YES,*,NO)
&ZBPKEDV = TRANS(&ZBPKED YES, '/' NO, ' ')
IF (&ZGUI = ' ')
  &MULTIPMT='Enter "/" to select option '
ELSE
  &MULTIPMT='Check box to select option '
.CURSOR = 'ZCMD'
)REINIT
REFRESH(ZDSCKOV ZBPKEDV)
&ZUT6 = 1
&ZUT7 = 1
&ZUT11 = 1
&ZUT12 = 1
&ZUT13 = 1
&ZUT14 = 1
)PROC
&ZUT6 = 0
&ZUT7 = 0
&ZUT11 = 0
&ZUT12 = 0
&ZUT13 = 0
&ZUT14 = 0
IF (&ZDSCKOV = ' ')
  &ZDSCKO = 'NO'
ELSE
  &ZDSCKO = 'YES'
&ZDSCKO = TRUNC(&ZDSCKO,1) /*DATA SET (DS) CHECK: */
&ZDSCHK = TRANS(TRUNC(&ZDSCKO,1),Y,Y,N,N)
&ZDSCKO = TRANS(TRUNC(&ZDSCKO,1),Y,YES,N,NO)
IF (&ZBPKEDV = ' ')
  &ZBPKED = 'NO'
ELSE
  &ZBPKED = 'YES'

```

Figure 62. Batch selection panel definition (ISRJPA) (Part 3 of 4)

```

&ZBPKED = TRUNC(&ZBPKED,1)
&ZBPKED = TRANS(TRUNC(&ZBPKED,1),Y,YES,N,NO)
&ZBPACK = TRANS(TRUNC(&ZBPKED,1),Y,YES,N,NO)
VER (&ZDSCKO,NB,LIST,YES,NO) /* Y= VERIFY DSN;N= NO VERIFICATION*/
VER (&ZBPKED,NB,LIST,YES,NO) /* Y = EXPAND PACKED DATA */
&DSN = ' ' /* INITIALIZE DATA SET NAME FIELD */
VPUT (ZDSCHK,ZDSCKO,DSN) SHARED /* PLACE IN SHARED POOL FOR ISRJB2 */
&RTNPNL = ISRJPB
VPUT (BJC1,BJC2,BJC3,BJC4,ZBPACK,ZBPKED) PROFILE
&ZCMDWRK = &Z
IF (&ZCMD = &Z)
  &ZCMDWRK = TRUNC(&ZCMD,'.')
  &ZTRAIL=.TRAIL
  IF (&ZCMDWRK = &Z)
    .MSG = ISRU000
&ZSEL = TRANS (TRUNC (&ZCMD,'.')
  1,'PGM(ISRJB2) PARM((ISRJP01) 1) NEWPOOL'
  2,'PGM(ISRJB2) PARM((ISRJP02) 2) NEWPOOL'
  3,'PGM(ISRJB2) PARM((ISRJP03) 3) NEWPOOL'
  5,'PGM(ISRJB2) PARM((ISRJP05) 5) NEWPOOL'
  6,'PGM(ISRJB2) PARM((ISRJP06) 6) NEWPOOL'
  7,'PGM(ISRJB2) PARM((ISRJP07) 7) NEWPOOL'
  10,'PGM(ISRJB2) PARM((ISRJP10) 10) NEWPOOL'
  12,'PGM(ISRJB2) PARM((ISRJP12) 12) NEWPOOL'
  13,'PGM(ISRJB2) PARM((EDCJP13,ISRJP13) 13) NEWPOOL'
  14,'PGM(ISRJB2) PARM((FANJP14,ISRJP14) 14) NEWPOOL'
  15,'PGM(ISRALTDI) PARM(EVGJP15,,ISRJP15,*) NOCHECK'
  16,'PGM(ISRJB2) PARM((EDCJP16,ISRJP16) 16) NEWPOOL'
  18,'CMD(ISPD TLC (PANEL SUBMIT RETURN)) MODE(FSCR)'
  19,'PGM(ISRALTDI) PARM(CBC3PE5A,+,ISRJP19,*,ISRJP19A) NOCHECK'
  ' ',' '
  *,'?')
)PNTS
FIELD(ZPS01001) VAR(ZCMD) VAL(1)
FIELD(ZPS01002) VAR(ZCMD) VAL(7)
FIELD(ZPS01003) VAR(ZCMD) VAL(15)
FIELD(ZPS01004) VAR(ZCMD) VAL(2)
FIELD(ZPS01005) VAR(ZCMD) VAL(10)
FIELD(ZPS01006) VAR(ZCMD) VAL(16)
FIELD(ZPS01007) VAR(ZCMD) VAL(3)
FIELD(ZPS01008) VAR(ZCMD) VAL(12)
FIELD(ZPS01009) VAR(ZCMD) VAL(18)
FIELD(ZPS01010) VAR(ZCMD) VAL(5)
FIELD(ZPS01011) VAR(ZCMD) VAL(13)
FIELD(ZPS01012) VAR(ZCMD) VAL(19)
FIELD(ZPS01013) VAR(ZCMD) VAL(6)
FIELD(ZPS01014) VAR(ZCMD) VAL(14)
)END
/* 5647-A01 (C) COPYRIGHT IBM CORP 1980, 1999 */
/* ISPD TLC Release: 5.6. Level: PID */
/* z/OS 01.06.00. Created - Date: 19 Nov 2003, Time: 12:37 */

```

Figure 62. Batch selection panel definition (ISRJPA) (Part 4 of 4)

```

)PANEL KEYLIST(ISRSAB,ISR)
)ATTR DEFAULT(...) FORMAT(MIX) /* ISRJPB - ENGLISH - 5.6 */
0B TYPE(AB)
0D TYPE(PS)
04 TYPE(ABSL) GE(ON)
05 TYPE(PT)
09 TYPE(FP)
0A TYPE(NT)
0C TYPE(NT) SKIP(ON)
11 TYPE(SAC)
22 TYPE(WASL) SKIP(ON) GE(ON)
08 TYPE(CH)
10 TYPE(ET)
26 AREA(SCRL) EXTEND(ON)
27 TYPE(CEF) PADC(USER) CKBOX(ON)
28 TYPE(NEF) CAPS(ON) PADC(USER)
)ABC DESC('Menu') MNEM(1)
PDC DESC('Settings') UNAVAIL(ZPM1) MNEM(1) ACC(CTRL+S)
ACTION RUN(ISRRROUTE) PARM('SET')
PDC DESC('View') UNAVAIL(ZPM2) MNEM(1) ACC(CTRL+V)
ACTION RUN(ISRRROUTE) PARM('BR1')
PDC DESC('Edit') UNAVAIL(ZPM3) MNEM(1) ACC(CTRL+E)
ACTION RUN(ISRRROUTE) PARM('ED1')
PDC DESC('ISPF Command Shell') UNAVAIL(ZPM4) MNEM(6) ACC(CTRL+C)
ACTION RUN(ISRRROUTE) PARM('C1')
PDC DESC('Dialog Test...') UNAVAIL(ZPM5) MNEM(8) ACC(CTRL+T)
ACTION RUN(ISRRROUTE) PARM('DAL')
PDC DESC('Other IBM Products...') UNAVAIL(ZPM6) MNEM(1) ACC(CTRL+O)
ACTION RUN(ISRRROUTE) PARM('OIB')
PDC DESC('SCLM') UNAVAIL(ZPM7) MNEM(3) ACC(CTRL+L)
ACTION RUN(ISRRROUTE) PARM('SCL')
PDC DESC('ISPF Workplace') UNAVAIL(ZPM8) MNEM(6) ACC(CTRL+W)
ACTION RUN(ISRRROUTE) PARM('WRK')
PDC DESC('Status Area...') UNAVAIL(ZPMS) MNEM(8) ACC(CTRL+A)
ACTION RUN(ISRRROUTE) PARM('SAM')
PDC DESC('Exit') MNEM(2) PDSEP(ON) ACC(CTRL+X) ACTION RUN(EXIT)
)ABCINIT
.ZVARS=ISR@OPT
)ABC DESC('Utilities') MNEM(1)
PDC DESC('Library') UNAVAIL(ZUT1) MNEM(1) ACC(ALT+1)
ACTION RUN(ISRRROUTE) PARM('U1')
PDC DESC('Data set') UNAVAIL(ZUT2) MNEM(1) ACC(ALT+2)
ACTION RUN(ISRRROUTE) PARM('U2')
PDC DESC('Move/Copy') UNAVAIL(ZUT3) MNEM(1) ACC(ALT+3)
ACTION RUN(ISRRROUTE) PARM('U3')
PDC DESC('Data Set List') UNAVAIL(ZUT4) MNEM(2) ACC(ALT+4)
ACTION RUN(ISRRROUTE) PARM('U4')
PDC DESC('Reset Statistics') UNAVAIL(ZUT5) MNEM(5) ACC(ALT+5)
ACTION RUN(ISRRROUTE) PARM('U5')
PDC DESC('Hardcopy') UNAVAIL(ZUT6) MNEM(1) ACC(ALT+6)
ACTION RUN(ISRRROUTE) PARM('U6')
PDC DESC('Download...') UNAVAIL(ZUTDT) MNEM(2) ACC(ALT+7)
ACTION RUN(ISRRROUTE) PARM('UDT')
PDC DESC('Outlist') UNAVAIL(ZUT7) MNEM(2) ACC(ALT+8)
ACTION RUN(ISRRROUTE) PARM('U8')
PDC DESC('Commands...') UNAVAIL(ZUT8) MNEM(1) ACC(ALT+9)
ACTION RUN(ISRRROUTE) PARM('U9')
PDC DESC('Reserved') UNAVAIL(ZUT9) MNEM(6) ACTION RUN(ISRRROUTE) PARM('U10')
PDC DESC('Format') UNAVAIL(ZUT10) MNEM(1) ACC(ALT+F1)
ACTION RUN(ISRRROUTE) PARM('U11')

```

Figure 63. Batch JCL generation panel definition (ISRJPB) (Part 1 of 4)

```

PDC DESC('SuperC') UNAVAIL(ZUT11) MNEM(1) PDSEP(ON) ACC(CTRL+F2)
  ACTION RUN(ISRROUTE) PARM('U12')
PDC DESC('SuperCE') UNAVAIL(ZUT12) MNEM(3) ACC(CTRL+F3)
  ACTION RUN(ISRROUTE) PARM('U13')
PDC DESC('Search-For') UNAVAIL(ZUT13) MNEM(2) ACC(CTRL+F4)
  ACTION RUN(ISRROUTE) PARM('U14')
PDC DESC('Search-ForE') UNAVAIL(ZUT14) MNEM(4) ACC(CTRL+F5)
  ACTION RUN(ISRROUTE) PARM('U15')
)ABCINIT
.ZVARS=PDFUTIL
  &zut9 = '1'
)ABC DESC('Help') MNEM(1)
PDC DESC('General') MNEM(1) ACTION RUN(TUTOR) PARM('ISR50000')
PDC DESC('Assembler') MNEM(1) ACTION RUN(TUTOR) PARM('ISR51000')
PDC DESC('COBOL') MNEM(1) ACTION RUN(TUTOR) PARM('ISR52000')
PDC DESC('VS Fortran') MNEM(4) ACTION RUN(TUTOR) PARM('ISR53000')
PDC DESC('PL/I') MNEM(2) ACTION RUN(TUTOR) PARM('ISR55000')
PDC DESC('VS Pascal') MNEM(4) ACTION RUN(TUTOR) PARM('ISR56000')
PDC DESC('Binder/Link editor') MNEM(1) ACTION RUN(TUTOR) PARM('ISR57000')
PDC DESC('VS COBOL II debug') MNEM(1) ACTION RUN(TUTOR) PARM('ISR5A000')
PDC DESC('Member parts list') MNEM(1) ACTION RUN(TUTOR) PARM('ISR5C000')
PDC DESC('C/370') MNEM(3) ACTION RUN(TUTOR) PARM('ISR5D000')
PDC DESC('REXX/370') MNEM(1) ACTION RUN(TUTOR) PARM('ISR5E000')
PDC DESC('ADA/370') MNEM(2) ACTION RUN(TUTOR) PARM('ISR5F000')
PDC DESC('AD/Cycle C/370') MNEM(5) ACTION RUN(TUTOR) PARM('ISR5G000')
PDC DESC('ISPD TLC') MNEM(5) ACTION RUN(TUTOR) PARM('ISR5I000')
PDC DESC('OS/390 C/C++') MNEM(1) ACTION RUN(TUTOR) PARM('ISR5J000')
PDC DESC('Appendices') MNEM(4) ACTION RUN(TUTOR) PARM('ISR00004')
)ABCINIT
.ZVARS=JPBHELP
)BODY CMD(ZCMD)
.. Menu. Utilities. Help.
-----
.                                     .Batch Selection Panel.
.
.Option ==>.Z
.SAREA38
)AREA SAREA38
.1 .Assembler          ...7 .*Binder/Link editor ...15.*ADA/370      ..
.2 .COBOL              ...10.*VS COBOL II debug ...16.*AD/Cycle C/370   ..
.3 .VS FORTRAN        ...12. Member Parts List ...18. ISPD TLC       ..
.5 .PLI               ...13.*C/370          ...19.*OS/390 C/C++   ..
.6 .VS PASCAL         ...14.*REXX/370          .
.
.          ..
.          ..*.No packed data support.
.Instructions:
. ..Enter option to continue generating JCL,.CANCEL.command to exit without.
. ..submitting job or.END.command to &ZBMSG
. ...
. ..&multipmt
. ..Z..Source data online
. ..Z..Source data packed
.
.Job Statement Information:
.
.==>.Z
.==>.Z
.==>.Z
.==>.Z

```

Figure 63. Batch JCL generation panel definition (ISRJPB) (Part 2 of 4)

```

)INIT
.ZVARS = '(ZCMD ZDSC ZBPK BJC1 BJC2 BJC3 BJC4)'
&ZWINTTL = ' '
.HELP = ISR50000
&ZDSC = ' '
&ZBPK = ' '
&ZUT6 = 1
&ZUT7 = 1
&ZUT11 = 1
&ZUT12 = 1
&ZUT13 = 1
&ZUT14 = 1
IF (&ZDSCKO = ' ')
  &ZDSCKO = Y
&ZDSCKO = TRANS(TRUNC(&ZDSCKO,1),N,NO,*,YES)
&ZDSC = TRANS(&ZDSCKO NO,' ' YES, '/')
IF (&ZBPKED = &Z)
  &ZBPKED = N
&ZBPKED = TRANS(TRUNC(&ZBPKED,1),Y,YES,*,NO)
&ZBPK = TRANS(&ZBPKED NO,' ' YES, '/')
IF (&ZJOBSTEP = 'YES')
  &ZBMSG = 'submit job.'
IF (&ZJOBSTEP = 'NO ')
  &ZBMSG = 'exit without submitting job.'
IF (&ZGUI = ' ')
  &MULTIPMT='Enter "/" to select option  '
ELSE
  &MULTIPMT='Check box to select option  '
.CURSOR = 'ZCMD'
)REINIT
REFRESH(ZDSC ZBPK)
&zut6 = 1
&zut7 = 1
&zut11 = 1
&zut12 = 1
&zut13 = 1
&zut14 = 1
)PROC
&zut6 = 0
&zut7 = 0
&zut11 = 0
&zut12 = 0
&zut13 = 0
&zut14 = 0
&ZDSCKO = TRANS(&ZDSC ' ', 'NO' *, 'YES')
&ZBPKED = TRANS(&ZBPK ' ', 'NO' *, 'YES')
&ZDSCKO = TRUNC(&ZDSCKO,1) /*DATA SET (DS) CHECK: */
VER (&ZDSCKO,NB,LIST,Y,N) /* Y= VERIFY DSN;N= NO VERIFICATION*/
&ZDSCHK = TRANS(TRUNC(&ZDSCKO,1),Y,Y,N,N)
&ZBPKED = TRUNC(&ZBPKED,1)
VER (&ZBPKED,NB,LIST,Y,N) /* Y = EXPAND PACKED DATA */
&ZBPKED = TRANS(TRUNC(&ZBPKED,1),Y,YES,N,NO)
&ZBPACK = TRANS(TRUNC(&ZBPKED,1),Y,YES,N,NO)
&DSN = ' ' /* INITIALIZE DATA SET NAME FIELD */
VPUT (ZDSCHK,ZDSCKO,DSN) SHARED /* PLACE IN SHARED POOL FOR ISRJB2 */
VPUT (ZBPACK,ZBPKED) PROFILE
&ZSEL = TRANS (&ZCMD
                C,C
                CAN,C
                CANCEL,C
                *, '*')

```

Figure 63. Batch JCL generation panel definition (ISRJPB) (Part 3 of 4)

```

if (&ZSEL = 'C') goto ENDD
&ZCMDWRK = &Z
IF (&ZCMD ^= &Z)
  &ZCMDWRK = TRUNC(&ZCMD, '.')
  &ZTRAIL=.TRAIL
  IF (&ZCMDWRK = &Z)
    .MSG = ISRU000
&ZSEL = TRANS (TRUNC (&ZCMD, '.'))
  1,'PGM(ISRJB2) PARM((ISRJP01) 1) NEWPOOL'
  2,'PGM(ISRJB2) PARM((ISRJP02) 2) NEWPOOL'
  3,'PGM(ISRJB2) PARM((ISRJP03) 3) NEWPOOL'
  5,'PGM(ISRJB2) PARM((ISRJP05) 5) NEWPOOL'
  6,'PGM(ISRJB2) PARM((ISRJP06) 6) NEWPOOL'
  7,'PGM(ISRJB2) PARM((ISRJP07) 7) NEWPOOL'
  10,'PGM(ISRJB2) PARM((ISRJP10) 10) NEWPOOL'
  12,'PGM(ISRJB2) PARM((ISRJP12) 12) NEWPOOL'
  13,'PGM(ISRJB2) PARM((EDCJP13,ISRJP13) 13) NEWPOOL'
  14,'PGM(ISRJB2) PARM((FANJP14,ISRJP14) 14) NEWPOOL'
  15,'PGM(ISRALTDI) PARM(EVGJP15,,ISRJP15,*) NOCHECK'
  16,'PGM(ISRJB2) PARM((EDCJP16,ISRJP16) 16) NEWPOOL'
  18,'CMD(ISPDTLC (PANEL SUBMIT RETURN)) MODE(FSCR)'
  19,'PGM(ISRALTDI) PARM(CBC3PE5A,+,ISRJP19,*,ISRJP19A) NOCHECK'
  ' ' ' '
  *,'?')
ENDD:
)PNTS
FIELD(ZPS01001) VAR(ZCMD) VAL(1)
FIELD(ZPS01002) VAR(ZCMD) VAL(7)
FIELD(ZPS01003) VAR(ZCMD) VAL(15)
FIELD(ZPS01004) VAR(ZCMD) VAL(2)
FIELD(ZPS01005) VAR(ZCMD) VAL(10)
FIELD(ZPS01006) VAR(ZCMD) VAL(16)
FIELD(ZPS01007) VAR(ZCMD) VAL(3)
FIELD(ZPS01008) VAR(ZCMD) VAL(12)
FIELD(ZPS01009) VAR(ZCMD) VAL(18)
FIELD(ZPS01010) VAR(ZCMD) VAL(5)
FIELD(ZPS01011) VAR(ZCMD) VAL(13)
FIELD(ZPS01012) VAR(ZCMD) VAL(19)
FIELD(ZPS01013) VAR(ZCMD) VAL(6)
FIELD(ZPS01014) VAR(ZCMD) VAL(14)
)END
/* 5647-A01 (C) COPYRIGHT IBM CORP 1980, 2000 */
/* ISPDTLC Release: 5.6. Level: PID */
/* z/OS 01.06.00. Created - Date: 19 Nov 2003, Time: 12:37 */

```

Figure 63. Batch JCL generation panel definition (ISRJPB) (Part 4 of 4)

The Batch option includes eight suboptions. Each Batch option has an associated panel, CLIST, and skeleton. For option 1, for example, the option panel name is ISRJP01, the CLIST name is ISRJC01, and the skeleton name is ISRJS01. Figure 64 on page 153 shows the Batch Assembler H panel definition.

**Note:** In Figure 64 on page 153, attribute characters have been replaced by blanks.

```

)PANEL KEYLIST(ISRSAB,ISR)
)ATTR DEFAULT(...) FORMAT(MIX) /* ISRJP01 - ENGLISH - 5.2 */
0B TYPE(AB)
04 TYPE(ABSL) GE(ON)
05 TYPE(PT)
09 TYPE(FP)
0A TYPE(NT)
0C TYPE(NT) SKIP(ON)
11 TYPE(SAC)
12 TYPE(CEF) PADC(USER)
19 TYPE(DT)
22 TYPE(WASL) SKIP(ON) GE(ON)
08 TYPE(CH)
26 TYPE(NEF) CAPS(ON) PADC(USER)
27 AREA(SCRL) EXTEND(ON)
28 TYPE(SAC) CSRGRP(99) RADIO(ON)
)ABC DESC('Menu') MNEM(1)
PDC DESC('Settings') UNAVAIL(ZPM1) MNEM(1) ACC(CTRL+S)
ACTION RUN(ISRRROUTE) PARM('SET')
PDC DESC('View') UNAVAIL(ZPM2) MNEM(1) ACC(CTRL+V)
ACTION RUN(ISRRROUTE) PARM('BR1')
PDC DESC('Edit') UNAVAIL(ZPM3) MNEM(1) ACC(CTRL+E)
ACTION RUN(ISRRROUTE) PARM('ED1')
PDC DESC('ISPF Command Shell') UNAVAIL(ZPM4) MNEM(6) ACC(CTRL+C)
ACTION RUN(ISRRROUTE) PARM('C1')
PDC DESC('Dialog Test...') UNAVAIL(ZPM5) MNEM(8) ACC(CTRL+T)
ACTION RUN(ISRRROUTE) PARM('DAL')
PDC DESC('Other IBM Products...') UNAVAIL(ZPM6) MNEM(1) ACC(CTRL+O)
ACTION RUN(ISRRROUTE) PARM('OIB')
PDC DESC('SCLM') UNAVAIL(ZPM7) MNEM(3) ACC(CTRL+L)
ACTION RUN(ISRRROUTE) PARM('SCL')
PDC DESC('ISPF Workplace') UNAVAIL(ZPM8) MNEM(6) ACC(CTRL+W)
ACTION RUN(ISRRROUTE) PARM('WRK')
PDC DESC('Status Area...') UNAVAIL(ZPMS) MNEM(8) ACC(CTRL+A)
ACTION RUN(ISRRROUTE) PARM('SAM')
PDC DESC('Exit') MNEM(2) PDSEP(ON) ACC(CTRL+X) ACTION RUN(EXIT)
)ABCINIT
.ZVARS=ISR@OPT
)ABC DESC('RefList') MNEM(1)
PDC DESC('Current Data Set List &ZDSCURT') MNEM(1) ACC(CTRL+ALT+P)
ACTION RUN(ISRRLIST) PARM('PL1')
PDC DESC('Current Library List &ZDSCURLT') MNEM(2) ACC(CTRL+SHIFT+P)
ACTION RUN(ISRRLIST) PARM('LL1')
PDC DESC('List of Personal Data Set Lists') MNEM(1) PDSEP(ON) ACC(CTRL+ALT+O)
ACTION RUN(ISRRLIST) PARM('PL2')
PDC DESC('List of Personal Library Lists') MNEM(2) ACC(CTRL+SHIFT+O)
ACTION RUN(ISRRLIST) PARM('LL2')

```

Figure 64. Batch Assembler H definition (ISRJP01) (Part 1 of 5)

```

)ABCINIT
.ZVARS=REFLIST
    VGET (ZCURTB ZCURLTB) PROFILE
    IF (&ZCURTB = &Z) &ZDSCURT = &Z
    ELSE &ZDSCURT= '(&ZCURTB)'
    IF (&ZCURLTB = &Z) &ZDSCURLT = &Z
    ELSE &ZDSCURLT= '(&ZCURLTB)'
)ABC DESC('Utilities') MNEM(1)
PDC DESC('Library') UNAVAIL(ZUT1) MNEM(1) ACC(ALT+1)
    ACTION RUN(ISRRROUTE) PARM('U1')
PDC DESC('Data set') UNAVAIL(ZUT2) MNEM(1) ACC(ALT+2)
    ACTION RUN(ISRRROUTE) PARM('U2')
PDC DESC('Move/Copy') UNAVAIL(ZUT3) MNEM(1) ACC(ALT+3)
    ACTION RUN(ISRRROUTE) PARM('U3')
PDC DESC('Data Set List') UNAVAIL(ZUT4) MNEM(2) ACC(ALT+4)
    ACTION RUN(ISRRROUTE) PARM('U4')
PDC DESC('Reset Statistics') UNAVAIL(ZUT5) MNEM(5) ACC(ALT+5)
    ACTION RUN(ISRRROUTE) PARM('U5')
PDC DESC('Hardcopy') UNAVAIL(ZUT6) MNEM(1) ACC(ALT+6)
    ACTION RUN(ISRRROUTE) PARM('U6')
PDC DESC('Download...') UNAVAIL(ZUTDT) MNEM(2) ACC(ALT+7)
    ACTION RUN(ISRRROUTE) PARM('UDT')
PDC DESC('Outlist') UNAVAIL(ZUT7) MNEM(2) ACC(ALT+8)
    ACTION RUN(ISRRROUTE) PARM('U8')
PDC DESC('Commands...') UNAVAIL(ZUT8) MNEM(1) ACC(ALT+9)
    ACTION RUN(ISRRROUTE) PARM('U9')
PDC DESC('Reserved') UNAVAIL(ZUT9) MNEM(6) ACTION RUN(ISRRROUTE) PARM('U10')
PDC DESC('Format') UNAVAIL(ZUT10) MNEM(1) ACC(ALT+F1)
    ACTION RUN(ISRRROUTE) PARM('U11')
PDC DESC('SuperC') UNAVAIL(ZUT11) MNEM(1) PDSEP(ON) ACC(CTRL+F2)
    ACTION RUN(ISRRROUTE) PARM('U12')
PDC DESC('SuperCE') UNAVAIL(ZUT12) MNEM(3) ACC(CTRL+F3)
    ACTION RUN(ISRRROUTE) PARM('U13')
PDC DESC('Search-For') UNAVAIL(ZUT13) MNEM(2) ACC(CTRL+F4)
    ACTION RUN(ISRRROUTE) PARM('U14')
PDC DESC('Search-ForE') UNAVAIL(ZUT14) MNEM(4) ACC(CTRL+F5)
    ACTION RUN(ISRRROUTE) PARM('U15')
)ABCINIT
.ZVARS=PDFUTIL
    &zut9 = '1'
)ABC DESC('Help') MNEM(1)
PDC DESC('Input Data Set') MNEM(1) ACTION RUN(TUTOR) PARM('ISR51001')
PDC DESC('SYSLIB Data Sets') MNEM(1) ACTION RUN(TUTOR) PARM('ISR51002')
PDC DESC('Object Data Set') MNEM(1) ACTION RUN(TUTOR) PARM('ISR51003')
PDC DESC('Listing') MNEM(1) ACTION RUN(TUTOR) PARM('ISR51004')
PDC DESC('Password protection') MNEM(1) ACTION RUN(TUTOR) PARM('ISR51007')
PDC DESC('Assembler selection') MNEM(1) ACTION RUN(TUTOR) PARM('ISR51008')
PDC DESC('Appendices') MNEM(4) ACTION RUN(TUTOR) PARM('ISR00004')
PDC DESC('Index') MNEM(2) ACTION RUN(TUTOR) PARM('ISR91000')
)ABCINIT
.ZVARS=JP1HELP

```

Figure 64. Batch Assembler H definition (ISRJP01) (Part 2 of 5)

```

)BODY CMD(ZCMD)
.. Menu. RefList. Utilities. Help.
-----
.
.                                     .Batch Assembler.
.Command ==>.Z
.SAREA39
)AREA SAREA39
.ISPF Library:.
. .Project . . .Z
. .Group . . .Z      . . .Z      . . .Z      . . .Z
. .Type . . .Z
. .Member . . .Z      ..(Blank or pattern for member selection list).
.
.Other Partitioned or Sequential Data Set:.
. .Data Set Name . .Z
. . .
.
.                                     .Assembler.
.List ID . . . .Z      ..(Blank for hardcopy). .Z..1..High Level Assembler.
.SYSOUT class . . .Z      ..(For hardcopy). .2..Assembler H
.
.Assembler options:.
...Term . . .Z      ..(TERM or NOTERM).
...Other . . .Z
.
.Additional input libraries:.
. . . ==>.Z
. . . ==>.Z
. . . ==>.Z
)INIT
.ZVARS = '(ZCMD PRJ1 LIB1 LIB2 LIB3 LIB4 TYP1 MEM DSN LID ZASMOPT BCLA +
          BHASMT BHASM BHAL1 BHAL2 BHAL3)'
.HELP = ISR51000
.ATTR(ZASMOPT)='CSRGRP(99) RADIO(ON)'
&ZUT6 = 1
&ZUT7 = 1
&ZUT11 = 1
&ZUT12 = 1
&ZUT13 = 1
&ZUT14 = 1
&ZMLCSR = ' '
&TYP1 = &ASMT
IF (&DSN ^= ' ')
  &MEM = ' '
  IF (.CURSOR = ' ') .CURSOR = DSN
  IF (&ZASMOPT ^= 1)
    IF (&ZASMOPT ^= 2)
      &ZASMOPT = 1
IF (&BASMT = ' ')
&BASMT = NOTERM
IF (&ZBTEMP = '')
  &ZBPACK = &ZBPACK
  VPUT (ZBPACK) SHARED
  &ZBTEMP = '0'
IF (&BHALEV = &Z)
  &BHALEV = 'HLASM'
VGET (BHAL1,BHAL2,BHAL3) PROFILE
)REINIT
REFRESH(PRJ1 LIB1 LIB2 LIB3 LIB4 TYP1 MEM DSN BHAL1 BHAL2 BHAL3)

```

Figure 64. Batch Assembler H definition (ISRJP01) (Part 3 of 5)

```

IF (&ZNXTMSG='ISRT') .CSRPOS = &ZCSRVP          /* AUTOTYPE */
                  .CURSOR = &ZCSRVP          /* AUTOTYPE */
ELSE &ZXZX = &Z                                /* AUTOTYPE */
&ZUT6 = 1
&ZUT7 = 1
&ZUT11 = 1
&ZUT12 = 1
&ZUT13 = 1
&ZUT14 = 1
IF (&ZMLCSR ^= ' ')                            /* @M1A*/
    .CURSOR = &ZMLCSR                          /* @M1A*/
IF (.MSG = ISRDS003)                            /* @M1A*/
    REFRESH (PRJ1,LIB1,LIB2,LIB3,LIB4,TYP1,MEM,DSN)
)PROC
&ZCSRVP = .CURSOR                              /* AUTOTYPE */
&ZCSRVP = .CSRPOS                              /* AUTOTYPE */
&ZODSNLN = 0                                  /* AUTOTYPE */
&ZODSNMB = &Z                                 /* AUTOTYPE */
&ZNAMES='ZCSRVP ZCSRVP PRJ1 LIB1 LIB2 LIB3 LIB4 TYP1 MEM '
&ZNAMES='&ZNAMES *.&ZODSNLN&ZODSNMB ZCMD'
IF (.CURSOR = DSN, BHAL1, BHAL2, BHAL3)
    &ZODSNLN = 56
    &ZODSNMB = &Z
    IF (.CURSOR = BHAL1) &ZODSNLN = 65
    IF (.CURSOR = BHAL1) &ZODSNMB = '%'
    IF (.CURSOR = BHAL2) &ZODSNLN = 65
    IF (.CURSOR = BHAL2) &ZODSNMB = '%'
    IF (.CURSOR = BHAL3) &ZODSNLN = 65
    IF (.CURSOR = BHAL3) &ZODSNMB = '%'
    &ZNAMES='ZCSRVP ZCSRVP * * * * * &ZCSRVP&ZODSNLN&ZODSNMB ZCMD'
PANEXIT((ZNAMES),LOAD,ISRAUTOT)                /* AUTOTYPE */
IF (&ZNXTMSG='ISRT') EXIT                      /* AUTOTYPE */
VER(&ZASMOPT,NONBLANK)
VER(&ZASMOPT RANGE,1,2)
&ZUT6 = 0
&ZUT7 = 0
&ZUT11 = 0
&ZUT12 = 0
&ZUT13 = 0
&ZUT14 = 0
VGET (ZRDSN) SHARED                            /* REFERENCE LIST CODE */
IF (&ZRDSN ^= ' ')                            /* IF DATA SET SELECTED */
    &DSN = &ZRDSN                             /* PUT DSN VARIABLE */
    &ZRDSN = ' '                               /* INTO PANEL */
    &ZRVOL = ' '                              /* INTO PANEL */
VPUT (ZRDSN ZRVOL) SHARED                     /* */
    .CURSOR = DSN
    .MSG = ISRDS003                            /* MSG PENDING */
VGET (DSALSEL) SHARED                         /* */
IF (&DSALSEL ^= ' ')                         /* IF LIBRARY SELECTED */
    VGET (DSA1,DSA2,DSA3,DSA4,DSA5,DSA6,DSA7) SHARED
    &PRJ1 = &DSA1                             /* PUT LIBRARY VARIABLES */
    &LIB1 = &DSA2                             /* INTO PANEL */
    &LIB2 = &DSA3                             /* . */

```

Figure 64. Batch Assembler H definition (ISRJP01) (Part 4 of 5)

```

&LIB3 = &DSA4                /*      .      */
&LIB4 = &DSA5                /*      .      */
&TYP1 = &DSA6                /*      .      */
&MEM = &DSA7                 /*      .      */
&DSN = ' '                   /* BLANK OUT DSN */
&DSALSEL = ' '               /* CLEAR LIBRARY SELECTION */
VPUT (DSALSEL) SHARED        /*
.CURSOR = MEM
.MSG = ISRDS003              /* MSG PENDING      */
/*
IF (&ZCMD = ' ') .MSG = ISPZ001 /* INVALID COMMAND */
VER (&LID,NAME)              /* LIST ID MUST BE VALID NAME */
IF (&ZDSCHK = 'N' )          /* IF NO SPF CHECK OF DATASET */
    IF (&DSN = ' ' )          /* AND SPF LIBRARY SPECIFIED */
        VER (&PRJ1,NB,NAME)    /* REQUIRED FIELD    */
        VER (&LIB1,NB,NAME)    /* REQUIRED FIELD    */
        VER (&TYP1,NB,NAME)    /* REQUIRED FIELD    */
    IF (&DSN = ' ' )          /* @M1A*/
        VER(&DSN DSNAMEPQ)
ELSE
    IF (&DSN = ' ' )
        VER(&DSN DSNAMEFEM)
IF (&ZASMOPT = 1)
    &BHALEV = 'HLASM'
IF (&ZASMOPT = 2)
    &BHALEV = 'HASM'
VER (&BHALEV,NB,LIST,HASM,HLASM) /* Assembler level OW10516*/
VER (&BHASMT,NB,LIST,TERM,NOTERM) /* TERM OR NOTERM REQUIRED */
IF (&LID = ' ' )
    VER (&BCLA,NONBLANK)      /* IF HARDCOPY DESIRED SPECIFY*/
    IF (&BHAL1 = ' ' )        /* VALID LOCAL SYSOUT CLASS */
        VER (&BHAL1,DSNAMEPQ) /* IF LIB #1 SPECIFIED @M2A*/
    IF (&BHAL2 = ' ' )        /* VERIFY ADDITIONAL LIB1@DSNQ*/
        VER (&BHAL2,DSNAMEPQ) /* IF LIB #2 SPECIFIED @M2A*/
    IF (&BHAL3 = ' ' )        /* VERIFY ADDITIONAL LIB2@DSNQ*/
        VER (&BHAL3,DSNAMEPQ) /* IF LIB #2 SPECIFIED @L2A*/
&ASMT = &TYP1                /* VERIFY ADDITIONAL LIB2@DSNQ*/
&ZSEL = 'CMD(%ISRJC01)'      /* SAVE ASSEMBLER TYPE */
IF (&LID = ' ' )             /* EXECUTE ASSEMBLE CLIST */
    IF (&BCLA = ' ' )        /* IF &BCLA NOT REQD */
        &BCLA = *           /* AND NOT SET */
/* DEFAULT TO MSGCLASS */
/* Begin @L3A */
&ZSYSDS1 = &BHAL1            /* Fill fields for ISRJFSYS */
&ZSYSDS2 = &BHAL2            /* to use as input. */
&ZSYSDS3 = &BHAL3            /* ZSYSDS? is a qualified */
&ZSYSCUR1 = 'BHAL1'          /* dataset. */
&ZSYSCUR2 = 'BHAL2'          /* ZSYSCUR? is were the cursor*/
&ZSYSCUR3 = 'BHAL3'          /* is placed on a error. */
VPUT (ZSYSDS1 ZSYSDS2 ZSYSDS3 ZSYSCUR1 ZSYSCUR2 ZSYSCUR3) SHARED /*
/* End @L3A */
VPUT (PRJ1,LIB1,LIB2,LIB3,LIB4,ASMT,BCLA,BHASMT,BHASM) PROFILE
VPUT (BHAL1,BHAL2,BHAL3,BHALEV,ZASMOPT) PROFILE
VPUT (PRJ1,LIB1,LIB2,LIB3,LIB4,ASMT,BCLA,BHASMT,BHASM) SHARED
VPUT (DSN,LID,BHAL1,BHAL2,BHAL3,BHALEV) SHARED /* OW10516*/
)END
/* 5694-A01 (C) COPYRIGHT IBM CORP 1980, 2001 */
/* ISPDTLC Release: 5.2. Level: PID */
/* z/OS 01.02.00. Created - Date: 10 Mar 2001, Time: 14:25 */

```

Figure 64. Batch Assembler H definition (ISRJP01) (Part 5 of 5)

The names of the ISPF-supplied panels, CLISTs, and skeletons for the Batch processing option are shown in the Table 27 on page 158.

Table 27. ISPF-supplied panels, CLISTs, and skeletons for Batch option

Option	Description	Panel ID	CLIST ID	SKEL ID
-	BATCH SELECTION MENU	ISRJPA	-	ISRJSJC
--	BATCH TERMINATION MENU	ISRJPB	--	--
1	ASSEMBLER	ISRJP01	ISRJC01	ISRJS01
2	VS COBOL II	ISRJP02	ISRJC02	ISRJS02
3	FORTRAN COMPILE	ISRJP03	ISRJC03	ISRJS03
5	PLI OPTIMIZER COMPILE	ISRJP05	ISRJC05	ISRJS05
6	VS PASCAL COMPILE	ISRJP06	ISRJC06	ISRJS06
7	BINDER/LINK EDIT	ISRJP07	ISRJC07	ISRJS07
10	VS COBOL II DEBUG	ISRJP10	ISRJC10	ISRJS10
12	MEMBER PARTS LIST	ISRJP12	ISRJC12	ISRJS12
13	C/370 COMPILE	--	--	--
14	REXX/370 COMPILE	--	--	--
15	ADA/370 COMPILE	--	--	--
16	AD/CYCLE C/370 COMPILE	--	--	--
17	AD/CYCLE C/370 COBOL/370	--	--	--
18	ISPD TLC	ISPCP01	--	ISPD TLB
19	OS/390 C C++	--	--	--

The overall flow of control for Batch processing is shown in Figure 65.

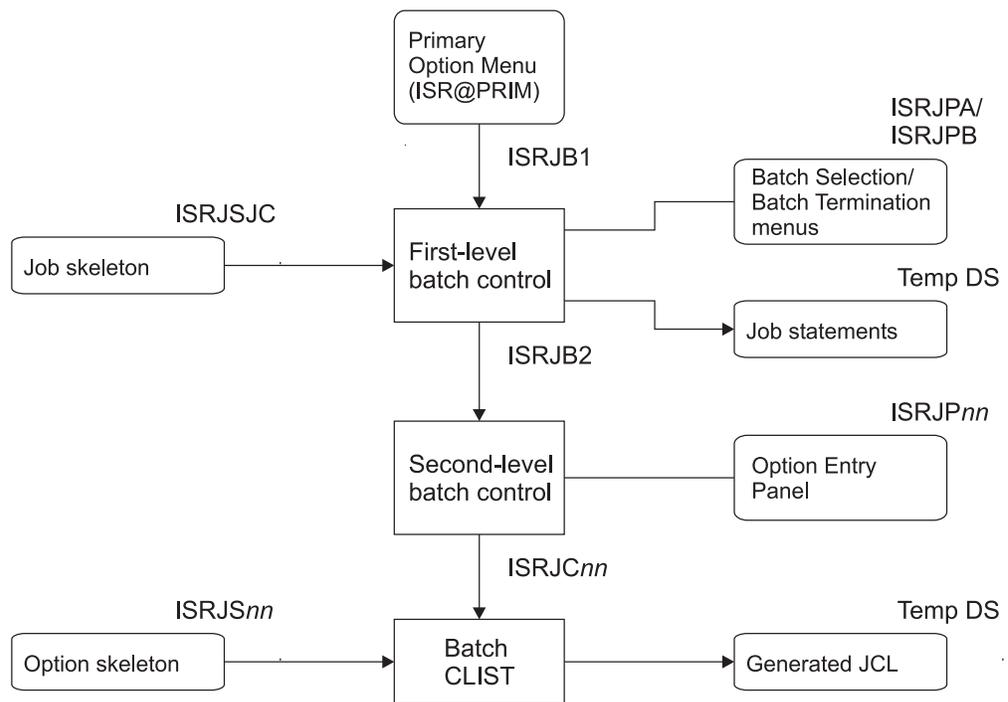


Figure 65. Batch processing flow

Two PDF programs control Batch processing: ISRJB1 and ISRJB2. Program ISRJB1 receives control directly from the ISPF Primary Option Menu, ISR@PRIM, through specification of these selection keywords when you select primary option 5:

```
'PGM(ISRJB1) PARM(ISRJPA) NOCHECK'
```

The parameter specifies the name of the Batch Selection panel. The NOCHECK keyword allows ISRJB1 to receive control if you specify option 5.n, where “n” is a batch option.

ISRJB1 either displays the Batch Selection panel (if you entered option 5 on the ISPF Primary Option Menu) or processes the Batch Selection panel in non-display mode (if you entered 5.n). ISRJB1 makes this determination by examining the ZTRAIL variable, which is set from the ISPF Primary Option Menu. ZTRAIL contains either the option number, or a blank if no option was specified.

Regardless of whether the Batch Selection panel is actually displayed, ISRJB1 invokes file tailoring services to write the four job statements from the Batch Selection panel to a temporary data set. Skeleton ISRJSJC is used to generate the job statement output. ISRJB1 then invokes program ISRJB2 (through the SELECT service) and passes a parameter containing the name of the first (or only) option panel to be displayed.

The selection keywords used to invoke ISRJB2 actually come from the Batch Selection panel, ISRJPA, where they are stored in variable ZSEL. ISRJB1 uses the DISPLAY service, rather than the SELECT service, to display the Batch Selection panel, and subsequently passes ZSEL as input to the SELECT service when it is ready to invoke ISRJB2.

Program ISRJB2 is analogous to the Foreground control program ISRFPR. See “Foreground processing panels and CLISTs” on page 130. ISRJB2 displays and processes the option panel and invokes the corresponding CLIST. The CLIST, in turn, uses file tailoring services to generate JCL statements for the particular option. All Batch JCL is accumulated in the same temporary data set that contains the four job statements.

Eventually, control returns to ISRJB1, the first batch program. ISRJB1 then either displays the Batch Selection panel (ISRJPB) with Job Step Generated, or processes it in non-display mode if display of the Batch Selection panel was bypassed. The Job Step Generated panel is similar in appearance to the Batch Selection Panel, except that the job statements can no longer be modified. From the Job Step Generated panel, you can select additional options, causing additional job steps to be generated.

Finally, ISRJB1 closes the temporary data set in which all the batch JCL was accumulated and submits it using the TSO SUBMIT command. The submission will be bypassed if either of these conditions occurs:

- No JCL was generated by the options.
- You entered CANCEL on the Batch Selection panel with Job Step Generated.

### **Variables that control batch processing**

The Batch Selection panel, ISRJPA, includes two variables that control Batch processing. The variable RTNPNL contains the name of the panel that corresponds to the Batch Selection panel (for ISRJPA the corresponding panel is ISRJPB). The variable ZDCKO is associated with the Source Data Online field on the Batch Selection panel. If you enter YES in that field, the Batch option provides error

checking and verification of the input data sets. If NO is entered, no data set verification is performed. The NO option allows you to submit a job to be run at a later time if the data sets are not online.

If the Source Data Online field, variable ZDSCKO, is set to YES, the dialog variables required for batch option panels are the same as those for foreground processing. That is, variables PRJ1, LIB1, LIB2, LIB3, LIB4, TYP1, MEM, DSN, and ZSEL must be defined in either the )INIT, )BODY, or )PROC section of the panel displayed by ISRJB2.

If the Source Data Online, variable ZDSCKO, is set to NO, these variables are not required. However, if a data set name is not supplied to ISRJB2 in these variables, the output variables ZDSQMEM, ZDSQ, ZDSQ2, ZDSQ3, ZDSQ4, ZDS, ZDS2, ZDS3, ZDS4, ZMEM, and ZORG are blank when control is passed to the CLIST referenced in variable ZSEL in the option panel. See “Required option entry panel variables” on page 141.

### Generated JCL

The job control language (JCL) statements generated by the Batch processing option are accumulated in a temporary data set named *userid.SPFTEMPn.CNTL* or *prefix.userid.SPFTEMPn.CNTL*, where *n* is the screen number. This data set contains the job cards generated by ISRJB1 and job steps generated by the option CLISTS. In all cases, file tailoring services are used to generate the JCL. If VIO is used, the generated JCL is in a VIO data set allocated to DDname ISPCTL1.

ISRJB1 generates up to four job statement lines from information that you supply in the Batch Selection panel, ISRJPA. A job statement line is not generated if the corresponding field on the panel is blank. An attempt is made to provide unique jobnames by using this algorithm:

1. The four lines entered in the selection panel are scanned for the first ‘//’ card.
2. If the string following the ‘//’ is equal to the TSO user ID, and if the user ID is followed by an alphabetic character or a numeric character, that character is automatically incremented each time a job is submitted.
3. If the string following the ‘//’ is not equal to the TSO user ID, the job name will not be automatically incremented each time a job is submitted. ISRJB1 increments and changes jobnames only if they begin with the user ID.

If you bypass the Batch Selection panel (for example, by selecting option 5.n from the ISPF Primary Option Menu), the job statement lines that would have been displayed on the selection panel are used as if you had not modified them.

The file skeleton associated with each option controls the JCL generated for that option. See “Batch (Option 5)” in the *z/OS ISPF User’s Guide Vol II* for a general description of skeleton formats. Each variable in the skeleton is replaced by its current value (the contents of the corresponding panel input field, as entered by you, or a previously entered value). Following variable substitution, each record in the skeleton is written to the temporary data set, which is eventually submitted to the job stream by the TSO SUBMIT command. Skeletons must be coded so that the maximum length of a record after substitution does not exceed 71 characters; otherwise, invalid JCL might be generated.

### Variables that control option panel redisplay

After an option has been processed, the Batch Selection panel with Job Step Generated is normally displayed, thereby allowing you to add another job step to the one just generated. It is possible, however, to link options together

automatically by using the shared variable ZNEXTPN. The program ISRJB2 sets ZNEXTPN to blank. After the completion of a option CLIST, ISRJB2 references the variable again to determine the next panel to display. If you wish, for example, to display the linkage edit panel (ISRJP07) after processing the FORTRAN panel (ISRJP03), set ZNEXTPN to ISRJP07 either in the )PROC section of the FORTRAN panel or in the FORTRAN CLIST (ISRJC03), and place the variable in the shared pool. You can extend this type of panel linkage to any length you want.

### Variable used by Batch CLISTs

In the distributed Batch option, all option panels select a CLIST that uses file tailoring services to generate the JCL job steps. (Only a CLIST or program can be selected from the option panel.) To obtain variables set from the panel and by program ISRJB2, the CLIST references the shared variable pool. The CLIST performs basic error checking on some of the variables associated with the panel fields. After generating the JCL, the CLIST returns to ISRJB2. If, because of some error condition, the CLIST does not generate a job step, a nonzero return code is returned to ISRJB2.

### Batch skeletons

The skeleton library contains the JCL and file tailoring control statements that are used to generate the JCL for a particular Batch option. A primary skeleton is associated with each option. This skeleton embeds other skeletons based on input data set type. For option 1, for example, skeleton ISRJS01 is used. If the data set was defined in the ISPF library section of the panel, ISRJS01 embeds skeleton ISRJS01I for appropriate JCL. If an “other” partitioned data set were specified, skeleton ISRJS01P would be embedded. If an “other” sequential data set were specified, skeleton ISRJS01S would be embedded.

### Batch modifications

Most of the Batch processing logic is in the option panels, CLISTs, and skeletons to make modifications as easy as possible. Before modifying existing options or adding new ones, you should study the distributed panels, CLISTs, and skeletons to understand the relationships among them.

ISPF does not attempt to diagnose all possible types of skeleton coding errors. Some types of errors are detected, and the appropriate error messages are displayed, but in many cases coding errors must be debugged by inspection of the generated JCL. To perform this inspection, enter the CANCEL command when the Batch Selection panel with Job Step Generated is displayed, and then browse the temporary data set that contains the generated JCL. (The temporary data set is closed, but not freed or erased, when the CANCEL command is specified.)

**Note:** This procedure cannot be used if the temporary CNTL data sets have been allocated to VIO because they have MVS-generated data set names.

**Steps to add a new Batch primary option to PDF:** The steps required to add a new PDF primary option that uses the Batch processing mechanism are listed here.

1. Add the new option (for example, '99') to the ISPF Primary Option Menu, panel name ISR@PRIM. In the translated value for option 99, use the PGM keyword to specify that program ISRJB1 is to receive control, and use the PARM keyword to pass the name of the Batch Selection panel. Specify the NOCHECK option to allow ISRJB1 to select subsequent panels. Set the ZTRAIL variable to the value remaining from the truncation function. (ZTRAIL is already set in panel ISR@PRIM, but you must set it yourself if you invoke ISRJB1 from some other selection panel.) For example:

```

)PROC
&ZSEL; = TRANS( TRUNC(&ZCMD, '.' )
. . .

99, 'PGM(ISRJB1) PARM(ZNEW99A) NOCHECK '
)
&ZTRAIL; = .TRAIL

```

2. Add a new Batch Selection panel (ZNEW99A in this example) to the panel library. Use panel ISRJPA as a model. For each option on the panel, use the PGM keyword to specify that program ISRJB2 is to receive control, and use the PARM keyword to pass the name of the option panel.
3. In the panel definition for ZNEW99A, set the variable RTNPNL to ZNEW99B (as an example) to specify the corresponding Batch Selection panel with Job Step Generated. Add new panel ZNEW99B to the panel library, using panel ISRJPB as a model.
4. Develop new option panels, CLISTs, and skeletons as described in "Steps to add a new Batch option to the PDF Batch Selection Panel."

**Steps to add a new Batch option to the PDF Batch Selection Panel:** The steps required to add a new option to the Batch Selection Panel and the panel showing Job Step Generated are listed here.

1. Add the new option (for example, '99') to panels (ISRJPA and ISRJPB). Use the PGM keyword to specify that program ISRJB2 is to receive control and use the PARM keyword to pass the name of the new option panel. For example:

```

)PROC
&ZSEL = TRANS(&ZCMD
. . .

99, 'PGM(ISRJB2) PARM(ZNEW99 99) '

```

2. Add new option panel ZNEW99 to the panel library. Use one of the distributed option panels (for example, ISRJP01) as a model.
3. Develop a corresponding CLIST (referenced from panel ZNEW99 by the ZSEL variable) and skeleton (referenced from the CLIST by the file tailoring FTINCL service). Use the distributed CLISTs and skeletons (for example, ISRJC01 and ISRJS01) as models. Add the CLIST to a library accessible to ddname SYSPROC, and add the skeleton to the skeleton library.

You can also develop a new option that displays a lower-level selection panel, from which user selections invoke the Batch CLISTs.

For example:

```

)PROC
&ZSEL = TRANS (&ZCMD
1, 'PGM(ISRJB2) PARM(ISRJP01) '
2, 'PGM(ISRJB2) PARM(ISRJP02) '
. . .

9, 'CMD(ZNEWCMD &ZCMD) '

```

In this example, the CLIST ZNEWCMD has been selected, and the parameter ZCMD (the option entered on the panel) has been passed to the CLIST. ZNEWCMD can then use the SELECT service to display a lower-level selection panel similar to panel ISRJPA, but without job statement information. When the options have been processed, the CLIST should end and return control to ISRJB1 to close the temporary data set and submit the job to TSO. If JCL was generated the invoked dialog (CLIST or PANEL) must end with a return code of 0. If no JCL was

generated the dialog must either end with a return code greater than 0 or set dialog variable ZADARC to a Y in either the ISPF SHARED or PROFILE pool.

**Note:** If the Source Data Online field, variable ZDSCKO, is set to YES, the dialog variables required for batch option panels are the same as those for foreground processing. That is, variables PRJ1, LIB1, LIB2, LIB3, LIB4, TYP1, MEM, DSN, and ZSEL must be defined in either the )INIT, )BODY, or )PROC section of the panel displayed by ISRJB2.

If the Source Data Online, variable ZDSCKO, is set to NO, these variables are not required. However, if a data set name is not supplied to ISRJB2 in these variables, the output variables ZDSQMEM, ZDSQ, ZDSQ2, ZDSQ3, ZDSQ4, ZDS, ZDS2, ZDS3, ZDS4, ZMEM, and ZORG are blank when control is passed to the CLIST referenced in variable ZSEL in the option panel.

### Batch processing options considerations

The SISPLPA data set includes the ISRSCAN load module. It is not executed under TSO but is invoked from batch jobs submitted through the ISPF PDF batch processing option. The ISRSUPC and ISRLEMIX load modules are executed in both foreground and batch.

To submit jobs through the ISPF batch processing option, the load modules ISRSCAN, ISRSUPC, and ISRLEMIX must be available. If they are not in your LPA or link library, insert a STEPLIB DD statement for the library that contains them in each skeleton member ISRJSxxx and ISRSBJCL in the ISP.SISPLIB data set. Insert the statement following the EXEC PGM=ISRSCAN, EXEC PGM=ISRLEMIX, or EXEC PGM=ISRSUPC statement, whichever is in the skeleton (see the comments in each of these members).

## ISRSCAN and ISRLEMIX programs

The ISPF-supplied Batch skeletons each generate one or more job steps. The first step executes either the ISRSCAN program or the ISRLEMIX program. Both programs find the input member and copy it from the library hierarchy (up to four partitioned data sets) to a temporary data set for input to the processing program executed in the second step. The difference between them is that ISRSCAN copies only one member to the output data set, while ISRLEMIX copies the primary member and expands any included members as part of the output data set. All members copied by ISRLEMIX are unpacked in the output data set and ISRSCAN does not unpack the input while ISRLEMIX does. The input to ISRSCAN must not be packed. ISRLEMIX also creates a member parts list (see ISPF Options 4.12 and 5.12).

If your input is a sequential data set, ISRLEMIX only copies the data to the output data set. **It unpacks the input but does not unpack any included members.**

The first step is essential for the operation of library hierarchies because the ISPF-supplied processing programs typically accept primary input only from fully qualified data sets (that is, from either a sequential data set or a single member of a single partitioned data set, not from concatenations of partitioned data sets).

**Using ISRSCAN:** The Batch job steps for using ISRSCAN are as follows:

```
//*ISRSCAN step
//SCANSTEP EXEC PGM=ISRSCAN,PARM='member name'
//STEPLIB DD DSN= (Library name if ISRSCAN is not in
                your system library)
```

```
//IN      DD ... Input data set in which the source
          member is found. Can be a concatenation of up to 8 data sets.
//OUT     DD ... Sequential output data set.
/*
```

**Return Codes:** ISRSCAN sets one of these return codes in register 15:

```
0      Normal completion
8      One of these:
      • DDNAME OUT not found.
      • Error retrieving data set information for OUT data set.
      • OUT data set is a PDS without a member specified.
12     Member not found
16     Unable to open input DCB
20     I/O error on input data set
24     Unable to open output DCB
28     I/O error on output data set
```

**Using ISRLEMX:** ISRLEMX Batch job steps are shown here.

```
/*ISRLEMX step
//LEMXSTEP EXEC PGM=ISRLEMX,PARM='parm1,parm2,...,parm15'
//STEPLIB DD DSN= (Library name if ISRLEMX is not in your
                system library)
//ISRLCODE DD ... Input data set in which the source member(s)
                are found. Can be a concatenation of up to 8 data sets.
//ISRLEXPDD DD ... Sequential output data set for expansion.
//ISRLXREF DD ... Sequential output data set for mem prts lst.
//ISRLMSG DD SYSOUT=A /* program error messages print here */
```

- or -

```
//ISRLMSG DD DSN=&ZPREFIX;.&LID;.LMSG,UNIT=SYSDA,
            SPACE=(TRK,(1,1)),DISP=(MOD,CATLG),
            DCB=(RECFM=FBA,LRECL=133,BLKSIZE=3059)
/*
```

**Return Codes:** ISRLEMX sets one of these return codes in register 15:

```
0      Normal completion
n      Parameter n is 1-15 (too long)
16     Too many parameters
17     Too few parameters
20     Severe error in expand module—an error is printed in the ISRLMSG data
      set.
```

**Notes:**

1. DDNAME ISRLEXPDD is needed if parameter 5 is either 'E' or blank. DDNAME ISRLXREF is needed if parameter 5 is 'L'. See this figure for additional information.
2. ISRLEMX reads data that is presented in a BSAM compatible form. If the SUBSYS parameter (or any parameter that will cause the UCB pointer in the TIOT to be 0) is used, PDF cannot verify that the input is on DASD. It is the responsibility of the user to ensure ISRLEMX will see the data in the correct form. If the data is not presented in a BSAM compatible form, results will be unpredictable.

The ISRLEMX parameter string contains up to 15 parameters, each separated from the next by commas. The parameters are:

Parameter	Length	Description
1	CHAR(3)	Language type of the input member to be processed: ASM - Assembler COB - COBOL FOR - FORTRAN PAS - Pascal PLI - PL/I SCR - Script
2	CHAR(8)	Member name of member to be expanded or of first member to be processed to create a member parts list.
3	CHAR(1)	'B' - The request is being run in batch 'F' - The request is being run in foreground
4	CHAR(1)	'Y' - Allocate a temporary sort data set if needed 'N' - Do not allocate a sort data set
5	CHAR(1)	'E' - Expand and unpack the specified member and all included members into one sequential data set 'L' - Create a member parts list starting with the specified member blank - Copy the specified input member to a temporary data set, unpacking the member but not expanding.
6	CHAR(1)	Number of concatenated input libraries that should be scanned to find the specified input number value can be 1-9
7	CHAR(20)	User trigger, a character string of maximum length 20 to be processed as an INCLUDE, COPY or IMBED statement when found in the member being processed. Enter a comma to skip this parameter if no user trigger is being used.
8	CHAR(2)	User trigger start column, specifies which column the user trigger listed here will start in the member being processed. Enter a comma to skip this parameter if no user trigger is being used.
9	CHAR(3)	Indicates the National language in use: ENU - English DEU - German (Deutsch) KAN - Japanese (Kanji) DES - Swiss German  This is used to build the name of the literal load module that is loaded by member expansion. If the language is not English and the load fails, the English table is loaded.
10	CHAR(1)	Position of the month value in the date string; for example, <i>yy/mm/dd</i> (default '4')
11	CHAR(1)	Position of the day value in the date string; for example, <i>yy/mm/dd</i> (default '7')
12	CHAR(1)	Position of the year value in the date string; for example, <i>yy/mm/dd</i> (default '1')
13	CHAR(1)	Delimiter to use in the date string; for example, <i>yy/mm/dd</i> (default '/')
14	CHAR(8)	Unit name to be used for all temporary data sets used by ISRLEMX. The unit name must be specified.

Parameter	Length	Description
15	CHAR(4)	The number of blocks used when allocating the temporary sort data sets. These are allocated to ddnames ISRKLWKnn, where nn is 1 through 4. The default for this parameter is 0100. It can be increased for very large or complex expansions.

**Note:** You can specify the date notation in the national language format (for example, yy/mm/dd, mm/dd/yy, dd/mm/yy). Therefore, you must specify the index for each portion. Use any valid character to delimit the date string (for example, yy/mm/dd, dd.mm.yy). This delimiter is required.

### Adding user-defined triggers

During installation time, you can add a user-defined trigger to the member expansion or member parts list functions, or to the foreground or batch options of PDF. The user defined trigger is interpreted as though it were an INCLUDE, COPY, or .im keyword.

**Adding a user-defined trigger to the member expansion function:** The user can define the member expansion function trigger by using up to 20 characters. The trigger must start in the user-defined start column and be followed by at least one blank. The next nonblank string is interpreted as the included member name. For more information about member expansion and triggers, see *z/OS ISPF User's Guide Vol II*.

**Adding a user-defined trigger to the Foreground options:** To define a user trigger for the PDF Foreground options, change the ISRLEMX SELECT statement in the CLIST for the appropriate option.

*Change from:*

```
ISPEXEC SELECT PGM(ISRLEMX) +
  PARM('XXX,&ZMEM,F,N,E,4, ,00,&ZFPRLANG,&ZFPRMMIX,+
  &ZFPRDDIX,&ZFPRYYIX,&ZFPRDLIM,&Z4UNIT')
```

*To:*

```
ISPEXEC SELECT PGM(ISRLEMX) +
  PARM('XXX,&ZMEM,F,N,E,4,trigger,nn,&ZFPRLANG,&ZFPRMMIX,+
  &ZFPRMMIX,&ZFPRDDIX,&ZFPRYYIX,&ZFPRDLIM,&Z4UNIT')
```

where *nn* is the trigger start column.

The CLIST names are as follows:

```
OPTION 4.1    ISRFC01
OPTION 4.2    ISRFC02
OPTION 4.3    ISRFC03
OPTION 4.5    ISRFC05
OPTION 4.6    ISRFC06
OPTION 4.9    ISRFC09
```

**Adding a user-defined trigger to the Batch options:** To define a user trigger for the PDF Batch options, change the ISRLEMX EXEC statement in the skeleton for the appropriate option.

*Change from:*

```
//EXPAND EXEC PGM=ISRLEMX,COND=(12,LE),
//  PARM=('XXX,&ZMEM,B,N,E,4, ,00,&ZJB2LANG,;
//  &ZJB2MMIX,&ZJB2DDIX,&ZJB2YYIX,&ZJB2DLIM,&Z5UNIT')
```

```
To:
//EXPAND EXEC PGM=ISRLEM,COND=(12,LE),
// PARM=('XXX,&ZMEM,B,N,E,4,trigger,nn,&ZJB2LANG,;
// &ZJB2MMIX,&ZJB2DDIX,&ZJB2YYIX,&ZJB2DLIM,&Z5UNIT')
```

where *nn* is the trigger start column.

The skeleton names are as follows:

```
OPTION 5.1   ISRJS01I, ISRJS01J, ISRJS01P, ISRJS01S
OPTION 5.2   ISRJS02I, ISRJS02J, ISRJS02P, ISRJS02S
OPTION 5.3   ISRJS03I, ISRJS03J, ISRJS03P, ISRJS03S
OPTION 5.5   ISRJS05I, ISRJS05J, ISRJS05P, ISRJS05S
OPTION 5.6   ISRJS06I, ISRJS06J, ISRJS06P, ISRJS06S
```

**Adding a user-defined trigger to the member parts list function:** The user can define the trigger for the member parts list function by using up to 20 characters. The trigger must start in the user-defined start column and be followed by at least one blank. The next nonblank string is interpreted as the included member name.

To define the trigger, change two SET statements in the CLISTs ISRFC12 (Foreground) and ISRJC12 (Batch), as follows:

*Change from:*

```
SET &UT = &STR(&Z)
SET &UTC = &STR(00)
```

*To:*

```
SET &UT = &STR(trigger)
SET &UTC = &STR(nn)
```

where *nn* is the trigger start column.

**Changing the size of the data set for the member parts list function:** The member parts list function allocates a temporary data set for collecting cross-reference data. The default size of this data set is 100 blocks each of primary and secondary storage. To change this size, change the SSPACE variable in the panels ISRF12 (foreground) and ISRJP12 (batch), as follows:

*Change from:*

```
)PROC
&SSPACE = '0100'
```

*To:*

```
)PROC
&SSPACE = 'nnnn'
```

where *nnnn* is the number of blocks to be allocated.

---

## Customizing Browse and Edit

Customizing Browse and Edit describes how to provide customized Browse and Edit panels, and how to enable users to browse data that is stored in a Unicode format.

## Providing customized Browse and Edit panels

Dialog developers can provide customized Browse or Edit data display panels for the dialogs they create. The name of the customized panel is passed as a parameter to the BROWSE or EDIT dialog service.

**Note:** Do not use the names ISREDDE, ISREDDE2, ISREDDE3, ISREDDE4, or ISREDDE5 as a panel name passed to the EDIT or VIEW services. When ISPF is using any of these panels, it can dynamically switch among any of these panels based on system configuration and edit profile settings.

Customized Browse panels must be patterned after the distributed panel ISRBROB (for Browse, as shown in Figure 66 on page 169). Customized Edit panels must be patterned after these panels:

ISREFR01 - Edit without action bars or extended highlighting (as shown in Figure 67 on page 170)

ISREFR02 - Edit with action bars and extended highlighting

ISREFR03 - Edit with action bars and no extended highlighting

ISREFR04 - Edit with extended highlighting but no action bars

The customized panel must include a Command field named ZCMD and a dynamic area named ZDATA. As shown in Figure 67 on page 170, this ZDATA dynamic area should be coded with an attribute that is defined with AREA(DYNAMIC). It is not acceptable to code this area with an attribute defined as AREA(SCRL). To enable extended highlighting, the customized edit panel must have a shadow variable called ZSHADOW and must have the attributes as defined in the attribute section of panel ISREFR04. Customized panels should meet the requirements for a movable Command line, so that the Command line can be displayed at the bottom of the screen at your request. See *z/OS ISPF Dialog Developer's Guide and Reference* for a discussion of these requirements.

Inclusion of a Scroll field is optional. The scroll fields in the distributed panels are named ZSCBR (for Browse) and ZSCED (for Edit). You can use these same names for scroll fields on customized panels. Inclusion of the (protected) variables that appear in the title line of the distributed panels is optional. If you want to display the volume of the data set, you can use variable ZDSVOL.

Because Edit and Browse translate non-displayable characters into blanks (for Edit) or a user-specified character (for Browse), hex codes that represent these characters cannot be placed in the data with the intent of using them as attribute characters. If you want to modify the Edit or Browse display, you must either use displayable characters, or redefine the existing Edit and Browse attributes. If you redefine an existing attribute, do not change the TYPE or FORMAT values of the attribute.

The HIDE EXCLUDED command uses reserved attribute characters to underscore the number field of the line preceding the excluded message line. As some customized Edit panels might already use these attribute characters, ISPF uses the dialog variable ZHIDEX to determine whether the attribute characters are available and correctly defined in the Edit panel. An example of setting ZHIDEX and providing the reserved attribute bytes 13, 16, 17, and 1D is shown in Figure 67 on page 170.

If the HIDE command is used in an initial macro, then the macro must set the ISPF dialog variable ZHIDEX to a value of ?Y? and the Edit panel in use must contain the required attribute character definitions as previously described.

Note, even when the HIDE EXCLUDED command is enabled using the ZHIDEX variable, the number field of the line preceding the excluded message is only underscored on ISPF-supported terminals with the extended highlighting feature available.

---

```

)ATTR DEFAULT( ) FORMAT(MIX)          /* ISRBROB - ENGLISH - 5.5 */
05 TYPE(PT)
09 TYPE(FP)
0A TYPE(NT)
13 TYPE(NEF) PADC(USER)
16 TYPE(VOI) PADC(USER)
26 AREA(DYNAMIC) EXTEND(ON) SCROLL(ON)
01 TYPE(DATAOUT) INTENS(LOW)
02 TYPE(DATAOUT)
0B TYPE(DATAOUT) FORMAT(DBCS) OUTLINE(L)
0C TYPE(DATAOUT) FORMAT(EBCDIC) OUTLINE(L)
0D TYPE(DATAOUT) FORMAT(&MIXED) OUTLINE(L)
10 TYPE(DATAOUT) INTENS(LOW) FORMAT(DBCS) OUTLINE(L)
11 TYPE(DATAOUT) INTENS(LOW) FORMAT(EBCDIC) OUTLINE(L)
12 TYPE(DATAOUT) INTENS(LOW) FORMAT(&MIXED) OUTLINE(L)
)BODY EXPAND(//) WIDTH(&ZWIDTH) CMD(ZCMD)
BROWSE Z/ /                               Line Z          Col Z
Command ==> Z/ /                          Scroll ==> Z
ZDATA/ /
/ /
/ /
)INIT
.ZVARS = '(ZTITLE ZLINES ZCOLUMNS ZCMD ZSCBR)'
.HELP = ISR1B000
&ZCMD = ' '
VGET (ZSCBR) PROFILE /* Fill Scroll Vars if */
IF (&ZSCBR = ' ') &ZSCBR = 'PAGE' /* Blank with Page */
IF (&ZMEMB = ' ') &ZTITLE = '&ZDSNT(&ZMEMB)&ZLEVEL ' /* 0Z91708 */
IF (&ZMEMB = ' ') &ZTITLE = '&ZDSNT&ZLEVEL '
&MIXED = MIX
IF (&ZPDMIX = N) &MIXED = EBCDIC
)REINIT
REFRESH(ZCMD,ZSCBR,ZDATA,ZLINES,ZCOLUMNS,ZTITLE)
)PROC
&ZCURSOR = .CURSOR
&ZCSROFF = .CSRPOS
VPUT (ZSCBR) PROFILE /* */
&ZLVLINE = LVLINE(ZDATA)
)END
/* 5647-A01 (C) COPYRIGHT IBM CORP 1995, 1999 */

```

---

Figure 66. Browse panel (ISRBROB)

```

)ATTR /* EDIT PANEL WITH NO ACTION BAR & NO HIGHLIGHTING */
  TYPE(INPUT) CAPS(OFF) INTENS(HIGH) FORMAT(&MIXED)
  | AREA(DYNAMIC) EXTEND(ON) SCROLL(ON) USERMOD(20)
  ! TYPE(OUTPUT) INTENS(HIGH) PAD(-)
01 TYPE(DATAOUT) INTENS(LOW)
02 TYPE(DATAOUT) INTENS(HIGH)
03 TYPE(DATAOUT) SKIP(ON) /* FOR TEXT ENTER CMD. FIELD */
04 TYPE(DATAIN) INTENS(LOW) CAPS(OFF) FORMAT(&MIXED)
05 TYPE(DATAIN) INTENS(HIGH) CAPS(OFF) FORMAT(&MIXED)
06 TYPE(DATAIN) INTENS(LOW) CAPS(IN) FORMAT(&MIXED)
07 TYPE(DATAIN) INTENS(HIGH) CAPS(IN) FORMAT(&MIXED)
08 TYPE(DATAIN) INTENS(LOW) FORMAT(DBCS) OUTLINE(L)
09 TYPE(DATAIN) INTENS(LOW) FORMAT(EBCDIC) OUTLINE(L)
0A TYPE(DATAIN) INTENS(LOW) FORMAT(&MIXED) OUTLINE(L)
0D TYPE(DATAIN) INTENS(LOW) CAPS(IN) FORMAT(&MIXED) COLOR(BLUE)
13 TYPE(DATAOUT) SKIP(ON) HILITE(USCORE)
16 TYPE(DATAIN) INTENS(LOW) CAPS(IN) HILITE(USCORE) FORMAT(&MIXED)
17 TYPE(DATAIN) CAPS(IN) HILITE(USCORE) FORMAT(&MIXED)
1D TYPE(DATAIN) INTENS(LOW) CAPS(IN) COLOR(BLUE) HILITE(USCORE)
  FORMAT(&MIXED)
20 TYPE(DATAIN) INTENS(LOW) CAPS(IN) FORMAT(&MIXED)
)BODY WIDTH(&ZWIDTH) EXPAND(//)
%EDIT -----!ZTITLE -----/-----%COLUMNS!ZCL
%COMMAND ==>_ZCMD / / %SCROLL ==
|ZDATA -----/-----
| / /
|-----/-----
)INIT
  &ZHIDEX = 'Y' /* Indicate the presence of */
                /* attrs 13, 16, 17 and 1D */
                /* for HIDE EXCLUDED */
  IF (&ZVMODET = 'VIEW') /* VIEW MODE */
    .HELP = ISR10000 /* DEFAULT TUTORIAL NAME */
  ELSE /* EDIT MODE */
    .HELP = ISR20000 /* DEFAULT TUTORIAL NAME */
    .ZVARS = 'ZSCED' /* SCROLL AMT VARIABLE NAME */
    &MIXED = MIX /* SET FORMAT MIX */
    IF (&ZPDMIX = N) /* IF EBCDIC MODE REQUESTED */
      &MIXED = EBCDIC /* SET FORMAT EBCDIC */
    VGET (ZSCED) PROFILE /* FILL SCROLL VARS IF */
    IF (&ZSCED = ' ') /* BLANK WITH PAGE. */
      &ZSCED = 'PAGE' /*
)REINIT
  REFRESH(ZCMD,ZSCED,ZDATA,ZTITLE,ZCL,ZCR)
  IF (&ZVMODET = 'VIEW') /* VIEW MODE */
    .HELP = ISR10000 /* DEFAULT TUTORIAL NAME */
  ELSE /* EDIT MODE */
    .HELP = ISR20000 /* DEFAULT TUTORIAL NAME */
)PROC
  &ZCURSOR = .CURSOR
  &ZCSROFF = .CSRPOS
  &ZLVLINE = LVLIN(ZDATA)
  VPUT (ZSCED) PROFILE
)END
/*****
/* Use variable ZDSVOL to display the volume of the data set */
/*****
/* DYNAMIC AREA SCREEN WIDTH FROM PQUERY. (80,132,160)
/* DYNAMIC AREA SCREEN DEPTH FROM PQUERY. (24,32,43,27,60)
/*
/* 5645-001, 5655-042 (C) COPYRIGHT IBM CORP 1980, 1996 */

```

Figure 67. Edit panel (ISREFR01)

The distributed panels have a dynamic area that automatically fills the available width and depth of the screen (after the first two lines). A customized panel can have a dynamic area that is fixed in width (in which case the WIDTH and EXPAND keywords should be omitted from the )BODY header statement), fixed in depth (in which case the EXTEND keyword should be omitted on the AREA statement in the )ATTR section), or fixed in both width and depth.

**Note:** If the dynamic area is less than the full width of the screen, the panel definition must include attribute bytes for protected fields on either side of the area.

See the description of dynamic areas in the *z/OS ISPF Dialog Developer's Guide and Reference*.

A customized panel can have additional text, input, or output fields outside the dynamic area. Any additional variable fields are transparent to the Browse and Edit programs. They can, however, be processed by the dialog that invoked Browse or Edit, or by edit macros provided by the dialog developer. Additional variable fields that appear on the panel should be refreshed (by using a REFRESH statement) in the )REINIT section of the panel definition. This ensures that the display screen is updated with the current contents of the variables each time the panel is displayed, which is after execution of each macro or built-in command, including SCROLL commands.

Any variables that are to be passed from the panel to an edit macro should be stored in the shared or profile pool by including a VPUT statement in the )PROC section of the panel definition. The macro must then issue a VGET to obtain them. The reason is that macros operate as nested dialogs, with a separate function pool from that of the dialog that invokes Edit.

The statements that appear in the )ATTR, )INIT, )REINIT, and )PROC sections of the distributed panels should be included in customized panels, with the possible exception of the EXTEND, WIDTH, and EXPAND keywords (discussed here) and the initialization of the scroll fields. Additional keywords can be added to the attribute definitions (for example, to produce different colors), but the same hexadecimal representation of the attribute bytes must be maintained, including the X'20' specified on the USERMOD keyword in the Edit panel.

The attribute keywords of the last attribute byte set before the dynamic area and of the attribute byte represented by X'01' should be the same for panel ISRBROB and any panels using it for an example.

Table 28 and Table 29 on page 172 list the ISPF-provided output-only variables you can use on Browse and Edit panels.

*Table 28. Browse output-only variables*

Variable	Description	Format/Length
ZCOLUMNS	First and last columns being displayed	CHAR 7
ZCSROFF	Cursor offset within cursor field	CHAR 4
ZCURSOR	Name of field where cursor is placed	CHAR 8
ZDADWD	Width of the dynamic area (ZDATA)	CHAR 4
ZDAMLN	Length of the dynamic area (ZDATA)	CHAR 4
ZDSN	Name of data set being displayed	CHAR 44
ZLEVEL	Version and mod level of member	CHAR 8

Table 28. Browse output-only variables (continued)

Variable	Description	Format/Length
ZLINES	Top line of the data display	CHAR 8
ZLVLINE	Last visible line of ZDATA after last interaction	CHAR 4
ZMEMB	Name of member being displayed	CHAR 8
ZTITLE	The title line DSN(MEMB) - VER.MOD	CHAR 65
ZWIDTH	Width of the panel	CHAR 4
ZDSVOL	Volume of the data set or first library in the concatenation	CHAR 6

Table 29. Edit output-only variables

Variable	Description	Format/Length
ZCL	Left column of the data display	CHAR 5
ZCR	Right column of the data display	CHAR 5
ZCSROFF	Cursor offset within cursor field	CHAR 4
ZCURSOR	Name of field where cursor is placed	CHAR 8
ZDADLN	Length of the dynamic area (ZDATA)	CHAR 4
ZDADWD	Width of the dynamic area (ZDATA)	CHAR 4
ZDSN	Name of data set being displayed	CHAR 44
ZLEVEL	Version and mod level of member	CHAR 8
ZLVLINE	Last visible line of ZDATA after last interaction	CHAR 4
ZMEMB	Name of member being displayed	CHAR 8
ZTITLE	The title line DSN(MEMB) -VER.MOD	CHAR 65
ZWIDTH	Width of the panel	CHAR 4
ZDSVOL	Volume of the data set or first library in the concatenation.	CHAR 6

## Enabling Browse panels to display Unicode data

You can enable users to browse data that is stored in a Unicode format. To do this, MVS Conversion Services must first be set up for the appropriate conversions. See *z/OS Support for Unicode: Using Unicode Services*.

## Customizing member list panels

Dialog developers can also provide customized member list display panels for their dialogs. The name of the customized member list display panel is passed as a parameter to the LMMDISP service.

A customized member list panel should be modeled after the LMMDISP default member list panel, ISRML000.

**Note:** Panel ISRML000 is provided as both source code (written in Dialog Tag Language, or DTL) and as generated output. If you use this panel as a model for creating your own customized panel, be sure to copy the DTL source code for modification, as it is easier to modify than the generated output panel. The DTL source code can be found in one of the SISPGxxx libraries, where xxx is the designator for a specified language.

The customized panel must include a Command field named ZCMD and a dynamic area named ZDATA. In addition, for customized member list panels, the dynamic area must be 80 characters wide. The Scroll field is optional. The name of the Scroll field on member list panels is ZSCML. This same name can be used on a customized member list panel. Inclusion of the variables that output the data set name, relative row number, and total number of rows is optional. If you want to display the volume of the data set you can use variable ZDSVOL. Customized member list panels should also meet the requirements for a movable Command line. Figure 68 shows an example of a customized member list panel that was generated by Dialog Tag Language code.

See the *z/OS ISPF Dialog Developer's Guide and Reference* for a discussion of these needs.

```
)PANEL KEYLIST(ISRSAB,ISR)
)ATTR DEFAULT(|_>) FORMAT(MIX)
0B TYPE(AB)
0D TYPE(PS)
2E TYPE(ABSL) GE(ON)
2F TYPE(PT)
29 TYPE(FP)
0A TYPE(NT)
13 TYPE(NEF) PADC(USER)
16 TYPE(VOI) PADC(USER)
26 TYPE(VOI) CAPS(ON) PADC(USER)
27 AREA(DYNAMIC)
08 TYPE(DATAOUT) PAS(ON)
09 TYPE(DATAOUT)
2A AREA(DYNAMIC) EXTEND(ON) SCROLL(ON)
01 TYPE(DATAIN) CAPS(ON) PADC(&ZMLPAD)
02 TYPE(DATAOUT) INTENS(&MLI2) COLOR(&MLC2) HILITE(&MLH2)
03 TYPE(DATAIN) CAPS(ON)
04 TYPE(DATAOUT) INTENS(&MLI3) COLOR(&MLC3) HILITE(&MLH3)
05 TYPE(DATAOUT)
06 TYPE(DATAOUT) INTENS(LOW)
15 TYPE(NEF) CAPS(ON) PADC(USER)
)ABC DESC('Functions') MNEM(1)
PDC DESC('Save') MNEM(1) ACTION RUN(SAVE)
PDC DESC('Change Colors...') MNEM(1) ACTION RUN(MLC)
)ABCINIT
.ZVARS=MEMOPT

)ABC DESC('Help') MNEM(1)
PDC DESC('General') MNEM(1) ACTION RUN(TUTOR) PARM('ISR01130')
PDC DESC('Scrolling') MNEM(1) ACTION RUN(TUTOR) PARM('ISR01131')
PDC DESC('Pattern matching') MNEM(1) ACTION RUN(TUTOR) PARM('ISR01232')
PDC DESC('LOCATE command') MNEM(1) ACTION RUN(TUTOR) PARM('ISR01132')
PDC DESC('SORT command') MNEM(2) ACTION RUN(TUTOR) PARM('ISR01226')
PDC DESC('SAVE command') MNEM(2) ACTION RUN(TUTOR) PARM('ISR01229')
PDC DESC('RESET command') MNEM(1) ACTION RUN(TUTOR) PARM('ISR01138')
PDC DESC('SELECT command') MNEM(2) ACTION RUN(TUTOR) PARM('ISR01133')
PDC DESC('S line command') MNEM(4) ACTION RUN(TUTOR) PARM('ISR01134')
PDC DESC('Statistics') MNEM(2) ACTION RUN(TUTOR) PARM('ISR01140')
PDC DESC('Appendices') MNEM(5) ACTION RUN(TUTOR) PARM('ISR00004')
PDC DESC('Index') MNEM(3) ACTION RUN(TUTOR) PARM('ISR91000')

)ABCINIT
.ZVARS=MEMLHELP
```

Figure 68. Customized member list panel (Part 1 of 2)

```

)BODY  CMD(ZCMD)

+
  Functions  Help%-----
%MEMBER LIST -- &ZDSN -----%ROW ZMLCR%OF ZMLTR+
%COMMAND ==>_ZCMD                %SCROLL ==>_Z  +
'ZMLCOLD
'ZDATA
'
)INIT
.ZVARS = '(ZDSN ZMLCR ZMLTR ZCMD ZSCML)'
.HELP = ISRO1130
.ATTR(ZMLCR)='JUST(RIGHT) PAD('0'' )'
.ATTR(ZMLTR)='JUST(RIGHT) PAD('0'' )'
VGET (MLC1 MLC2 MLC3 MLH1 MLH2 MLH3 ZMLPD) PROFILE

IF (&ZGUI = ' ' AND &ZMLPD = ' ')
  &ZMLPAD = ' _ '
ELSE
  &ZMLPAD = 'USER'
&ZSCKEY = 'ZFRML'
IF (&MLC1 = ' ')
  &MLC1 = 'TURQ'
IF (&MLC2 = ' ')
  &MLC2 = 'BLUE'
IF (&MLC3 = ' ')
  &MLC3 = 'GREEN'

VGET (ZSCML) PROFILE      /* Fill Scroll Vars if      */
IF (&ZSCML = ' ')        /* Blank with page.        */
  &ZSCML = 'PAGE'        /*                          */
)PROC
VPUT (ZSCML) PROFILE
IF (.CURSOR = ZDATA OR .CURSOR = ZMLCOLD)
  &ZMSCPOS = &ZCURPOS;
ELSE
  &ZMSCPOS = '0000'
/*****
/* Use variable ZDSVOL to display the volume of the data set      */
*****/
)PNTS
)END
/* 5655-042 (C) COPYRIGHT IBM CORP 1980, 1995 */
/* ISPD TLC Release: 4.3.  Level: PID                      */
/* OS/390 R.3.  Created - Date: 14 Oct 1996, Time: 11:18:33  */

```

Figure 68. Customized member list panel (Part 2 of 2)

As with edit and browse customized panels, the distributed member list panels have a dynamic area that extends to fill the available depth of the screen. But, unlike edit and browse panels, (the dynamic area on a member list panel does not expand to fill the width of the screen), it must be 80 characters wide. The dynamic area can also be fixed in length by removing the EXTEND keyword from the )ATTR section of the panel definition.

See the description of dynamic areas in the *z/OS ISPF Dialog Developer's Guide and Reference*.

See "Customizing Browse and Edit" on page 167 for a more complete discussion on customizing panels.

Table 30 lists the ISPF-provided output-only variables you can use on member list panels.

Table 30. Member list panel output-only variables

Variable	Description	Format/Length
ZDAZWD	Width of dynamic area (ZDATA)	CHAR 4
ZDAMLN	Length of dynamic area (ZDATA)	CHAR 4
ZDSN	First or only data set in concatenation	CHAR 44
ZMLCOLS	Member statistics column headings	CHAR 80
ZMLCR	Relative row number of top row	FIXED 4
ZMLTR	Total number of rows in member list	FIXED 4
ZDSVOL	Volume of the data set or first library in the concatenation.	CHAR 6

---

## Customizing the z/OS UNIX directory list panel

Dialog developers can also provide a customized z/OS UNIX directory list display panel for their dialogs. The name of the customized directory list display panel is passed as a parameter to the DIRLIST service.

A customized directory list panel must be modeled after the DIRLIST default directory list panel, ISRUUDL0.

**Note:** Panel ISRUUDL0 is shipped as both source code (written in Dialog Tag Language, or DTL) and as generated output. If you use this panel as a model for creating your own customized panel, you have the option of modifying a copy of the DTL source code and using this to generate the customized panel. The DTL source code can be found in one of the SISPGxxx libraries, where xxx is the designator for a specified language.

The customized panel must include a Command field named ZCMD and a dynamic area named ZDATA which is used to display the column headings. ISPF uses the TBDISPL service to display the directory list and dynamically builds a model line in the value of variable ZULMODL. Therefore, the customized panel must specify variable ZULMODL as a variable model line. The panel is designed to use the available screen width and both the model line and column heading line are required to expand for the width of the screen. ISPF dynamically builds a list of ZVARs for the panel in the value of variable ZULZVARs.

For information on variable model lines refer to the description of the TBDISPL service in the *ISPF Services Guide*, and the section titled "Requirements for Model Section" in the *ISPF Dialog Developer's Guide and Reference*.

---

## IBM Products option

Module ISRALTDI is used to determine if products are installed for the IBM Products Option (option 9). Products are considered to be installed if a specific panel for the product can be found in the panel library. If the panel is found, then ISPF will attempt to invoke it. No other checks are made to see if the products are correctly and completely installed or available.

For Information/System invocation, all parameters are allowed to default. To add parameters to the invocation, modify panel ISRDINFX by placing this statement immediately after the string PGM(BLGINIT):

PARM(*your additional parameters*)

To pass suboptions to the product panels, use the TRUNC and TRAIL functions with the ZALTTR variable before invoking program ISRALTDI. ZALTTR can hold up to 80 characters. Use this code:

```
&ZALTTR; = TRUNC(&ZCMD, '.')
&ZALTTR; = .TRAIL
&ZSEL; = TRANS(TRUNC*&ZCMD, '.')
```

The operation of ISRALTDI is:

1. Search the panel library for the primary panel.
2. If the primary panel exists, SELECT or DISPLAY the primary or alternate panel. If the primary panel does not exist, search the panel library for the secondary panel.
3. If the secondary panel exists, SELECT or DISPLAY the secondary or alternate panel. If the secondary panel does not exist, set an error message.

ISRALTDI is intended to be invoked from a selection panel. The parameter list follows:

```
SELECT PGM(ISRALTDI) PARM(
    Primary-panel
    , < Primary-application-id >
    , Secondary-panel
    , < Secondary-application id >
    , < Alternate-panel >
    , < Alternate-application-id >
) NOCHECK
```

After the last nonblank parameter, trailing commas are optional. The parameters are described here.

Primary-panel and secondary-panel are panel names and are required.

Primary- and secondary-application-id can be:

<b>APPLICATION ID</b>	0 to 4 character APPLID under which a panel is to be SELECTed.
*	DISPLAY this panel.
+	Use alternate panel.

Alternate-panel is an optional panel name.

Alternate APPLID can be:

<b>APPLICATION ID</b>	0 to 4 character APPLID under which a panel is to be SELECTed.
*	DISPLAY this panel.

Some examples of the parameter string for ISRALTDI follow:

- If PANEL1 exists, SELECT it with application ID 'AP'. Otherwise, DISPLAY panel 'NOTHERE'.  
PANEL1,AP,NOTHERE,\*
- If PANEL1 exists, DISPLAY it. Otherwise, DISPLAY panel 'NOTHERE'.  
PANEL1,\*,NOTHERE,\*
- If PANEL1 exists, SELECT PANEL3 with application ID 'CSP'. Otherwise, SELECT panel 'UNAVAIL' with application ID 'ISR'.  
PANEL1,+,UNAVAIL,ISR,PANEL3,CSP

- If PANEL1 exists, SELECT PANEL3 with application ID 'CSP'. Otherwise, SELECT panel 'UNAVAIL' with current APPLID.  
PANEL1,+,UNAVAIL,,PANEL3,CSP
- If PANEL1 exists, SELECT PANEL1 with current application ID. Otherwise, SELECT panel 'UNAVAIL' with current APPLID.  
PANEL1,,UNAVAIL



## Part 2. Reference

<b>Chapter 8. PDF installation-wide exits</b> . . . . .	181	Error processing . . . . .	202
Data set list filter exit. . . . .	181	Return codes . . . . .	202
Analyze Dsname and Volume . . . . .	181		
Analyze data set name . . . . .	182	<b>Chapter 9. ISPF Configuration Table keywords</b>	
Return codes . . . . .	182	<b>and values</b> . . . . .	203
Data set allocation exit . . . . .	182	PDF exits . . . . .	203
Exit parameters. . . . .	182	Data set allocation settings . . . . .	204
Return codes . . . . .	186	Outlist data set specifications . . . . .	205
Activity monitoring exits . . . . .	186	SuperC data set specifications . . . . .	205
Exit parameters. . . . .	186	LMF . . . . .	206
Usage notes . . . . .	186	Edit recovery data set specifications . . . . .	206
Exit 1: BROWSE primary command start . . . . .	187	Move/Copy settings . . . . .	207
Exit 2: BROWSE primary command end . . . . .	187	DSLISL removable media interface settings . . . . .	207
Exit 3: EDIT primary command start . . . . .	187	Edit-related settings . . . . .	209
Exit 4: EDIT primary command end . . . . .	187	Edit site-wide profile customizations . . . . .	212
Exit 5: EDIT macro start . . . . .	188	ISPF site-wide profile customizations . . . . .	217
Exit 6: EDIT macro end . . . . .	188	Default CUA color settings . . . . .	227
Exit 7: EDIT line command start . . . . .	188	Miscellaneous settings . . . . .	231
Exit 8: EDIT line command end . . . . .	188	Values formerly in ISPDFLTS . . . . .	235
Exit 9: Library/Data Set utility sub-function start . . . . .	188	VSAM data set restrictions . . . . .	235
Exit 10: Library/Data Set utility sub-function end . . . . .	189	ISPSPROF general values . . . . .	236
Exit 11: Data Set List utility line command start . . . . .	189	ISPSPROF workstation defaults . . . . .	243
Exit 12: Data Set List utility line command end . . . . .	190	ISPSPROF workstation download defaults. . . . .	247
Error processing . . . . .	190		
Return codes . . . . .	190	<b>Chapter 10. Exits</b> . . . . .	249
Data set list line command exit . . . . .	190	Exit macros . . . . .	249
Exit parameters. . . . .	190	Exit 1: ISPF session initialization exit . . . . .	251
Usage notes . . . . .	190	Exit parameters. . . . .	251
Error processing . . . . .	191	Return codes . . . . .	251
Return codes . . . . .	191	Exit 2: ISPF session termination exit . . . . .	251
Print utility exit . . . . .	192	Exit parameters. . . . .	252
Exit parameters. . . . .	192	Return codes . . . . .	252
Return codes . . . . .	193	Exit 3: SELECT service start exit . . . . .	252
Error processing . . . . .	193	Exit parameters. . . . .	252
Print utility exit on ISPF termination and		Return codes . . . . .	253
LOG/LIST commands . . . . .	193	Exit 4: select service end exit . . . . .	254
Exit parameters. . . . .	194	Exit parameters. . . . .	254
Return codes . . . . .	195	Return codes . . . . .	255
Error processing . . . . .	196	Exit 5: TSO command start exit . . . . .	255
Compress request exit . . . . .	196	Exit parameters. . . . .	255
Exit parameters. . . . .	196	Return codes . . . . .	256
Return codes . . . . .	197	Exit 6: TSO command end exit . . . . .	256
Error processing . . . . .	197	Exit parameters. . . . .	256
Data set name change exit . . . . .	197	Return codes . . . . .	256
Exit parameters. . . . .	198	Exit 7: LIBDEF service exit . . . . .	257
Usage notes . . . . .	199	Exit parameters. . . . .	257
Member list filter exit . . . . .	199	Return codes . . . . .	257
Analyze patterns . . . . .	200	Exit 8: RESERVE exit . . . . .	258
Exit parameters. . . . .	200	Exit parameters. . . . .	258
Return codes . . . . .	200	Return codes . . . . .	258
Analyze member names. . . . .	200	Exit 9: RELEASE exit . . . . .	258
Exit parameters. . . . .	200	Exit parameters. . . . .	259
Return codes . . . . .	200	Return codes . . . . .	259
Member list built-in line command exit. . . . .	201	Exit 10: logical screen start exit . . . . .	259
Exit parameters. . . . .	201	Exit parameters. . . . .	259
Usage notes . . . . .	201	Return codes . . . . .	259

Exit 11: logical screen end exit . . . . .	259
Exit parameters. . . . .	260
Return codes . . . . .	260
Exit 12: ISPF service start exit . . . . .	260
Exit parameters. . . . .	260
Return codes . . . . .	260
Exit 13: ISPF service end exit . . . . .	260
Exit parameters. . . . .	261
Return codes . . . . .	261
Exit 14: SWAP exit. . . . .	261
Exit parameters. . . . .	261
Return codes . . . . .	261
Exit 15: DISPLAY service exit . . . . .	261
Exit parameters. . . . .	262
Return codes . . . . .	263
Exit 16: log, list, trace, and temporary data set allocation exit . . . . .	263
Exit parameters. . . . .	263
Return codes . . . . .	264

<b>Chapter 11. PDF translation tables . . . . .</b>	<b>265</b>
Translation table for valid data set name characters	265
Translation table for invalid data set name characters . . . . .	265
Translation table for hexadecimal characters . . . . .	265
Translation table for numeric characters . . . . .	266
Translation table for alphanumeric characters. . . . .	266
Translation table for edit terminal output characters	267
Translation table for generic string characters. . . . .	267
Translation table for generic string special characters . . . . .	267
Usage notes for the GSC and GSS tables . . . . .	268
Translation table for uppercase characters . . . . .	268
Translation table for lowercase characters . . . . .	269
Modifying the GSM to use the user character subset . . . . .	269
Generic string master translation table . . . . .	269
ISPF and PDF terminal translation table relationship . . . . .	271

<b>Chapter 12. Dialog development model listings</b>	<b>273</b>
DM and PDF services in CLIST commands . . . . .	273
DM and PDF services in COBOL programs . . . . .	275
DM and PDF services in EXEC commands . . . . .	278
DM and PDF services in FORTRAN programs . . . . .	279
Message format . . . . .	282
Panel formats and statements . . . . .	282
DM and PDF services in PL/I programs . . . . .	285
File tailoring control statements . . . . .	287
DM and PDF services in Pascal programs . . . . .	287
DM and PDF services in TSO/REXX commands	290
SCLM architecture definition formats . . . . .	292
SCLM project definition macros and templates . . . . .	293
ISPF Dialog Tag Language models . . . . .	293
DM and PDF services in C/370 programs . . . . .	293
Dialog Tag Language models . . . . .	296

<b>Chapter 13. Programming interface macros for customers . . . . .</b>	<b>299</b>
---	------------

<b>Chapter 14. ISPF data set descriptions . . . . .</b>	<b>301</b>
---	------------

---

## Chapter 8. PDF installation-wide exits

### Programming Interface information

PDF allows installations to satisfy unique processing needs by providing installation-wide exits at these points:

- Data set list and member list
- Data set allocation
- Print utility
- Compress request
- Data set name and member name change.

The sections that follow describe each of these installation-wide exits and list the parameters, return codes, and any error processing.

**Note:** All PDF exits should be AMODE(31). Consider making any exit routines that you write reusable, and preferably reentrant. If you write exits that are *not* reentrant, they cannot be put in the Link Pack Area (LPA) library. Non-reentrant exit routines placed in the LPA can cause abend errors.

---

### Data set list filter exit

In the ISPF configuration table keyword file, this exit is set with keyword `DATA_SET_LIST_FILTER_PROGRAM_EXIT`.

The data set list filter exit is used by option 3.4 to allow the installation to determine what data set names appear in the data set list. The exit is called at two different points:

- First, with the `DSNAME LEVEL` and `VOLUME` specified on option 3.4 panel
- Second, for each data set whose name matches the `DSNAME LEVEL` and `VOLUME`.

If an exit routine is not provided, PDF uses the `dsname level` and `volume` specified on the panel and adds all data sets that match the data set name level and volume to the list.

PDF calls this exit routine using the standard conventions. The routine must be a program. All ISPF and system services are available to the routine.

See “Disable generic high-level qualifiers” on page 74 for an example of a data set list filter exit.

### Analyze Dsname and Volume

Parameter	Type	Len	Description	Modifiable
CODE	FIXED	4	Code that indicates that this is the analyze dsname and volume call. Call=1.	No
DSNAME LEVEL	CHAR	44	Data set name level that was specified on the option 3.4 panel.	Yes
VOLUME	CHAR	6	Volume that was specified on the option 3.4 panel.	Yes

## Analyze data set name

Parameter	Type	Len	Description	Modifiable
CODE	FIXED	4	Code that indicates that this is the analyze data set name call. Call=2	No
DSNAME LEVEL	CHAR	44	Data set name level that you specified on the option 3.4 panel.	No
VOLUME	CHAR	6	Volume that you specified on the option 3.4 panel.	No
DATA SET NAME	CHAR	44	Data set name that is to be added to the list.	No
VOLUME NAME	CHAR	6	Volume on which this data set resides.	No

## Return codes

With the DSNAME LEVEL and VOLUME specified on option 3.4 panel.

- 0 The data set name and volume is used as entered on the panel.
- 4 Either the data set name or the volume has been modified by the exit and the modified data set name or volume is used.
- 8 A list is not displayed.

For each data set whose name matches the DSNAME LEVEL and VOLUME.

- 0 Data set name and volume is included in the list without any modification.
- 4 Data set name is not added to the list.
- 8 The list is stopped, no more items are added to the list.

---

## Data set allocation exit

In the ISPF configuration table keyword file, this exit is set with keyword DATA\_SET\_ALLOCATION\_PROGRAM\_EXIT.

You can specify an installation-written exit routine to create, delete, allocate, and deallocate data sets instead of using those functions provided by PDF. However, allocations done by ISPF, the TSO ALLOCATE command, or TSO commands are not handled by the exit. If you use your own data set allocation exit routine, it must be a program. CLISTs are not allowed.

## Exit parameters

PDF passes the SVC 99 parameter list/dynamic allocation request table as input to the exit routine. The information in the parameter list depends on the type of request being processed and is obtained from the user. Register 1 points to the parameter list.

PDF uses these parameters to communicate with the allocation exit:

- SVC99 parmlist pointer** A pointer to the SVC99 parmlist/dynamic allocation request table.
- User storage pointer** A pointer to a 120-byte area that can be used to add new or changed text units.
- Path name pointer** A pointer to an area of storage containing the absolute path name of the z/OS UNIX file to be

allocated. Set to zero if this is not a request to allocation a z/OS UNIX file.

**Path name length**

A fullword binary integer that is the length of the z/OS UNIX file path name. Set to zero if this is not a request to allocation a z/OS UNIX file.

In addition to these parameters, PDF allocates enough space for a total of 30 text unit pointers in the text unit pointer list. The unused pointers can be used in conjunction with the 120-byte user data area for adding or changing text units.

If the request is to *allocate* a data set, the parameter list contains the information in Table 31.

*Table 31. Parameter list during allocation*

Key	#Parms	Length	Parameter	Description	Exit may modify these parameters
0001	1	8	DDNAME	ddname to allocate data set to	No
0002	1	44	DSNAME	Data set name to allocate	No
0004	1	1	STATUS	Data set status	Yes
0005	1	1	NDISP	Normal data set disposition	Yes
0006	1	1	CDISP	Conditional data set disposition	Yes
0010	1	6	VLSER	Volume serial	No
0015	1	8	UNIT	unit name	No
0050	1	8	PASSW	Password for protected data set if a password was specified	No
0057	1	8	RTORG	Return data set organization	No
005D	1	6	RTVOL	Return volume serial from allocation	No
8017	1	1023 (max)	PATHNAME	Path name of the z/OS UNIX file to allocate. Pathname is of the form: <i>/dev/fdnnn</i>  where <i>nnn</i> is the file descriptor number. The real path name can be obtained via the path name pointer and path name length parameters.	No
8018	1	4	PATHOPT	The file options for the z/OS UNIX file	No
801D	1	1	FILEORG	The organization of the z/OS UNIX file	No

If the request is to *concatenate* a data set, the parameter list contains the information in Table 32 on page 184.

Table 32. Parameter list during concatenation

Key	#Parms	Length	Parameter	Description	Exit may modify these parameters
0001	1	8	DDNAMEX	List of ddnames corresponding to the data sets being concatenated	No
0004	0	0	PERMCC	Permanently concatenated attribute	No

If the request is to *create* a data set, the parameter list contains the information in Table 33.

Table 33. Create data set allocation parameter list

Key	#Parms	Length	Parameter	Description	Modifiable?
0001	1	8	DDNAME	ddname to allocate data set to	No
0002	1	44	DSNAME	Data set name to allocate	No
0004	1	1	STATUS	Data set status	Yes
0005	1	1	NDISP	Normal data set disposition	Yes
0006	1	1	CDISP	Conditional data set disposition	Yes
0007	1	0	TRKS	Space allocated in tracks or	Yes
0008	1	0	CYLS	Space allocated in cylinders, or	Yes
0009	1	3	BLKS	Space allocated in blocks	Yes
000A	1	3	PSPACE	Primary space quantity	Yes
000B	1	3	SSPACE	Secondary space quantity	Yes
000C	1	3	DBLKS	Number of directory blocks	Yes
0010	1	6	VOLSER	Volume serial	Yes
0015	1	8	UNIT	Unit group (esoteric) name Device type Specific unit address	Yes
0020	1	1	PASPR	Data set is password protected:  X'10' Data set cannot be read, changed, extended, or deleted. X'30' Data set can be read, but not changed, extended, or deleted.	Yes
0022	1	5	EXPDT	Expiration date - YYDDD or	Yes
006D	1	7	EXPDTL	Expiration date - YYYYDDD or	Yes
0023	1	2	RETPD	Retention period	Yes
0030	1	2	BLKSZ	Block size	Yes
003C	1	2	DSORG	Data set organization	Yes
0042	1	2	LRECL	Logical record length	Yes
0049	1	1	RECFM	Record format	Yes
0050	1	8	PASSWORD	Data set password if specified	Yes
0052	0	0	PERM	Permanently allocated attribute	No
0057	1	2	RTORG	Return data set organization	No

Table 33. Create data set allocation parameter list (continued)

Key	#Parms	Length	Parameter	Description	Modifiable?
005D	1	6	RTVOL	Return volume serial from allocation	No
8004	1	8	STORCLAS	Storage class used to allocate the data set	Yes
8005	1	8	MGMTCLAS	Management class used to allocate the data set	Yes
8006	1	8	DATACLAS	Data class used to allocate the data set	Yes
8010	1	1	AVGREC	Unit of allocation in terms of average record size:  X'80' Bytes X'40' Kilobytes X'20' Megabytes	Yes
8012	1	1	DSNTYPE	Data set name type used to allocate the data set:  X'80' Library X'40' PDS X'10' HFS X'08' Extreq X'04' Extpref X'02' Basic X'01' Large	Yes

If the request is to *deallocate* a data set, the parameter list contains the information in Table 34.

Table 34. Parameter list during deallocation

Key	#Parms	Length	Parameter	Description	Exit may modify these parameters
0001	1	8	DDNAME	ddname to free	No
0007	0	0	UNALC	Unallocate option	No

If the request is to *delete* a data set, the parameter list contains the information in Table 35.

Table 35. Parameter list during deletion

Key	#Parms	Length	Parameter	Description	Exit may modify these parameters
0002	1	44	DSNAME	DSNAME to free	No
0007	0	0	UNALL	Unallocate option	No

If the request is to *delete* a data set, it is done through the MVS SCRATCH macro. This allows unexpired data sets to be deleted.

## Return codes

- 0 No errors; the exit has issued the SVC 99.
- 4 No errors; PDF issues the SVC 99.
- 8 Error occurred; the exit has formatted a message.
- 20 Severe error from the exit. The exit formats a message and PDF displays it in an error box.

If the allocation exit issues the SVC 99 and requests PDF to evaluate the error, the exit should return the SVC 99 return code to PDF in register 15 with the high-order bit of the register turned on.

If PDF issued the SVC 99, it places (by way of a VPUT) these variables into the shared pool in character format:

**Z99RC** Return code from SVC 99  
**Z99ERROR** SVC 99 error code  
**Z99INFO** SVC 99 information code.

---

## Activity monitoring exits

In the ISPF configuration table keyword file, this exit is set with keyword `ACTIVITY_MONITORING_PROGRAM_EXIT`.

The activity monitoring exits provide monitoring information for these PDF functions:

- Primary commands invoked from BROWSE
- Primary commands invoked from EDIT
- Primary commands invoked from VIEW
- Edit macros invoked from EDIT
- Line commands invoked from EDIT
- Line commands invoked from VIEW
- Library/Data Set utility sub-functions
- Data Set List utility built-in line commands.

The activity monitoring exit routine is given control at the start and end of each command or function.

## Exit parameters

PDF uses these parameters to communicate with the monitoring exit.

Parameter	Type	Len	Description	Modifiable
EXIT ID	FIXED	4	Numeric code for the exit point.	No
COMMAND/ SUB-FUNCTION ADDRESS	PTR	4	Storage location containing the command name or sub-function identifier.	No
COMMAND/ SUB-FUNCTION LENGTHS	FIXED	4	Length of the command name or sub-function identifier. The value ranges from 1-255.	No

## Usage notes

- Activity monitoring exits are the same for VIEW as for EDIT. From within the VIEW function, the primary and line commands are monitored exactly like they are from EDIT.

- A single exit routine is allowed for all monitoring PDF exit points. The name of the module is placed into the ISPF configuration table when the product is installed.
- 31-bit addressing must be supported by the exit routine as it is given control in AMODE 31. ISPF/PDF does not restrict the RMODE of the exit, but RMODE ANY is recommended.
- Standard OS linkage conventions are used to branch to the exit routine that is defined.
- You can make PDF service calls from within the exit routine, with the exception of those services that could cause recursion (for example, calling the EDIT service for an EDIT primary command exit call).
- You cannot activate or deactivate the exit while an PDF session is in progress. You can make changes to the exit routine load module at any time, but because PDF loads the module only once at session initialization, the changes are not recognized until the next ISPF session.
- PDF does not provide data areas. You can use a predetermined ddname as a work area. The ISPF initialization exit could allocate and open a data set to this ddname. Each PDF exit could read and write to the data set as needed. The ISPF termination exit could close and free the data set.

---

## **Exit 1: BROWSE primary command start**

This exit point marks the start of a valid BROWSE primary command. The exit routine is given control immediately before the function is performed.

---

## **Exit 2: BROWSE primary command end**

This exit point marks the end of a valid BROWSE primary command. The exit routine is given control immediately after the command is performed. It can be used together with the BROWSE primary command start exit point for monitoring purposes.

---

## **Exit 3: EDIT primary command start**

This exit point marks the start of a valid, user-entered EDIT primary command. The exit routine is given control immediately before the command is performed. All edit macro statements (such as ISREDIT SAVE) are processed as if they were primary commands. Individual edit macro statements executed from within the edit macro optionally invoke the exit routine. A field in the configuration table determines whether the activity monitoring exit is invoked at the start and stop of each edit macro command. If an alias has been defined for the command (via the DEFINE command of EDIT), and if the alias name is specified, the actual command that is invoked is passed to the exit.

---

## **Exit 4: EDIT primary command end**

This exit point marks the end of a user-entered EDIT primary command. The exit routine is given control immediately after the command is performed. It can be used together with the EDIT primary command start exit point for monitoring purposes.

---

## Exit 5: EDIT macro start

This exit point marks the start of a user-entered EDIT macro or an invalid EDIT primary command. The exit routine is given control immediately before the macro is invoked. Individual edit macro statements executed from within an edit macro optionally invoke the edit primary command exit routines (exits 3 and 4). Line commands processed by an edit macro optionally invoke the Edit line command exit routines (exits 7 and 8).

---

## Exit 6: EDIT macro end

This exit point marks the end of a user-entered EDIT primary command. The exit routine is given command immediately after the macro is performed. It can be used together with the EDIT macro command start exit point for monitoring purposes.

---

## Exit 7: EDIT line command start

This exit point marks the start of a valid user-entered EDIT line command. Line commands that are being processed by an edit macro optionally invoke the exit routine and indicate that it is a line command executed by an edit macro. The exit routine is given control immediately before the command is performed.

If multiple line commands are entered, the exit routine is given control once for each line command. A line command entered in block form (CC, for example) causes the exit to be called only once. Destination line commands (A,B,O) and defining line labels do not call the exit.

---

## Exit 8: EDIT line command end

This exit point marks the end of an Edit line command. The exit routine is given control immediately after the command is performed.

If multiple line commands are entered, the exit routine is given control once for each line command. A line command entered in the block form (CC, for example) causes the exit to be called only once. Destination line commands (A, B, O) and defining line labels do not cause the exit to be called.

This exit point can be used together with the Edit line command start exit point for monitoring purposes.

---

## Exit 9: Library/Data Set utility sub-function start

This exit point marks the start of a sub-function from either the Library (option 3.1) or Data Set (option 3.2) Utilities. These are the sub-function identifiers and their descriptions:

<b>A</b>	Allocate data set
<b>C</b>	Catalog data set
<b>D</b>	Delete data set
<b>I</b>	Data set information
<b>M</b>	Display member list
<b>MB</b>	Browse member
<b>MD</b>	Delete member
<b>MP</b>	Print member
<b>MR</b>	Rename member
<b>P</b>	Print entire data set

R	Rename data set
RU	Rename data set using ISPF option 3.2 (catalog is not updated)
S	Data set information (short)
U	Uncatalog data set
X	Print data set index listing
Z	Compress data set

The exit routine is given control immediately before the sub-function is performed.

If multiple line commands (member list) are entered, the exit routine is given control once for each sub-function.

---

## Exit 10: Library/Data Set utility sub-function end

This exit point marks the end of a Library or Data Set utility sub-function. The exit routine is given control immediately after the sub-function is performed. It can be used together with the Library/Data Set utility sub-function start exit point for monitoring purposes.

---

## Exit 11: Data Set List utility line command start

This exit point marks the start of a line command from the Data Set List (option 3.4) utility. These are the subfunction identifiers and their descriptions:

Identifier	Description
B	Browse data set
C	Catalog data set
D	Delete data set
E	Edit data set
F	Free unused space
I	Data set information
M	Display member list
MB	Browse member
MD	Delete member
ME	Edit member
MP	Print member
MR	Rename member
P	Print entire data set
R	Rename data set
S	Data set information (short)
U	Uncatalog data set
X	Print data set index listing
Z	Compress data set.

In addition to the one character command name, the name of the data set is provided for all commands that operate on entire data sets. For example, when an R is typed next to data set MY.TEST.DATA, the command pointed to by the command parameter is:

```
R 'MY.TEST.DATA'
```

Note that the data set name provided is the one shown in the data set list, even if the name has been changed by a data set name change exit.

For the M (member list) command, the command parameter includes the data set name. However, for commands executed within the member list, just the command code is given (such as MB for the B (browse) line command).

The exit routine is given control immediately before the command is performed.

If multiple line commands are entered, the exit routine is given control once for each line command. Only built-in commands cause the exit to be called.

---

## Exit 12: Data Set List utility line command end

The exit routine is given control immediately after the command is performed.

If multiple line commands are entered, the exit routine is given control once for each line command. Only built-in commands cause the exit to be called.

This exit point is provided to mark the end of a Data Set List utility line command. It can be used together with the Data Set List utility line command start exit point for monitoring purposes.

### Error processing

If a nonzero return code is returned from an exit call, a message is conditionally set.

### Return codes

0 Normal Completion  
Nonzero Severe Error.

The ISPF Select Exit can be used by installations to monitor TSO commands and CLISTs issued from the Data Set List utility.

---

## Data set list line command exit

In the ISPF configuration table keyword file, this exit is set with keyword DATA\_SET\_LIST\_LINE\_COMMAND\_PROGRAM\_EXIT.

The data set list line command exit allows installations to change or restrict the line commands entered in the Data Set List utility.

### Exit parameters

ISPF uses these parameters to communicate with the line command exit routine.

Parameter	Type	Len	Description	Modifiable
LINE COMMAND	CHAR	255	A 255-line character field containing the data set list line command. The field is left-justified with trailing blanks.	Yes

### Usage notes

- The name of the module is placed into the ISPF configuration table when the product is installed.

- 31-bit addressing must be supported by the exit routine as it is given control in AMODE 31. PDF does not restrict the RMODE of the exit, but RMODE ANY is recommended.
- Standard OS linkage conventions are used to branch to the exit routine that is defined.
- You can make PDF service calls from within the exit routine.
- You cannot activate or deactivate the exit while a PDF session is in progress. You can make changes to the exit routine load module at any time, but because PDF loads the module only once at session initialization, the changes are not recognized until the next ISPF session.
- On input to the exit, the 255-character field contains the line command (left justified) after ISPF substitutes the data set name. If you explicitly specify a slash character (/) in place of the data set name, the slash is replaced by the data set name before the exit receives the command. If you do not specify a slash, the data set name is appended to the end of the command that you enter. One or more spaces will precede the data set name in this case.
- The exit can change the command buffer, subject to these rules:
  - If the final command to be executed is not one of the ISPF built-in commands (E, C, V, and so on), then the buffer can be changed to anything. ISPF does no validation before sending the command to TSO.
  - If the final command to be executed *is* one of the built-in commands, the data set name cannot be changed.

Thus, user commands can be converted to built-in commands, built-in commands can be converted to user commands (like CLISTs), or the command string can be blanked out entirely to ignore the command.

- The Slash (/) command is considered a single command. Selections from the pop-up list are not reprocessed by the line command exit.
- Because the format of the data set list line command is free form, the exit should not have any dependencies on specific columns, or on specific numbers of blanks in the string.
- The intended interface for saving data across exit invocations is through standard ISPF methods (variables or tables).
- If the command buffer is changed, the exit must return a code of 4 in register 15 to notify ISPF.

If the exit name has been specified in the ISPF configuration table the exit is called before each line command is processed. The exit can analyze the line command and determine whether you should invoke the command as entered, as modified by the exit, or not at all.

If the exit modifies the line command, it is responsible for ensuring that the modified line command is valid.

## Error processing

If a nonzero return code is returned from an exit call, a message is conditionally set.

## Return codes

- |    |   |
|----|---|
| 0  | Use line command as specified by the user |
| 4  | Use line command as changed by the exit   |
| 8  | Do not invoke this command                |
| 20 | Severe error occurred.                    |

---

## Print utility exit

The Print utility exit receives control from ISPF in two distinct modes:

- Print utility exit for general printing (described here).
- Print utility exit on ISPF termination and LOG/LIST commands (see page 193).

In the ISPF configuration table keyword file, this exit is set with keywords PRINT\_UTILITY\_PROGRAM\_EXIT and PRINT\_UTILITY\_COMMAND\_EXIT.

The print utility exit lets you define your own print facility to replace or supplement PDF's print facilities. If you define a print routine for this exit, PDF calls that routine from options 3.6, 3.8, and all 4.x options. The print routine is also called on ISPF termination and for the LOG and LIST commands as described in "Print utility exit on ISPF termination and LOG/LIST commands" on page 193.

The print routine can:

- Define its own print facility or use the TSO/E Information Center Facility printer definitions. To indicate that you are using the TSO/E Information Center Facility printer definitions, set the PRINT\_USING\_ICF keyword in the ISPF configuration table to YES.
- Specify additional print options.
- Supply print parameters automatically.
- Censor print requests.

## Exit parameters

PDF communicates with the installation-written print routine through dialog variables in the ISPF shared pool. PDF passes data set, job card, and print option information to the print routine that you supply on the PDF print panels. As a system programmer, you can modify those print panels so that the print routines receive more or less information.

If variable ZPRLGLST is not defined or does not have a value of LOG or LIST, then the variables shown in Table 36 are available to the print exit routine. However, if variable ZPRLGLST has a value of LOG or LIST, then the variables shown in Table 37 on page 194 are available to the print exit routine.

*Table 36. Variables available to the print exit routine.*

Variable	Type	Len	Description	Modifiable
ZPRDSN	CHAR	44	Fully qualified data set name (no quotes) of the data set to be printed	No
ZPRMEM	CHAR	8	Member to be printed, if the data set is a partitioned data set and a member is defined	No
ZPRPMD	CHAR	5	Print mode - BATCH, LOCAL, or TSO/E ICF	Yes
ZPROPT	CHAR	1	Disposition of the data set as specified by the user - K (keep) or D (delete)	Yes
ZPRDSORG	CHAR	8	Organization of the data set - PO or PS	No
ZPRRECFM	CHAR	6	Record format of the data set	No
ZPRLRECL	CHAR	5	Logical record length of the data set	No
ZPRBLKSZ	CHAR	5	Physical block size of the data set	No
ZPRVOLSE	CHAR	6	Volume serial of the data set	No
ZPRPASS	CHAR	8	Data set password (if it is password protected)	No

Table 36. Variables available to the print exit routine. (continued)

Variable	Type	Len	Description	Modifiable
ZPRYSO	CHAR	80	Batch SYSOUT class as specified by the user	Yes
ZPRYSOL	CHAR	80	Local SYSOUT class as specified by the user	Yes
ZPRLPRT	CHAR	8	Local printer ID as specified by the user	Yes
ZPRICFPL	CHAR	15	TSO/E Information Center Facility printer location as specified by the user	Yes
ZPRICFPF	CHAR	8	TSO/E Information Center Facility printer format as specified by the user	Yes
ZPRICFPT	CHAR	8	TSO/E Information Center Facility printer type as specified by the user	Yes
ZPRICFNC	CHAR	3	TSO/E Information Center Facility number of copies as specified by the user	Yes
ZPRICFEF	CHAR	1	TSO/E Information Center Facility enable font selection	Yes
ZPRJB1	CHAR	72	Print job card 1 as specified by the user	Yes
ZPRJB2	CHAR	72	Print job card 2 as specified by the user	Yes
ZPRJB3	CHAR	72	Print job card 3 as specified by the user	Yes
ZPRJB4	CHAR	72	Print job card 4 as specified by the user	Yes

## Return codes

- 0 Successful termination of print routine. PDF does no further processing.
- 2 Successful termination of print routine. PDF submits any generated JCL.
- 4 Successful termination of print routine. PDF prints the data according to the information in the dialog variables located in the ISPF shared pool. If no message is waiting to be displayed, PDF issues a message.
- 20 Installation print routine failed. PDF keeps the data and, if the installation print exit has issued an ISPF SETMSG service request, ISPF displays the requested message.

**Note:** ISPF processes the data set based on the updated value of the ZPROPT variable for return codes 0, 2, or 4. This allows the exit to delete the data set or instruct ISPF to delete the data set. ZPROPT can not be set to a value of 'D' for PDS or PDSE data sets.

## Error processing

The installation-wide exit routine is responsible for handling all errors that occur while it is in control. The TSO/E Information Center Facility handles all error conditions while it is in control. In addition, PDF displays any error messages generated by the TSO/E Information Center Facility on the current print panel.

---

## Print utility exit on ISPF termination and LOG/LIST commands

In the ISPF configuration table keyword file, this exit is set with keywords PRINT\_UTILITY\_PROGRAM\_EXIT (if your exit is a program) and PRINT\_UTILITY\_COMMAND\_EXIT (if your exit is a CLIST).

The print utility exit lets you define your own print facility to replace or supplement ISPF's print facilities. If you define a print routine for this exit, ISPF calls that routine upon ISPF termination and from ISPF's LOG and LIST commands. The print routine is also called from options 3.6, 3.8, and all 4.x options as described in "Print utility exit" on page 192. To define a print routine create your own JCL for batch jobs or your own local print routine such as PRINTDS. The exit parameters defined in Table 37 that are marked with an asterisk (\*) are system variables. They can be modified in the exit, but they are not saved in the system profile. The variables defined as not modifiable should not be modified.

**Note:** When calling the print utility exit upon ISPF termination and ISPF's LOG and LIST commands, the print log is turned off. The print utility operates without a log.

If the print utility exit is invoked, a new LOG or LIST is allocated if ISPF produces further LOG or LIST data.

The print routine can:

- Define its own print facility and submit the job.
- Specify additional print options.
- Supply print parameters automatically.
- Censor print requests.

## Exit parameters

ISPF communicates with the installation-written print routine through dialog variables in the ISPF shared pool. PDF passes data set, job card, and print option information to the print routine that you supply on the ISPF LOG/LIST termination and command panels.

If variable ZPRLGLST has a value of LOG or LIST, then the variables shown in Table 37 are available to the print exit routine. However, if variable ZPRLGLST is not available or blank, then the variables shown in Table 36 on page 192 are available to the print exit routine.

*Table 37. Print exit routine variables*

Variable	Type	Len	Description	Modifiable
ZPRLGLST	CHAR	4	Value is LOG if processing the LOG, LIST if processing the LIST.	No
ZPRLGDSN	CHAR	44	Fully qualified log data set name (no quotes) of the data set to print.	No
ZPRLSDSN	CHAR	44	Fully qualified list data set name (no quotes) of the data set to print.	No
ZPRLGPMD	CHAR	5	Log print mode, BATCH or LOCAL.	Yes
ZPRLSPMD	CHAR	5	List print mode, BATCH or LOCAL.	Yes
*ZPRLGOPT	CHAR	1	Disposition of the log data set as specified by the user D (delete)	Yes
*ZPRLSOPT	CHAR	1	Disposition of the list data set as specified by the user D (delete)	Yes
*ZPRRECFM	CHAR	6	Record format of the list data set.	No
*ZPRLRECL	CHAR	5	Logical record length of the list data set.	No

Table 37. Print exit routine variables (continued)

Variable	Type	Len	Description	Modifiable
*ZPRLGSYS	CHAR	15	Batch SYSOUT class for the log data set as specified by the user.	Yes
ZPRLGSYSL	CHAR	15	Local SYSOUT class for the log data set as specified by the user	Yes
*ZPRLSSYS	CHAR	15	Batch SYSOUT class for the list data set as specified by the user	Yes
ZPRLSSYL	CHAR	15	Local SYSOUT class for the list data set as specified by the user	Yes
*ZPRLGPRT	CHAR	17	Local printer ID for the log data set as specified by the user.	Yes
*ZPRLSPRT	CHAR	17	Local printer ID for the list data set as specified by the user.	Yes
*ZPRJB1	CHAR	72	Print job card 1 as specified by the user	Yes
*ZPRJB2	CHAR	72	Print job card 2 as specified by the user	Yes
*ZPRJB3	CHAR	72	Print job card 3 as specified by the user	Yes
*ZPRJB4	CHAR	72	Print job card 4 as specified by the user	Yes
ZPRMSGF	CHAR	1	Data set disposition message suppression. A value of 'S' stops ISPF setting a final disposition message.	Yes

**Note:** The user can only specify the disposition of the data set as D (delete). The exit can change the disposition of the data set to K (keep).

## Return codes

When the print exit routine is invoked upon ISPF termination or ISPF LOG or LIST commands these return codes can occur.

**0** Successful termination of print routine.

**Batch** Exit provides job card and other JCL needed to print the job. Exit submits the job.

**Local** Exit provides necessary information to print the job. Exit issues the print command.

ISPF issues a message on return from the print utility exit, indicating the return code from the exit.

**2** Successful termination of the print routine.

Exit provides job card and JCL needed to print the job. ISPF submits the JCL contained in the file pointed to by the ZTEMPF variable.

ISPF issues a message on return from the print utility exit, indicating the return code from the exit.

**4** Successful termination of the print routine.

**Batch** Exit may alter the modifiable variables and place them in ISPF's shared pool. ISPF submits the print job using the information in the dialog variables located in the ISPF shared pool.

**Local** Exit may alter the modifiable variables and place them in ISPF's

shared pool. ISPF issues the PRINTDS command using the information in the dialog variables located in the ISPF shared pool.

ISPF issues a message on return from the print utility exit, indicating the return code from the exit.

**Note:**

| ISPF processes the data set based on the updated value of the  
| disposition variable (ZPRLGOPT or ZPRLSOPT) for return codes 0,  
| 2, or 4. This allows the exit to either delete the data set itself, or  
| instruct ISPF to delete the data set. ISPF does not delete  
| pre-allocated LOG or LIST data sets.

| For return codes 0, 2, and 4, ISPF issues a message regarding the  
| disposition of the data set. The message is based on the dialog  
| variables ZPRLGOPT, ZPRLSOPT, ZPRLGPRT, and ZPRLSPRT  
| located in the ISPF shared pool. The exit can suppress this message  
| by setting variable ZPRMSGF to a value of 'S'.

20 The print routine failed. ISPF keeps the data and displays an error message.

## Error processing

The exit routine is responsible for handling all errors that occur while it is in control.

---

## Compress request exit

In the ISPF configuration table keyword file, this exit is set with the keywords COMPRESS\_UTILITY\_PROGRAM\_EXIT (if your exit is a program) and COMPRESS\_UTILITY\_CLIST\_EXIT (if your exit is a CLIST).

Instead of using the IEBCOPY system utility interface that PDF provides, you can specify that PDF use an installation-written exit routine to handle requests to compress partitioned data sets under options 3.1, 3.4, and the LMCOMP service. When the compress exit gets control, the data set will have already been allocated and enqueued as exclusive. The exit will have to re-allocate the data set to a specific ddname. As part of termination processing, the exit must free the allocated ddname. The FREE operation must not be done by data set name. PDF allows the exit routine to:

- Use an alternate compression technique
- Provide backup before allowing normal PDF compression
- Selectively prevent compression.

PDF calls the exit routine with the SELECT service. The routine can be either a program or CLIST. All ISPF and system services are available.

## Exit parameters

PDF communicates with the exit routine through variables in the shared pool. PDF considers these variables to be read-only and ignores any changes the exit routines make to the parameters. A VERASE is done at the end of the compress to delete the variables from all variable pools.

The variables available to the compress exit routine are:

Variable	Type	Len	Description
ZCMPDSN	CHAR	44	Fully qualified data set name with no quotes
ZCMPVOL	CHAR	6	Volume serial
ZCMPPSWD	CHAR	8	Data set password
ZCMPORIG	CHAR	8	Origin of the call.
			<b>LMCOMP</b> The compress service (LMCOMP) called the exit routine.
			<b>OPTION31</b> The compress exit routine is being called from option 3.1, the compress utility.
			<b>OPTION34</b> The compress exit routine is being called from option 3.4, the data set list utility.

## Return codes

0	Data set successfully compressed
2	PDF should compress the data set
4	Data set is not eligible for compression
8	Data set could not be allocated
12	Data set not cataloged
16	Error in exit; PDF should continue processing
17	LMCOMP recursion error
18	Unknown return code from exit
19	Error encountered invoking exit
20	Severe error; PDF should not continue processing
Other	PDF treats the error as return code 16.

## Error processing

The exit routine is responsible for handling all errors that occur while it is in control. Failure to use correct allocation, serialization, and deallocation causes unpredictable results.

---

## Data set name change exit

In the ISPF configuration table keyword file, this exit is set with keyword `DATA_SET_NAME_CHANGE_PROGRAM_EXIT`.

You can use the data set name change exit to change the name of the data set entered on the data entry panel. This allows PDF to process using the new data set name without the variables originally entered on the panel being changed. Anywhere the data set name is displayed, other than the data entry panel, can reflect either the original or the changed data set name. The data entry fields will not reflect any changes made to the data set name. If the exit is not specified, all data set names are used exactly as they are entered.

PDF calls this exit routine using the standard conventions. The exit is invoked for every data set specified on a data entry panel or in a PDF service call at the point in the PDF processing flow when the data set name is built from its component

parts. This includes the library access services, EDIT and BROWSE services, and the PDF product dialogs themselves. The routine must be a program. All ISPF, PDF and system services are available to it.

If more than one ISPF library is specified on a data entry panel, the exit is invoked once for each library specified before each library is allocated. The libraries are not concatenated until each library is processed by the exit and allocated.

## Exit parameters

PDF passes the data set name information as it was entered on the panel to the exit routine. Register 1 points to the parameter list.

PDF uses these parameters to communicate with the data set name change exit:

Variable	Type	Len	Description
PROJECT	CHAR	8	Project name as it was entered on the panel
GROUP	CHAR	8	Group name as it was entered on the panel
TYPE	CHAR	8	Type name as it was entered on the panel
MEMBER	CHAR	8	Member name as it was entered on the panel
OTHER DATA SET	CHAR	56	On input to the exit, the other data set name as entered on the panel. On output from the exit, blank or the data set name to be used. This can be used to replace an ISPF library.
REASON	CHAR	8	Reason the data set is being allocated:

### RECOVERY

The data set is the edit recovery data set. The Other Data Set field contains the fully qualified data set name. The data set name change exit is not called for edit recovery data sets which are listed in the edit recovery table with an associated disposition of 'K'.

**TEMP** The data set is a temporary PDF data set. This includes:

- The data set name specified on the member list SAVE command
- The data set name specified on the data set list SAVE command
- The data set used during the outlist utility processing (ISPF option 3.8, when using program ISRUOLP instead of CLIST ISRUOL).

### blank

- Any data set entered on an PDF data entry panel
- Data sets specified on the LMINIT service
- Data sets specified on any service that does not need an LMINIT to have been previously performed.

---

Any of the parameters can be modified and are picked up by PDF with the exception of the Reason field, which is for the information of the exit only. Both an ISPF library (a PROJECT/GROUP/TYPE combination) and an OTHER DATA SET

can be present in the parameter list. In this case, the OTHER DATA SET name supersedes the ISPF library name. If an ISPF library combination is specified as input to the exit and the exit wishes to create a new name that does not fit the ISPF library naming convention, the ISPF library name can be replaced by the exit filling in the Other Data Set field.

After the data set name has been changed by the exit, PDF does its normal data set processing, such as appending the user's prefix if an OTHER DATA SET name without quotes is returned by the exit. The changed data set name values are not saved in any variable pool, but are saved internally by PDF. The original input fields on the data entry panel are not changed, but any title line on a panel that contains a data set name can reflect either the original or the changed data set name.

## Usage notes

1. Those data sets marked with a reason of TEMP can go through the exit more than once. The exit should not blindly add qualifiers to the TEMP data set names.
2. Any sequential data set that is used for output, such as being edited, the target of a Move/Copy (option 3.3), the source of an Edit MOVE command or the target of an Edit REPLACE command, or being reallocated by Edit after a space ABEND (B37), is sent through the exit twice. In addition, if a member is being browsed via the Library utility browse line command (browse under option 3.1), the data set is sent through the exit twice.
3. Data sets that are allocated in a CLIST as well as a program or that have their data set names built in a CLIST or skeleton cannot be modified by the data set name exit unless the CLIST or skeleton has also been modified to change the data set name in the same way that the exit does. These data sets include:
  - The list data set name produced by the language processors in Foreground and Batch, and the term data sets in Batch.
  - The input and output data sets for SuperC. The CLISTs to be modified are ISRSFORG and ISRSSRCH.
  - The data set created by the Outlist utility (option 3.8) if CLIST ISRUOL is used rather than program ISRUOLP.
4. If the data set is being allocated for recovery, the data set name change exit should check the recovery table and generate a unique data set name.

---

## Member list filter exit

In the ISPF configuration table keyword file, this exit is set with keyword MEMBER\_LIST\_FILTER\_PROGRAM\_EXIT.

The member list filter exit provides you with two capabilities:

1. You can dictate which members of a partitioned data set or concatenation of partitioned data sets are to be included in a member list when it is created.
2. You can specify which members of a member list are to be selected when the SELECT primary command is issued.

If a member list filter exit routine is not defined, PDF uses its default pattern matching conventions to determine which members are displayed, and which members are selected if the SELECT primary command is issued.

For each case in which the member list filter exit routine would be invoked, it is called first to allow verification/modification of the specified pattern, and then once for each member in the data set that matches the pattern. The exit routine, through return codes, dictates which members are to be included in the member list, or selected if the SELECT primary command is issued.

PDF invokes the member list filter exit routine using standard linkage conventions. The exit routine must be a program.

From the exit routine, any of the PDF or ISPF services can be invoked. However, be careful when invoking services that generate a member list as part of their internal processing (LMMLIST, LMMDISP, LMMOVE, LMPROM, LMPRINT, and LMMSTATS). These invocations would result in a recursive call to the exit routine.

## Analyze patterns

### Exit parameters

PDF uses these parameters to communicate with the exit routine when it is invoked to verify/modify the pattern that you entered either on a member list entry panel, as a parameter to an PDF service, or with the SELECT primary command.

Parameter	Type	Len	Description	Modifiable
CODE	FIXED	4	Code that indicates that this is the analyze pattern call. Call=1.	No
PATTERN	CHAR	8	Pattern as entered by the user	Yes

### Return codes

These return codes are expected from the member list filter exit routing to indicate the described conditions.

- 0 The pattern that you entered generates the member list.
- 4 The pattern is updated by the member list exit program and generates the member list.
- 8 The member list request or SELECT command request is canceled and a conditional SETMSG is issued.

## Analyze member names

### Exit parameters

PDF uses these parameters to communicate with the exit routine when the routine determines which members that matched the pattern are to be included in the member list, or which members are to be selected if a SELECT command was issued.

Parameter	Type	Len	Description	Modifiable
CODE	FIXED	4	Code indicating that this is the analyze member name call. Call=2.	No
MEMNAME	CHAR	8	Member name.	No

### Return codes

These return codes are expected from the member list filter exit program to indicate the described conditions.

- 0 Include this member in the member list.
- 4 Do not include this member in the member list.
- 8 The list is stopped; no more items are added to the list.

---

## Member list built-in line command exit

In the ISPF configuration table keyword file, this exit is set with keywords MEMBER\_LIST\_LINE\_COMMAND\_PROGRAM\_EXIT and MEMBER\_LIST\_LINE\_COMMAND\_COMMAND\_EXIT.

The member list line command exit enables installations to change or restrict the behavior of the built-in line commands entered in a PDF member list. This exit is invoked for single-character member list built-in line commands invoked from ISPF option 3.1, ISPF option 3.4, and the ISPF Workplace (option 11).

This exit can be used to modify the behavior of built-in line commands. For example, when the E (edit) command is entered next to a member, the exit might examine the contents of the member and invoke a program other than the ISPF editor to process that member. If the exit then determines that the ISPF editor does not need to be invoked, it must return a return code of 8.

## Exit parameters

PDF uses these parameters to communicate with the line command exit routine. These parameters are passed as a single 65-character string. A single blank is placed between parameters.

Parameter	Type	Len	Description	Modifiable
LINE COMMAND	CHAR	1	A 1-character field containing the data set list line command.	No
DATA SET NAME AND MEMBER	CHAR	56	A 56-character field containing data set name and member against which the line command is run. The format of this field is: 'DATA.SET.NAME(MEMBER)'. The field is left-justified with trailing blanks.	No
VOLUME SERIAL	CHAR	6	A 6-character field containing the volume on which the data set specified in DATA SET NAME AND MEMBER can be found.	No

## Usage notes

- The name of the module is placed into the ISPF configuration table when the product is installed.
- 31-bit addressing must be supported by the exit routine as it is given control in AMODE 31. PDF does not restrict the RMODE of the exit, but RMODE ANY is recommended.
- For the program version of the exit, standard OS linkage conventions are used to branch to the exit routine that is defined.
- For the command version of the exit, the parameters are passed as operands.
- You can make PDF service calls from within the exit routine.
- You cannot activate or deactivate the exit while a PDF session is in progress. You can make changes to the exit routine load module at any time, but because PDF loads the module only once at session initialization, the changes are not recognized until the next ISPF session.

- If the exit name has been specified in the ISPF configuration table and the member list was generated from ISPF option 3.1, 3.4, or 11, the exit is called before each built-in line command is processed. The exit can analyze the line command and determine whether you should invoke the command as entered, or not at all.

## **Error processing**

If a nonzero return code is returned from an exit call, a message is conditionally set.

## **Return codes**

- 0      Use line command as specified by the user
- 8      Do not invoke this built-in command
- 20     Severe error occurred.

\_\_\_\_\_ **End of Programming Interface information** \_\_\_\_\_

---

## Chapter 9. ISPF Configuration Table keywords and values

ISPF Configuration Table keywords and values explains the keywords in the ISPF Configuration Table and their allowable and default values.

---

### PDF exits

These fields specify the name of the program or command exit to be invoked at each exit point. COMMAND exits can be either CLIST or REXX. Exit names can have a maximum length of 8 characters. If both a program exit and a command exit are specified for the same exit point, the program exit is used.

#### **DATA\_SET\_ALLOCATION\_PROGRAM\_EXIT**

The program to be invoked as the data set allocation exit. This should be the name of a load module in your standard MVS search sequence.

#### **PRINT\_UTILITY\_PROGRAM\_EXIT**

The program to be invoked as the print utility exit. This should be the name of a load module in your standard MVS search sequence.

#### **PRINT\_UTILITY\_COMMAND\_EXIT**

The command to be invoked as the print utility exit. This should be the name of a member in your SYSPROC or SYSEXEC allocation.

#### **COMPRESS\_UTILITY\_PROGRAM\_EXIT**

The program to be invoked as the compress utility exit. This should be the name of a load module in your standard MVS search sequence.

#### **COMPRESS\_UTILITY\_CLIST\_EXIT**

The command to be invoked as the compress utility exit. This should be the name of a member in your SYSPROC or SYSEXEC allocation.

#### **DATA\_SET\_LIST\_FILTER\_PROGRAM\_EXIT**

The program to be invoked as the data set list filter exit. This should be the name of a load module in your standard MVS search sequence.

#### **MEMBER\_LIST\_FILTER\_PROGRAM\_EXIT**

The program to be invoked as the member list filter exit. This should be the name of a load module in your standard MVS search sequence.

#### **DATA\_SET\_NAME\_CHANGE\_PROGRAM\_EXIT**

The program to be invoked as the data set name change exit. This should be the name of a load module in your standard MVS search sequence.

#### **DATA\_SET\_LIST\_LINE\_COMMAND\_PROGRAM\_EXIT**

The program to be invoked as the data set list line command exit. This should be the name of a load module in your standard MVS search sequence.

#### **ACTIVITY\_MONITORING\_PROGRAM\_EXIT**

The program to be invoked as the activity monitoring exit. This should be the name of a load module in your standard MVS search sequence.

#### **MEMBER\_LIST\_LINE\_COMMAND\_PROGRAM\_EXIT**

The program to be invoked as the member list line command exit. This should be the name of a load module in your standard MVS search sequence.

## MEMBER\_LIST\_LINE\_COMMAND\_COMMAND\_EXIT

The command to be invoked as the member list line command exit. This should be the name of a member in your SYSPROC or SYSEXEC allocation.

---

## Data set allocation settings

### PDF\_DEFAULT\_UNIT

The unit name used by PDF when allocating work data sets. This value is used any time PDF needs to allocate a new data set on behalf of the user with the exception of option 3.2, edit recovery and ISPF work, control, and list data sets (ISPWRKx, ISPCTLx, and ISPLSTx). These new data sets may be temporary or permanent depending on the option of PDF being used.

For improved performance it is recommended that the VIO=YES option be added to the UNITNAME macro for the unit you specify in this field, but a VIO-only unit name is not recommended. Several of the ISPF options (including the Move/Copy utility and Outlist utility) will not function with VIO data sets.

The default is SYSALLDA.

### ALLOWED\_ALLOCATION\_UNITS

This field controls which unit names are eligible to a user when the user is creating a data set through option 3.2. Valid values are:

- |                  |   |
|------------------|---|
| <b>ANY</b>       | Any unit may be used  |
| <b>UADS</b>      | Indicates the UNIT parameter in the users UADS entry should control the unit used |
| <b>unit-name</b> | Indicates that specific unit should be used                                       |

The default is ANY.

### ALLOCATE\_BEFORE\_UNCATALOG

Indicates whether data sets to be uncataloged should first be allocated to accommodate those security packages the process during allocation. Valid values are YES or NO.

The default is NO.

### VERIFY\_EXPIRATION\_DATE

Should expiration dates entered in option 3.2 when creating a data set be validated to ensure they are not in the past. Valid values are YES or NO.

The default is YES.

### VOLUME\_OF\_MIGRATED\_DATA\_SETS

The volume name that indicates a data set is migrated.

The default is MIGRAT.

### COMMAND\_TO\_DELETE\_MIGRATED\_DATA\_SETS

The command that should be invoked when the D line command is used in option 3.4 to delete a migrated data set.

The default is HDELETE.

---

## Outlist data set specifications

### **OUTLIST\_RECORD\_LENGTH**

Record length of the temporary data set used by the Outlist utility.

The default is 133.

### **OUTLIST\_BLOCK\_SIZE**

Block Size of the temporary data set used by the Outlist utility. Block size should be an even multiple of the record length unless a zero is specified for system determined block size.

The default is 13566.

### **OUTLIST\_PRIMARY\_QUANTITY**

Primary number of tracks to be allocated for the Outlist Utility.

The default is 200.

### **OUTLIST\_SECONDARY\_QUANTITY**

Secondary number of tracks to be allocated for the Outlist Utility.

The default is 100.

---

## SuperC data set specifications

### **SUPERC\_LIST\_DATA\_SET\_BLOCK\_SIZE**

The block size for the SuperC listing data set. The list data set is a record format FBA, record length 133 data set. The block size should be an even multiple of 133 unless 0 is specified for system determined block size.

The default is 0.

This field is only used when the USE\_SUPER\_C\_PROGRAM\_INTERFACE field is set to YES.

### **SUPERC\_UPDATE\_DATA\_SET\_BLOCK\_SIZE**

The block size for the SuperC update data set. The update data set is a record format FB, record length 80 data set. The block size should be an even multiple of 80 unless 0 is specified for system determined block size.

The default is 0.

This field is only used when the USE\_SUPER\_C\_PROGRAM\_INTERFACE field is set to YES.

### **SUPERC\_PROFILE\_DATA\_SET\_BLOCK\_SIZE**

The block size for the SuperC profile data set. The profile data set is a record format FB, record length 80 data set. The block size should be an even multiple of 80 unless 0 is specified for system determined block size.

The default is 0.

This field is only used when the USE\_SUPER\_C\_PROGRAM\_INTERFACE field is set to YES.

### **SUPERC\_STATEMENTS\_DATA\_SET\_BLOCK\_SIZE**

The block size for the SuperC statements data set. The statements data set is a record format FB, record length 80 data set. The block size should be an even multiple of 80 unless 0 is specified for system determined block size.

The default is 0.

This field is only used when the USE\_SUPER\_C\_PROGRAM\_INTERFACE field is set to YES.

#### **USE\_SUPER\_C\_PROGRAM\_INTERFACE**

Should SuperC be invoked directly from ISPF rather than invoked via clists ISRSFORG OR ISRSSRCH. Specifying YES in this field will improve the performance of the SuperC interface.

The default is YES.

#### **SUPER\_C\_LISTING\_PRIMARY\_QUANTITY**

Primary number of blocks for the SuperC Listing data set.

The default is 50.

This field is only used when the USE\_SUPER\_C\_PROGRAM\_INTERFACE field is set to YES.

#### **SUPER\_C\_LISTING\_SECONDARY\_QUANTITY**

Secondary number of blocks for the SuperC Listing data set.

The default is 100.

This field is only used when the USE\_SUPER\_C\_PROGRAM\_INTERFACE field is set to YES.

#### **SUPER\_C\_UPDATE\_PRIMARY\_QUANTITY**

Primary number of blocks for the SuperC Update data set.

The default is 15.

This field is only used when the USE\_SUPER\_C\_PROGRAM\_INTERFACE field is set to YES.

#### **SUPER\_C\_UPDATE\_SECONDARY\_QUANTITY**

Secondary number of blocks for the SuperC Update data set.

The default is 30.

This field is only used when the USE\_SUPER\_C\_PROGRAM\_INTERFACE field is set to YES.

---

## **LMF**

#### **FAIL\_ON\_LMF\_LOCK**

The default is YES.

---

## **Edit recovery data set specifications**

#### **EDIT\_RECOVERY\_BLOCK\_SIZE**

The block size for the edit recovery data set. This data set is a record format U, record length 0 data set. System determined block size is not supported for this field.

The default is 13680, the minimum allowed value is 3120.

#### **EDIT\_RECOVERY\_PRIMARY\_QUANTITY**

The primary number of blocks that should be allocated for the edit recovery data set.

The default is 40.

#### **EDIT\_RECOVERY\_SECONDARY\_QUANTITY**

The secondary number of blocks that should be allocated for the edit recovery data set.

The default is 200.

---

## Move/Copy settings

### MAXIMUM\_GOOD\_IEBCOPY\_RETURN\_CODE

Indicates the maximum return code from IEBCOPY that will allow Move/Copy processing to continue. Any return code higher than this value will be considered an error.

The default is 0.

### USE\_IEBCOPY\_COPY\_OR\_COPYMOD\_OPTION

Should ISPF use COPY or COPYMOD when invoking IEBCOPY to process load modules. Valid values are:

- 1 Use COPY if the target library block size is the same or greater than the source library block size, COPYMOD if the target block size is smaller.
- 2 Use COPY if the target library block size and source library block size are the same, COPYMOD if they are different.
- 3 Always use COPYMOD.

The default is 2.

### WHEN\_TO\_USE\_IEBCOPY

When should ISPF use IEBCOPY instead of a read/write loop to process load modules. Valid values are:

- 0 Only use IEBCOPY when processing a PDSE, or when copying from a larger block size to a smaller block size and COPYMOD was requested (see "USE\_IEBCOPY\_COPY\_OR\_COPYMOD\_OPTION").
- 1 Always use IEBCOPY for load modules.
- 2 Only use IEBCOPY for PDSEs.

The default is 0.

### ALLOW\_DATA\_SET\_CREATION\_FOR\_MOVE\_COPY

If the target data set for Move/Copy does not exist, should the data set be created for the user. The user can specify either that the data set be created with the same characteristics as the original, or can specify the characteristics for the new data set. Valid values are YES and NO.

The default is YES.

---

## DSLISL removable media interface settings

The keywords DSLISL\_RM\_ENABLED, DSLISL\_RM\_COMMAND, and DSLISL\_RM\_APPLID define an external interface to DFSMSrmm or an equivalent product. This can be used to allow the Data Set List utility (ISPF Option 3.4) to process selected line commands for data sets stored on tape or some other removable media.

### Return codes

The Data Set List utility handles return codes from the removable media interface command as follows:

- 0 or 4** Normal completion. Data Set List utility continues processing.

- 8 Error. Data Set List utility stops processing. Any further line commands are not processed. No error message is issued, allowing the tape/removable media interface to issue an error message via the ISPF SETMSG service.
- 12 Error. Data Set List utility issues an error message and stops processing. Any further line commands are not processed.
- Other** Severe Error. Data Set List utility stops processing. Any further line commands are not processed. If the command was not found, message ISPD223 is issued, otherwise message ISRU671 is issued.

The command specified by DSLIST\_RM\_COMMAND should set the return code to 12 for any line commands that are not supported by the interface.

#### DSLIST\_RM\_ENABLED

Controls whether support is enabled in ISPF option 3.4 to call the removable media interface for these line commands: I, S, D, R, C, M, P, X, CO, MO. Valid values are YES and NO.

The default is NO.

#### DSLIST\_RM\_COMMAND

Specifies a command name, including parameters, to be invoked using the ISPF SELECT CMD service. The command name can be up to 45 characters in length, and may contain 2 special characters:

- ? Is replaced with the option 3.4 line command. Two consecutive '?' characters will be replaced with a single '?' and no substitution of the line command will be performed. If '?' is not included within the command, the line command is automatically appended as a parameter to the end of the command.
- / Is replaced with the data set name (ZDLDSN) and the first volume (ZDLVOL), separated by a space. Two consecutive '/' characters will be replaced with a single '/' and no substitution of the data set name and first volume will be performed. Where the '/' is not included within the command, both the data set name and first volume will be automatically appended as a parameters to the end of the command. If a '%' character is specified before the command, the ISPF SELECT CMD service will bypass any attempt to load the command as a load module.

The default value for command is %EDGRPD34, which is equivalent to %EDGRPD34 ? /.

#### Examples

These examples demonstrate different ways of specifying the DSLIST\_RM\_COMMAND parameter:

```
tapecmd ? /
tapecmd OPT(?) /
tapecmd OPT(?) DSV(/) ALTLIB(YES)
```

#### DSLIST\_RM\_APPLID

The syntax for the DSLIST\_RM\_APPLID keyword is:

```
DSLIST_RM_ APPLID = applid | NONE
```

Where *applid* is the application ID used to invoke the tape interface command defined by DSLIST\_RM\_COMMAND. *Applid* is a 1-character to

4-character name. If *applid* is set to NONE, no application ID will be used. Where the *applid* is specified, the PASSLIB parameter will also be included on the ISPF SELECT CMD service.

The default is EDG.

---

## Edit-related settings

### MAXIMUM\_EDIT\_PROFILES

The maximum number of ISPF Edit profiles. If the number of profiles exceeds this number the least recently used unlocked profile is deleted from the profile table. The value must be between 1 and 255.

The default is 25.

### SCLM\_WARNING\_LEVEL

Indicates the level of SCLM checking that should be done when SCLM-controlled members are processed outside SCLM. Valid values are:

<b>NONE</b>	No checking is done. SCLM-controlled members may be edited and processed outside SCLM.
<b>WARN</b>	If an SCLM-controlled member is processed by Edit or Reset Statistics, a message is displayed to warn the user that the SCLM accounting data will be invalidated by the pending request.
<b>ERROR</b>	When Edit of an SCLM member is attempted, an error message is displayed and the edit is denied.

The default is WARN.

### UNDO\_STORAGE\_SIZE

The maximum amount of kilobytes of storage available to the edit UNDO command to be used for keeping a history of edit changes. A minimum value of 1024 (1024 KB) is recommended. If this value is 0 then UNDO will be available from the edit recovery data set only. If the field is not 0 but is less than 128, a value of 128 will be used. This value will be rounded down to the nearest multiple of 64 (64 KB).

The default is 0.

**Note:** Use of storage for saving the record of changes provides better response time for individual users, but may have a slight detrimental effect on overall system performance.

### ALLOW\_EDIT\_HIGHLIGHTING

Should ISPF Edit highlighting be available to all users and applications. Valid values are YES or NO. A value of NO disables Edit highlighting for all applications, a value of YES enables Edit highlighting for any dialog that uses a panel enabled for highlighting.

The default is YES.

### DEFAULT\_EDIT\_DISPLAY

This field determines how the editor appears when it is invoked via ISPF either interactively, or via an Edit service call that does not specify a user edit panel. It controls the availability of action bars and edit highlighting. Valid values are:

<b>0</b>	No actions bars should be displayed, and Edit highlighting should not be available.
----------	---

- 1 Actions bars should be displayed, but Edit highlighting should not be available.
- 2 No actions bars should be displayed, but Edit highlighting should be available.
- 3 Actions bars should be displayed, and Edit highlighting should be available.

The default is 3.

#### **MAXIMUM\_STORAGE\_ALLOWED\_FOR\_EDIT**

The maximum number of kilobytes of storage that the editor can use when initially reading in data. If the initial read of the data requires more storage than this value, browse will be substituted instead. To allow edit to use as much storage as is available, set the value to 0.

The default is 0.

#### **ENABLE\_ASSEMBLER\_CONTINUATION\_ERRORS**

Enable the use of reverse video pink to highlight Assembler continuations which start before column 16. Set this value to NO if your site uses a different start column for Assembler continuation (via the ICTL Assembler instruction) or if assembler highlighting is used for data other than Assembler programs (such as SCLM architecture definitions).

The default is YES.

#### **WARN\_ON\_TRUNCATION\_OF\_TRAILING\_BLANKS**

Should a warning message be displayed if a user edits variable data (record format = V) with one or more records that ends in a blank. The editor will truncate these blanks when the edit data is saved unless the editor is told to preserve the blanks.

Blanks will be preserved if:

- The "Preserve VB record length" field on the Edit Entry panel is selected.
- The PRESERVE keyword is specified on the EDIT service invocation.
- The PRESERVE ON Edit command is entered.

The default is YES.

#### **SITE\_WIDE\_INITIAL\_MACRO**

Site-wide Edit initial macro. The macro specified here will be run before any user-specified macros. This can allow you to alter or disallow edit sessions. You may want to use a macro that does a PROFILE RESET to force all new profiles to use the settings in this configuration table.

The default is NONE (no macro).

#### **TEXT\_FLOW\_TERMINATORS**

What characters should cause the edit text flow function (line command TF, edit macro command TFLOW) to stop processing. These characters generally indicate a new paragraph or section of a document.

The default characters are:

.:&<

#### **EDIT\_CUT\_DEFAULT**

Valid values are REPLACE and APPEND. Selecting REPLACE means that the cut information replaces whatever information is already in the clipboard. Selecting APPEND means that cut information is added at the end of existing information in the clipboard.

The default is REPLACE.

#### **EDIT\_PASTE\_DEFAULT**

Valid values are KEEP and DELETE. Selecting KEEP means that the information remains in the clipboard even after pasting it into a separate file. Selecting DELETE means that the information is deleted from the clipboard after it has been pasted into a separate file.

The default is KEEP.

#### **ALLOW\_DATA\_SET\_CREATION\_FOR\_CREATE\_REPLACE**

If the target data set for the Edit CREATE or REPLACE command does not exist, should the data set be created for the user. The user can specify either that the data set be created with the same characteristics as the original, or can specify the characteristics for the new data set. Valid values are YES and NO.

The default is YES.

#### **FORCE\_ISRE776\_FOR\_RCHANGE**

Ensures that when RCHANGE is issued from a PF key, it does not try to process input from the command line. In this case RCHANGE will treat anything that you type on the command line as an invalid parameter and will return an error message ISRE776.

This keyword sets a site default for the EDITSET option "Force ISRE776 if RCHANGE passed arguments". Valid values are YES and NO.

The default is NO.

#### **FORCE\_PRESERVE\_VB\_RECORD\_LENGTH**

Forces the users' Preserve VB Record Length setting in Edit to be the value selected in the PRESERVE\_VB\_RECORD\_LENGTH field. Valid values are YES and NO.

The default is NO.

#### **PRESERVE\_VB\_RECORD\_LENGTH**

The Preserve VB Record Length option in Edit will be selected. This option will cause the editor to save trailing blanks for variable-length files. Valid values are YES and NO.

The default is NO.

#### **MAXIMUM\_NUMBER\_OF\_EDIT\_CLIPBOARDS**

The maximum number of Edit clipboards allowed. Edit clipboards are used by the Edit CUT and PASTE commands and are kept in data spaces that last the life of the TSO session. Data spaces are allocated by the CUT command and released by the PASTE command, and are paged out when not in use. This can be a number between 1 and 11.

The default is 11.

#### **MAXIMUM\_EDIT\_CLIPBOARD\_SIZE**

The maximum size for the Edit clipboards, in 4K increments. A value of 1 would mean 4 kilobytes. A value of 0 indicates that the IBM default data space size (239 4K blocks) of the value set through IEFUSI should be used.

The default is 0.

#### **VSAM\_EDIT\_ENABLED**

Is editing of VSAM data set enabled on the system. ISPF Edit will invoke the command in the VSAM\_EDIT\_COMMAND field for any VSAM data

sets specified unless restricted (see “VSAM\_RESTRICTED\_EDIT\_DATASET” on page 235). Valid values are YES and NO.

The default is NO.

#### **VSAM\_EDIT\_COMMAND**

The command to be invoked when a VSAM data set is specified to ISPF Edit. A slash (/) may be used to specify the data set name specified. The maximum length is 50 characters.

The default is **FMNMAIN DSE /**

#### **VSAM\_EDIT\_LIMITED**

Should users be restricted from editing certain VSAM data sets? If this field is set to YES, the VSAM\_RESTRICTED\_EDIT\_DATASET field must be used to specify the restricted data sets. Valid values are YES and NO.

The default is NO.

#### **VSAM\_BROWSE\_ENABLED**

Is browsing of VSAM data set enabled on the system. ISPF Browse will invoke the command in the VSAM\_BROWSE\_COMMAND field for any VSAM data sets specified unless restricted (see “VSAM\_RESTRICTED\_BROWSE\_DATASET” on page 235). Valid values are YES and NO.

The default is NO.

#### **VSAM\_BROWSE\_COMMAND**

The command to be invoked when a VSAM data set is specified to ISPF Browse. A slash (/) may be used to specify the data set name specified. The maximum length is 50 characters.

The default is **FMNMAIN DSB /**

#### **VSAM\_VIEW\_ENABLED**

Is viewing of VSAM data set enabled on the system. ISPF View will invoke the command in the VSAM\_VIEW\_COMMAND field for any VSAM data sets specified unless restricted (see “VSAM\_RESTRICTED\_VIEW\_DATASET” on page 235). Valid values are YES and NO.

The default is NO.

#### **VSAM\_VIEW\_COMMAND**

The command to be invoked when a VSAM data set is specified to ISPF View. A slash (/) may be used to specify the data set name specified. The maximum length is 50 characters.

The default is **FMNMAIN DSB /**

---

## **Edit site-wide profile customizations**

These fields set the defaults for new Edit Profiles created by any user. For information about the values for each field, refer to the *z/OS ISPF Edit and Edit Macros*. These processing rules apply:

- Some items in this section of the ISPF Configuration table can be *forced*. The default for all of the FORCE fields is NO. To force users to use the value specified, change the FORCE field value to YES. If a user then attempts to change one of the forced settings from within the editor an error message is displayed.

- If the user has a ZDEFAULT Edit profile or if a ZEDFAULT profile exists in the ISPTLIB concatenation, then these settings have no effect *except* for those options that are forced.
- If a user has no ZDEFAULT profile and no ZEDFAULT profile exists in ISPTLIB these settings are used to create all new profiles. They have no effect on existing profiles.
- The ZDEFAULT profile is no longer automatically created the first time a user enters the editor, as it was in releases of ISPF before OS/390 Release 5.0. Profiles used with previous versions might contain a ZDEFAULT profile, and it is honored. Users can also specifically create a ZEDFAULT profile so that they can establish their own defaults.

#### **STATS**

Control whether the editor maintains ISPF statistics for PDS members.  
Valid values are ON and OFF.

The default is ON.

#### **FORCE\_STATS**

Force the specified value.

The default is NO.

#### **STATS\_EXT\_ENABLED**

Allows you to use STATS EXT command, extended stats option on reset member statistics panels, and LM services EXT option to cause extended statistics to be created.

The default is NO.

#### **STATS\_EXT**

Control whether the editor maintains ISPF statistics for PDS members.  
Valid values are ON and OFF.

The default is OFF.

#### **FORCE\_STATS\_EXT**

Force the specified value.

The default is NO.

#### **RECOVERY**

Determines whether the editor maintains its recovery file to keep track of the edit session, making it possible for users to recover from system failures. Valid values are ON and OFF.

The default is OFF.

#### **FORCE\_RECOVERY**

Force the specified value.

The default is NO.

#### **RECOVERY\_WARNING\_MESSAGE**

Determines whether the editor should warn users when they enter an edit session with RECOVERY set OFF. Valid values are WARN and NOWARN.

The default is WARN.

#### **FORCE\_RECOVERY\_WARNING\_MESSAGE**

Force the specified value.

The default is NO.

**SETUNDO**

Determines whether the UNDO command will be available in Edit. Valid values are ON and OFF.

The default is ON.

**FORCE\_SETUNDO**

Force the specified value.

The default is NO.

**PACK** Determines whether the editor will save data in packed or unpacked format. Valid values are ON and OFF.

The default is OFF.

**FORCE\_PACK**

Force the specified value.

The default is NO.

**IMACRO**

Specifies the sitewide initial macro to be run for all users.

The default is NONE (no macro).

**FORCE\_IMACRO**

Force the specified value.

The default is NO.

**CAPS** Determines whether the editor automatically rolls text to uppercase. Valid values are ON and OFF.

The default is OFF.

**NOTE** Determines whether the editor displays ==NOTE== lines when the Edit MODEL command is used. Valid values are ON and OFF.

The default is ON.

**HEX** Determines whether the edit data is displayed in hex mode.

- ON
- OFF
- VERT
- DATA

The default is OFF.

**NULLS**

Determines whether trailing spaces on edit data are written to the screen as nulls or blanks.

- STD
- ALL
- OFF

The default is STD.

**DISPLAY\_SEQUENCE\_NUMBERS**

Determines whether the editor will maintain sequence numbers. Valid values are ON and OFF.

The default is ON.

**COBOL\_NUMBERS**

Determines whether the sequence numbers should be maintained in COBOL format (in columns 1-6). Valid values are ON and OFF.

The default is OFF.

#### **STANDARD\_NUMBERS**

Determines whether the sequence numbers should be maintained in standard format (in columns 1-8 for variable data, the last 8 columns for fixed data). Valid values are ON and OFF.

The default is ON.

#### **AUTONUM**

Determines whether the editor automatically renumbers edit data. Valid values are ON and OFF.

The default is OFF.

#### **AUTOLIST**

Determines whether the editor writes the edit data to the ISPF LIST data set when a user ends an edit session in which data has been changed or saved. Valid values are ON and OFF.

The default is OFF.

#### **PROFILE**

Determines whether an edit profile can be deleted if it is the least recently used profile. Specify LOCK to prevent profiles from being deleted. Valid values are LOCK and UNLOCK.

The default is UNLOCK.

#### **AUTOSAVE**

Determines whether the editor automatically saves changes when the END command is entered. Valid values are ON and OFF.

The default is ON.

#### **AUTOSAVE\_PROMPT**

Determines whether the editor prompts the user to have the data changes saved if AUTOSAVE is OFF and the END command is entered. Valid values are PROMPT and NOPROMPT.

The default is PROMPT.

#### **HILITE**

Determines whether the editor uses color to highlight the data being edited. Valid values are ON and OFF.

The default is OFF.

#### **HILITE\_DOLOGIC**

Determines whether editor highlighting should use color to match DO/END statements. Valid values are ON and OFF.

The default is OFF.

#### **HILITE\_IFLOGIC**

Determines whether editor highlighting should use color to match IF/ELSE statements. Valid values are ON and OFF.

The default is OFF.

#### **HILITE\_PAREN**

Determines whether editor highlighting should use color to match open and close parentheses. Valid values are ON and OFF.

The default is OFF.

### **HILITE\_FIND**

Determines whether editor highlighting should use color to highlight the target of the FIND command. Valid values are ON and OFF.

The default is ON.

### **HILITE\_CURSOR**

Determines whether editor highlighting should use color to highlight current cursor location. Valid values are ON and OFF.

The default is ON.

### **HILITE\_LANGUAGE**

The default language to be used by edit highlighting. Valid values are:

- 1 Automatic language determination
- 2 Assembler
- 3 PL/I
- 4 COBOL
- 5 Pascal
- 6 C
- 7 BookMaster<sup>®</sup>
- 8 Rexx
- 9 ISPF Panel language
- 10 ISPF Skeleton language
- 11 JCL
- 12 ISPF Dialog Tag Language (DTL)
- 13 Other (CLIST, etc.)
- 14 Default (no highlighting)
- 15 PL/X
- 16 IDL
- 17 SuperC Listing
- 18 HTML
- 19 XML

The default is 1.

### **HILITE\_MARGIN\_C**

Determines the left and right margins to be used by edit highlighting when processing C statements. Values are specified in the form (*left-margin, right-margin*). Although the IBM C/C++ compiler supports left and right margins in the range 1 to 32760, ISPF only supports highlighting data sets up to a record length of 255. Therefore, the *left-margin* must be between 1 and 254 and the *right-margin* must be between 1 and 255.

You can specify (\*,\*) to use the default margins defined for the language. The default *left-margin* is 1. The default *right-margin* is the lesser of 255 and the last input column. The last input column is 8 less than the record length when the data set has fixed records and is in standard number mode; otherwise it is the same as the record length.

### **HILITE\_MARGIN\_PLI**

Determines the left and right margins to be used by edit highlighting when processing PL/I statements. Values are specified in the form (*left-margin, right-margin*). *left-margin* must be between 1 and 100. *right-margin* must be between 1 and 200.

You can specify (\*,\*) to use the default margins defined for the language. The default *left-margin* is 2. The default *right-margin* is the lesser of 200 and the last input column. The last input column is 8 less than the record

length when the data set has fixed records and is in standard number mode; otherwise it is the same as the record length.

#### **HILITE\_MARGIN\_PLX**

Determines the left and right margins to be used by edit highlighting when processing PL/X statements. Values are specified in the form (*left-margin,right-margin*). *left-margin* must be between 1 and 65. *right-margin* must be between 15 and 80.

You can specify (\*,\*) to use the default margins defined for the language. The default *left-margin* is 2. The default *right-margin* is the lesser of 80 and the last input column. The last input column is 8 less than the record length when the data set has fixed records and is in standard number mode; otherwise it is the same as the record length.

---

## **ISPF site-wide profile customizations**

ISPF site-wide defaults are not enabled until the VERSION\_LEVEL\_OF\_SITEWIDE\_DEFAULTS field is modified.

The RESET fields mean that ISPF resets the site-wide defaults for each user. This is done once each time the VERSION\_LEVEL\_OF\_SITEWIDE\_DEFAULTS field is incremented. Users can change the values of their fields after the incrementation has caused the reset.

#### **VERSION\_LEVEL\_OF\_SITEWIDE\_DEFAULTS**

This field indicates the modification level of the current site-wide defaults values. This field is set to 43000 initially by ISPF and will not be changed with new versions of ISPF. Each time you modify any of the sitewide defaults, increment this value by 1. For example, the first time you change any defaults you should set this field to 43001. This value is then saved in the ISPF system profile table.

When ISPF is initialized it checks the value saved in the system profile against the value in this field. If the value of this field is greater than that in the system profile, ISPF will use the values specified here. You must increment this value every time you modify a default or it will not be picked up by ISPF.

#### **TAB\_TO\_POINT\_AND\_SHOOT**

Enable tabbing to point and shoot fields. Valid values are YES and NO.

The default is NO.

#### **RESET\_TAB\_TO\_POINT\_AND\_SHOOT**

Reset the value specified here.

The default is NO.

#### **TAB\_TO\_ACTION\_BARS**

Enable tabbing to action bars. Valid values are YES and NO.

The default is YES.

#### **RESET\_TAB\_TO\_ACTION\_BARS**

Reset the value specified here.

The default is NO.

#### **USE\_SESSION\_MANAGER**

Value is used to set system variable ZSESS and is used to initialize system variable ZSM. Valid values are YES and NO.

The default is NO.

**RESET\_USE\_SESSION\_MANAGER**

Reset the value of system variable ZSM to the value of the USE\_SESSION\_MANAGER keyword.

**JUMP\_FROM\_LEADER\_DOTS**

Enable the ISPF jump command (for example, =2) from fields with leader dots. Valid values are YES and NO.

The default is YES.

**RESET\_JUMP\_FROM\_LEADER\_DOTS**

Reset the value specified here.

The default is NO.

**SHOW\_SPLIT\_LINE**

Should ISPF show the split line when a user is running in split screen mode. Valid values are YES and NO.

The default is YES.

**RESET\_SHOW\_SPLIT\_LINE**

Reset the value specified here.

The default is NO.

**LONG\_MESSAGES\_IN\_POPUP**

Should ISPF long messages always be shown in pop-up windows, or only when they are longer than 78 characters. Valid values are YES and NO.

The default is YES.

**RESET\_LONG\_MESSAGES\_IN\_POPUP**

Reset the value specified here.

The default is NO.

**EDIT\_PRINTDS\_COMMAND**

Should the user be allowed to modify the PRINTDS command generated by ISPF before its submission. Valid values are YES and NO.

The default is NO.

**RESET\_EDIT\_PRINTDS\_COMMAND**

Reset the value specified here.

The default is NO.

**RESTORE\_TEST\_TRACE\_OPTIONS**

Should the original TEST and TRACE options specified on ISPF invocation be restored when a user exits from Dialog Test. Dialog test will set TEST mode on. Valid values are YES and NO.

The default is YES.

**RESET\_RESTORE\_TEST\_TRACE\_OPTIONS**

Reset the value specified here.

The default is NO.

**DISPLAY\_PANELS\_IN\_CUA\_MODE**

Should ISPF panels be displayed in CUA mode. Valid values are YES and NO.

The default is YES.

### **RESET\_DISPLAY\_PANELS\_IN\_CUA\_MODE**

Reset the value specified here.

The default is NO.

### **LOG\_DATA\_SET\_DISPOSITION**

The default disposition for the ISPF Log data set. Valid defaults are:

- 1 Process option not set
- 2 Print and delete
- 3 Delete without printing
- 4 Keep
- 5 Keep and allocate a new log

The default is 1, process option not set.

**Note:** When manually editing the configuration table the valid options are:

<b>NONE</b>	Process option not set
<b>J</b>	Print and delete
<b>D</b>	Delete without printing
<b>K</b>	Keep
<b>R</b>	Keep and allocate a new log data set

The default is NONE, process option not set.

### **RESET\_LOG\_DATA\_SET\_DISPOSITION**

Reset the value specified here. The default is NO.

### **LIST\_DATA\_SET\_DISPOSITION**

The default disposition for the ISPF List data set. Valid defaults are:

- 1 Process option not set
- 2 Print and delete
- 3 Delete without printing
- 4 Keep
- 5 Keep and allocate a new list

The default is 1, process option not set.

**Note:** When manually editing the configuration table the valid options are:

<b>NONE</b>	Process option not set
<b>J</b>	Print and delete
<b>D</b>	Delete without printing
<b>K</b>	Keep
<b>R</b>	Keep and allocate a new list data set

The default is NONE, process option not set.

### **RESET\_LIST\_DATA\_SET\_DISPOSITION**

Reset the value specified here.

The default is NO.

### **COMMAND\_LINE\_PLACEMENT**

Placement of the ISPF command line. Valid values are:

- BOTTOM** Float the command line to the bottom of the panel.
- ASIS** Leave command line as coded on the panel.

The default is BOTTOM.

### **RESET\_COMMAND\_LINE\_PLACEMENT**

Reset the value specified here. The default is NO.

**USE\_KEYLISTS**

Specifies whether ISPF uses Keylists for pfkey definitions. Valid values are YES and NO.

The default is YES.

**RESET\_USE\_KEYLISTS**

Reset the value specified here.

The default is NO.

**SHOW\_PFKEYS**

Specifies whether ISPF will display the current PFKEY settings. Valid values are ON and OFF.

The default is ON.

**RESET\_SHOW\_PFKEYS**

Reset the value specified here.

The default is NO.

**SCROLL\_MEMBER\_LIST**

Specifies if ISPF should scroll to the first member selected in the member list after processing or disable the member list from automatic scrolling and instead place the cursor in front of the last member selected.

The default is YES.

**RESET\_SCROLL\_MEMBER\_LIST**

Reset the value specified here.

The default is NO.

**SCROLL\_DEFAULT**

Select the default scroll value. The valid settings are:

**CSR** When scrolling forward, move the line that currently contains the cursor to the first data line. When scrolling backward, move the line that currently contains the cursor to the last data line.

**DATA** Scroll by a full page minus one line when scrolling up or down, or by a full page minus one column when scrolling left or right.

**HALF** Half a screenful of data.

**MAX** Scroll to the top, bottom, left, or right margin.

**PAGE** One screenful of data.

The default is PAGE.

**SCROLL\_MIN**

Select the minimum scroll value allowed.

The default is 0.

**SCROLL\_MAX**

Select the maximum scroll value allowed.

The default is 9999.

**RESET\_SCROLL\_VALUE**

Force an update of ISPSPROF from the configuration table values at ISPF initialization.

The default is NO.

**DISPLAY\_EMPTY\_MEMBER\_LIST**

Controls whether an empty member list is displayed.

The default is NO.

**DISPLAY\_EMPTY\_MEMBER\_LIST\_PATTERN**

If the DISPLAY\_EMPTY\_MEMBER\_LIST option is set, this field controls whether an empty list that results from a nonmatching pattern will be displayed.

The default is NO.

**DISPLAY\_EMPTY\_MEMBER\_LIST\_FUNCTION**

Whether empty member list options apply to non-edit functions such as View and Browse.

The default is YES.

**RESET\_EMPTY\_MEMBER\_LIST\_OPTIONS**

Reset the values specified in the DISPLAY\_EMPTY\_MEMBER\_LIST fields.

The default is NO.

**STATUS\_AREA\_DEFAULT**

Select the default status area value. The valid settings are:

- SES** Session
- FUN** Function Keys
- CAL** Calendar
- USE** User status
- UPS** User Point and Shoot
- OFF** None

The default is SESSION.

**LIST\_DATA\_SET\_RECORDS\_PER\_BLOCK**

The number of records per block for the ISPF List data set. This value must be in the range 0 to 32760. A value of 0 will result in an ISPF list data set being allocated using a system-determined block size.

The default is 26.

**LOG\_DATA\_SET\_BLOCK\_SIZE**

The block size of the ISPF Log data set. This value must be in the range 0 to 32760. A value of 0 will result in an ISPF log data set being allocated using a system-determined block size.

The default is 129.

**LOG\_DATA\_SET\_RECORD\_LENGTH**

The record length of the ISPF Log data set.

The default is 125.

**BLOCK\_SIZE\_FOR\_TEMPORARY\_CNTL\_DATA\_SETS**

The block size for ISPF temporary control (CNTL) data sets. This value must be in the range 0 to 32760. A value of 0 will result in the data sets being allocated using a system-determined block size.

The default is 800.

**RECORD\_LENGTH\_FOR\_TEMPORARY\_CNTL\_DATA\_SETS**

The record length for ISPF temporary control (CNTL) data sets.

The default is 80.

**BLOCK\_SIZE\_FOR\_TEMPORARY\_LIST\_DATA\_SETS**

The block size for ISPF temporary list data sets. This value must be in the range 0 to 32760. A value of 0 will result in the data sets being allocated using a system-determined block size.

The default is 3146.

**RECORD\_LENGTH\_FOR\_TEMPORARY\_LIST\_DATA\_SETS**

The record length for ISPF temporary list data sets.

The default is 121.

**BLOCK\_SIZE\_FOR\_TEMPORARY\_WORK\_DATA\_SETS**

The block size for ISPF temporary work data sets. This value must be in the range 0 to 32760. A value of 0 will result in the data sets being allocated using a system-determined block size.

The default is 2560.

**RECORD\_LENGTH\_FOR\_TEMPORARY\_WORK\_DATA\_SETS**

The record length for ISPF temporary work data sets.

The default is 256.

**ISPCTL0\_BLOCK\_SIZE**

The block size for ISPCTL0 temporary control (CNTL) data set. This value must in the range 0 to 32760.

The default is 800.

**ISPCTL0\_RECORD\_LENGTH**

The record length for the ISPCTL0 temporary control (CNTL) data set.

The default is 80.

**ISPCTL0\_PRIMARY\_QUANTITY**

Primary number of blocks to be allocated for the ISPCTL0 temporary control (CNTL) data set.

The default is 10.

**ISPCTL0\_SECONDARY\_QUANTITY**

Secondary number of blocks to be allocated for the ISPCTL0 temporary control (CNTL) data set.

The default is 100.

**ISPCTL\_PRIMARY\_QUANTITY**

Primary number of blocks to be allocated for file tailoring CNTL data sets.

The default is 10.

**ISPCTL\_SECONDARY\_QUANTITY**

Secondary number of blocks to be allocated for file tailoring CNTL data sets.

The default is 100.

**ISPLST\_PRIMARY\_QUANTITY**

Primary number of blocks to be allocated for ISPF temporary list data sets.

The default is 10.

### **ISPLST\_SECONDARY\_QUANTITY**

Secondary number of blocks to be allocated for ISPF temporary list data sets.

The default is 100.

### **ISPWRK\_PRIMARY\_QUANTITY**

Primary number of blocks to be allocated for the file tailoring WORK data sets.

The default is 10.

### **ISPWRK\_SECONDARY\_QUANTITY**

Secondary number of blocks to be allocated for the File Tailoring WORK data sets.

The default is 100.

### **USE\_PDFCUNIT\_FOR\_TEMP\_ISPF\_DATA\_SETS**

Indicates whether to use the PDFCUNIT value for the units field when allocating ISPF temporary data sets. Valid values are YES and NO.

The default is NO.

### **ISPF\_TEMPORARY\_DATA\_SET\_QUALIFIER**

An additional qualifier that will be appended to the ISPF log, list, and temporary control data set names. The qualifier will come after the ISPF assigned prefix, but before the suffix area. If Exit 16 is active, this qualifier will be part of the 26-byte prefix area passed to the exit. If the configuration table field USE\_ADDITIONAL\_QUAL\_FOR\_PDF\_DATA\_SETS is set to YES, the qualifier is also appended to the default names of data sets created by PDF utilities.

The qualifier can be either:

1. A valid data set qualifier, comprising 1 to 8 alphanumeric characters, the first being alphabetic (not numeric)
2. A string containing 1 or more system symbolic variables.

The string may be up to 32 characters in length, but when resolved it will be truncated to 8 characters. Truncation errors are ignored.

Other characters may be included between the symbolic variables, providing they are alphanumeric characters and the first character is nonnumeric. The use of any of the date and time symbols requires an alphabetic character before the symbol name to ensure that the qualifier is valid. If the resulting qualifier is invalid, it is ignored without an error message being issued.

Examples:

```
&SYSNAME.  
SYS&SYSNAME(1:4).
```

See the *z/OS MVS Initialization and Tuning Reference* for details on valid system symbols.

The default is ISP&SEQ. where ISPF Profile Sharing is enabled, otherwise the default is NONE (no qualifier).

### **PROFILE\_SHARING**

Enable the ISPF Profile Sharing facility. Valid values are YES and NO. The default is NO.

#### **PROFILE\_ENQLOCK\_WAIT**

The time in milliseconds that ISPF is to wait when it is unable to obtain an enqueue for an ISPF profile member. The value is an integer from 0 to 9999. The default is 1000 milliseconds (1 second). A value of 0 results in no wait being issued.

#### **PROFILE\_ENQLOCK\_RETRY\_COUNT**

The number of attempts to retry the enqueue request when ISPF is unable to obtain the enqueue for an ISPF profile member. The value is an integer from 0 to 99. The default value is 1. A value of 0 results in no further attempt to obtain a failed enqueue.

#### **PROFILE\_ENQLOCK\_PROMPT**

Enable the prompt for further action when ISPF is unable to obtain the enqueue for an ISPF profile member. Valid values are YES and NO. The default is YES. When ISPF is running in a background environment or a value of NO is specified and ISPF is unable to obtain the enqueue for an ISPF profile member, the enqueue request is failed. When a value of YES is specified and ISPF is unable to obtain the enqueue for an ISPF profile member, a pop-up window is displayed giving you the option to either retry or fail the enqueue request.

#### **PROFILE\_SYSPROF\_CONFLICT**

The initial action to be taken when the ISPF system profile member, ISPSPROF, has been updated after it was read from disk. Valid values are:

##### **PROMPT**

Prompts you to either KEEP or DISCARD the changes to the ISPF system profile from the current ISPF session. You can also disable the prompt so that the same action is taken for further changes to the ISPF system profile during the current ISPF session.

**KEEP** Updates the System profile with changes from the current session. Other updates to the ISPF system profile made on other systems may be lost.

##### **DISCARD**

Updates made to the ISPF system profile during the current session are discarded.

The default is KEEP.

#### **PROFILE\_ISPPROF\_CONFLICT**

The initial action to be taken when the ISPF profile member, normally ISPPROF, has been updated after it was read from disk. Valid values are:

##### **PROMPT**

Prompts you to either KEEP or DISCARD the changes to the ISPF system profile from the current ISPF session. You can also disable the prompt so that the same action is taken for further changes to the ISPF system profile during the current ISPF session.

**KEEP** Updates the profile with changes from the current session. Other updates to the ISPF profile made on other systems may be lost.

##### **DISCARD**

Updates made to the ISPF profile during the current session are discarded.

The default is KEEP.

### **PROFILE\_APPPROF\_CONFLICT**

The initial action to be taken when an ISPF application profile member has been updated after it was read from disk. Valid values are:

#### **PROMPT**

Prompts you to either KEEP or DISCARD the changes to the ISPF system profile from the current ISPF session. You can also disable the prompt so that the same action is taken for further changes to the ISPF system profile during the current ISPF session.

**KEEP** Updates the Application profile with changes from the current session. Other updates to the ISPF application profile made on other systems may be lost.

#### **DISCARD**

Updates made to the ISPF application profile during the current session are discarded.

The default is KEEP.

### **PROFILE\_REFLIST\_CONFLICT**

The initial action to be taken when an ISPF reference list member, ISRLLIST, ISRPLIST, or ISRSLIST, has been updated after it was read from disk. Valid values are:

#### **PROMPT**

Prompts you to either KEEP or DISCARD the changes to the ISPF system profile from the current ISPF session. You can also disable the prompt so that the same action is taken for further changes to the ISPF system profile during the current ISPF session.

**KEEP** Updates the reference list with changes from the current session. Other updates to the ISPF reference list made on other systems may be lost.

#### **DISCARD**

Updates made to the ISPF reference list during the current session are discarded.

The default is KEEP.

### **PROFILE\_EDIT\_CONFLICT**

The initial action to be taken when an ISPF edit profile, *xxxxEDRT*, has been updated after it was read from disk. Valid values are:

#### **PROMPT**

Prompts you to either KEEP or DISCARD the changes to the ISPF system profile from the current ISPF session. You can also disable the prompt so that the same action is taken for further changes to the ISPF system profile during the current ISPF session.

**KEEP** Updates the edit profile with changes from the current session. Other updates to the ISPF edit profile made on other systems may be lost.

#### **DISCARD**

Updates made to the ISPF edit profile during the current session are discarded.

The default is KEEP.

### **PROFILE\_BATCH\_CONFLICT**

The action to be taken when ISPF is executing in a background environment and any profile has been updated after it was read from disk. Valid values are:

**KEEP** Updates the profile member with changes from the current session. Other updates to the ISPF profile member made on other systems may be lost.

#### **DISCARD**

Updates made to the ISPF profile member during the current session are discarded. The default is DISCARD.

### **PROFILE\_OTHER\_CONFLICT**

The initial action to be taken when any other ISPF table in the ISPF ISPF profile data set has been updated after it was read from disk. Valid values are:

#### **PROMPT**

Prompts you to either KEEP or DISCARD the changes to the ISPF system profile from the current ISPF session. You can also disable the prompt so that the same action is taken for further changes to the ISPF system profile during the current ISPF session.

**KEEP** Updates the ISPF table with changes from the current session. Other updates to the ISPF table made on other systems may be lost.

#### **DISCARD**

Updates made to the ISPF table during the current session are discarded.

The default is KEEP.

Any conflicts that occur during ISPF termination where PROMPT is specified and the ISPF display services are no longer available are converted to the default action.

### **RESET\_PROFILE\_SHARING\_SETTINGS**

Option to specify that all the Profile Sharing settings are to be reset to the ISPF Configuration defined values. Valid values are YES and NO. The default is NO.

### **PRINTDS\_DEST\_OR\_WRITER\_OPTION**

PRINTDS option, valid values are:

**DEST** Indicates a local printer ID is being used.

**WRITER** Indicates an external writer name is being used.

The default is DEST.

### **LOCAL\_PRINTDS\_OPTIONS**

Parameters appended to the PRINTDS command when a local print is done. No verification of these fields is done. The maximum length of this value is 128 characters. To disable printing through PRINTDS, specify a value of DISABLE.

The default is NONUM.

### **USE\_ALTERNATE\_DIALOG\_TEST\_PANEL**

Select alternate dialog test panels ISPYFP, ISPYFPA, or ISPYFPB. Panel ISPYFPA is formatted with the most frequently used fields at the top of the panel. Panel ISPYFPB is similar to panel ISPYFPA, but it has a selection

field that allows the user to select a function: panel, command, program, or request. Unlike the panels ISPYFP or ISPYFPA, on panel ISPYFPB the panel, command, program, or request fields can all contain values. Valid options are:

- 1 ISPYFP panel
- 2 ISPYFPA panel
- 3 ISPYFPB panel

The default is 1.

#### USE\_ADDITIONAL\_QUAL\_FOR\_PDF\_DATA\_SETS

An additional qualifier is included in the default data set name for data sets generated by PDF utilities. The data sets affected by this setting are:

- Listing and update data sets produced by the SuperC and Srchfor utilities.
- Listing data sets produced by the data set list and member list SRCHFOR commands.
- Trace data sets generated by the ISPVCALL, ISDPTRC, and ISPFTRC commands.
- Data set created by the table utility EXPORT and FEXPORT commands.

The value specified for the ISPF\_TEMPORARY\_DATA\_SET\_QUALIFIER field is used as the additional qualifier which is included at the start of the default suffix for these data set names.

Valid values are YES and NO. The default is NO.

---

## Default CUA color settings

These fields in the ISPF Configuration table define the color and highlighting for each of the CUA panel elements. The setting for each element consists of a 3-character numeric field, with each position meaning:

#### Position 1 — COLOR

- 1 Blue
- 2 Red
- 3 Pink
- 4 Green
- 5 Turquoise
- 6 Yellow
- 7 White

#### Position 2 — INTENSITY

- 0 Low
- 2 High

#### Position 3 — HILITE

- 0 None
- 1 Blink
- 2 Reverse Video
- 4 Underscore

ISPF site-wide defaults are not enabled until the VERSION\_LEVEL\_OF\_SITEWIDE\_DEFAULTS field is modified. See “ISPF site-wide profile customizations” on page 217 for more details. If you select any RESET fields here, you *must* increment the value in the VERSION\_LEVEL\_OF\_SITEWIDE\_DEFAULTS.

The RESET fields here mean that ISPF resets the site-wide defaults for each user. This is done once each time the VERSION\_LEVEL\_OF\_SITEWIDE\_DEFAULTS field is incremented. Users can change the values of their fields after the incrementation has caused the reset.

**ACTION\_BAR\_SELECTED\_CHOICE**

Default is 600 (yellow, low intensity, no highlighting).

**RESET\_ACTION\_BAR\_SELECTED\_CHOICE**

Reset the value specified here.

The default is NO.

**ACTION\_BAR\_SEPARATOR\_LINE**

Default is 100 (Blue, low intensity, no highlighting).

**RESET\_ACTION\_BAR\_SEPARATOR\_LINE**

Reset the value specified here.

The default is NO.

**ACTION\_BAR\_UNSELECTED\_CHOICE**

Default is 720 (White, high intensity, no highlighting).

**RESET\_ACTION\_BAR\_UNSELECTED\_CHOICE**

Reset the value specified here.

The default is NO.

**ACTION\_MESSAGE\_TEXT**

Default is 220 (Red, high intensity, no highlighting).

**RESET\_ACTION\_MESSAGE\_TEXT**

Reset the value specified here.

The default is NO.

**CAUTION\_TEXT**

Default is 620 (Yellow, high intensity, no highlighting).

**RESET\_CAUTION\_TEXT**

Reset the value specified here.

The default is NO.

**CHOICE\_ENTRY\_FIELD**

Default is 504 (Turquoise, low intensity, underscored)

**RESET\_CHOICE\_ENTRY\_FIELD**

Reset the value specified here.

The default is NO.

**COLUMN\_HEADING**

Default is 120 (Blue, high intensity, no highlighting).

**RESET\_COLUMN\_HEADING**

Reset the value specified here.

The default is NO.

**DESCRIPTIVE\_TEXT**

Default is 400 (Green, low intensity, no highlighting).

**RESET\_DESCRIPTIVE\_TEXT**

Reset the value specified here.

The default is NO.

**EMPHASIZED\_TEXT**

Default is 520 (Turquoise, high intensity, no highlighting).

**RESET\_EMPHASIZED\_TEXT**

Reset the value specified here.

The default is NO.

**ERROR\_EMPHASIS**

Default is 622 (Yellow, high intensity, reverse video)

**RESET\_ERROR\_EMPHASIS**

Reset the value specified here.

The default is NO.

**FIELD\_PROMPT**

Default is 400 (Green, low intensity, no highlighting).

**RESET\_FIELD\_PROMPT**

Reset the value specified here.

The default is NO.

**FUNCTION\_KEYS**

Default is 100 (Blue, low intensity, no highlighting).

**RESET\_FUNCTION\_KEYS**

Reset the value specified here.

The default is NO.

**INFORMATIONAL\_MESSAGE\_TEXT**

Default is 720 (White, high intensity, no highlighting).

**RESET\_INFORMATIONAL\_MESSAGE\_TEXT**

Reset the value specified here.

The default is NO.

**LIST\_ENTRY\_FIELD**

Default is 504 (Turquoise, low intensity, underscored)

**RESET\_LIST\_ENTRY\_FIELD**

Reset the value specified here.

The default is NO.

**LIST\_ITEM\_DESCRIPTION**

Default is 400 (Green, low intensity, no highlighting).

**RESET\_LIST\_ITEM\_DESCRIPTION**

Reset the value specified here.

The default is NO.

**LIST\_ITEM**

Default is 700 (White, low intensity, no highlighting).

**RESET\_LIST\_ITEM**

Reset the value specified here.

The default is NO.

**NORMAL\_ENTRY\_FIELD**

Default is 504 (Turquoise, low intensity, underscored)

**RESET\_NORMAL\_ENTRY\_FIELD**

Reset the value specified here.

The default is NO.

**NORMAL\_TEXT**

Default is 400 (Green, low intensity, no highlighting).

**RESET\_NORMAL\_TEXT**

Reset the value specified here.

The default is NO.

**PANEL\_ID**

Default is 100 (Blue, low intensity, no highlighting).

**RESET\_PANEL\_ID**

Reset the value specified here.

The default is NO.

**PANEL\_INFORMATION**

Default is 400 (Green, low intensity, no highlighting).

**RESET\_PANEL\_INFORMATION**

Reset the value specified here.

The default is NO.

**PANEL\_TITLE**

Default is 100 (Blue, low intensity, no highlighting).

**RESET\_PANEL\_TITLE**

Reset the value specified here.

The default is NO.

**POINT\_AND\_SHOOT**

Default is 520 (Turquoise, high intensity, no highlighting).

**RESET\_POINT\_AND\_SHOOT**

Reset the value specified here.

The default is NO.

**PULLDOWN\_AVAILABLE\_CHOICE**

Default is 700 (White, low intensity, no highlighting).

**RESET\_PULLDOWN\_AVAILABLE\_CHOICE**

Reset the value specified here.

The default is NO.

**PULLDOWN\_UNAVAILABLE\_CHOICE**

Default is 100 (Blue, low intensity, no highlighting).

**RESET\_PULLDOWN\_UNAVAILABLE\_CHOICE**

Reset the value specified here.

The default is NO.

**REFERENCE\_PHRASE**

Default is 720 (White, high intensity, no highlighting).

**RESET\_REFERENCE\_PHRASE**

Reset the value specified here.

The default is NO.

**SCROLL\_INFORMATION**

Default is 720 (White, high intensity, no highlighting).

**RESET\_SCROLL\_INFORMATION**

Reset the value specified here.

The default is NO.

**SELECTION\_AVAILABLE\_CHOICE**

Default is 700 (White, low intensity, no highlighting).

**RESET\_SELECTION\_AVAILABLE\_CHOICE**

Reset the value specified here.

The default is NO.

**SELECTION\_UNAVAILABLE\_CHOICE**

Default is 100 (Blue, low intensity, no highlighting).

**RESET\_SELECTION\_UNAVAILABLE\_CHOICE**

Reset the value specified here.

The default is NO.

**VARIABLE\_OUTPUT\_INFORMATION**

Default is 500 (Turquoise, low intensity, no highlighting).

**RESET\_VARIABLE\_OUTPUT\_INFORMATION**

Reset the value specified here.

The default is NO.

**WARNING\_MESSAGE\_TEST**

Default is 620 (Yellow, high intensity, no highlighting).

**RESET\_WARNING\_MESSAGE\_TEST**

Reset the value specified here.

The default is NO.

**WARNING\_MESSAGE**

Default is 220 (Red, high intensity, no highlighting).

**RESET\_WARNING\_MESSAGE**

Reset the value specified here.

The default is NO.

**WORKAREA\_SEPARATOR\_LINE**

Default is 100 (Blue, low intensity, no highlighting).

**RESET\_WORKAREA\_SEPARATOR\_LINE**

Reset the value specified here.

The default is NO.

---

## Miscellaneous settings

**MONITOR\_EDIT\_MACRO\_COMMANDS**

Should the ISPF Activity Monitoring Exit be invoked for ISREDIT commands invoked from an ISPF Edit Macro. Valid values are YES and NO.

The default is NO.

**ALLOW\_SUBMIT\_FROM\_BROWSE**

Should users be allowed to issue the SUBMIT command from with a Browse session. Valid values are YES and NO.

The default is YES.

**ALLOW\_SUBMIT\_FROM\_VIEW**

Should users be allowed to issue the SUBMIT command from with a View session. Valid values are YES and NO.

The default is YES.

**WARN\_ON\_RENAME\_TO\_GDG\_NAME**

Should a warning panel be displayed when a user attempts to rename a data set to a new name that matches the naming convention of a GDG generation. The renamed data set may become a valid generation if it matches the naming convention of an existing GDG data set. If that new generation causes the LIMIT parameter value specified when the GDG was defined to be exceeded, the system will take action based on the SCRATCH/NOSCRATCH and EMPTY/NOEMPTY parameters that were specified when the GDG was defined. This action may result in one or all of the existing generations being deleted or uncataloged. Valid values are YES and NO.

The default is YES.

**DEFAULT\_EDIT/BROWSE/VIEW\_MEMBER\_LIST**

Should option 3.4 (Data Set List utility) use the enhanced member list for the Edit, Browse and View actions. Performance is improved if the traditional member list ID used, but capability is improved using the enhanced member list. Valid values are YES and NO.

The default is YES.

**IS\_VIEW\_SUPPORTED**

Should users be allowed to use the View function from option 1 or only the Browse function. Because it is based on ISPF Edit, View can impact system resource utilization. Valid values are YES and NO.

The default is YES.

**USE\_ALTERNATE\_PANEL\_ISRTSOA**

Should alternate ISPF Command Shell panel ISRTSOA be used in place of panel ISRTSO. ISRTSOA contains both an ISPF command line and a TSO command line, ISRTSO contains only one input field for both ISPF and TSO commands. Valid values are YES and NO.

The default is NO.

**PRINT\_USING\_ICF**

Indicates whether foreground print requests should be processed using an ICF printer definition. Valid values are YES and NO.

The default is NO.

**DISALLOW\_WILDCARDS\_IN\_HLQ**

Indicates whether wildcards (\* or %) are allowed in the high-level qualifier for data set list. Valid values are YES and NO.

The default is NO.

**MAXIMUM\_NUMBER\_OF\_SPLIT\_SCREEN**

Maximum number of separate logical screens a user can have active. The maximum value for this field is 32, and minimum value is 4.

The default is 8.

#### **APPLID\_FOR\_USER\_COMMAND\_TABLE**

The application ID for 1 to 3 user command tables. The application ID must be 1 to 4 alphanumeric or special characters, with the first character being either alphabetic or a special character. The application ID values can be specified in either of the formats:

**User** to specify a single application ID.

**(usr1) or (usr1,usr2) or (usr1,usr2,usr3)**

to specify 1 to 3 application IDs.

The default value is NONE (no user command tables).

In addition, a special format can be used to obtain the application ID from the current system name (ISPF dialog variable ZSYSID). The special format is:

**\*, \*m, or \*m:n.**

As the system name can be up to 8 characters, *m* and *n* are the start and end positions within the system name used to determine the application ID for the user command tables. The values for *m* and *n* must be in the range 1 to 8, where *m* is less than or equal to *n* and the difference in their values is no more than 3.

The default value for *m* is 1. The default value for *n* is *m*+3, to a maximum value of 8.

#### **Compatibility Issue:**

For compatibility of the ISPF configuration options with previous releases, do not specify any of the special formats (\*, \*m, or \*m:n) as either the first user or site application ID.

#### **APPLID\_FOR\_SITE\_COMMAND\_TABLE**

The application ID for 1 to 3 site command tables. The application ID must be 1 to 4 alphanumeric or special characters, with the first character being either alphabetic or a special character. The application ID values can be specified in either of the formats:

**Site** to specify a single application ID.

**(sit1) or (sit1,sit2) or (sit1,sit2,sit3)**

to specify 1 to 3 application IDs.

The default value is NONE (No site-wide command tables).

In addition, a special format can be used to obtain the application ID from the current system name (ISPF dialog variable ZSYSID). The special format is:

**\*, \*m, or \*m:n.**

As the system name can be up to 8 characters, *m* and *n* are the start and end positions within the system name used to determine the application ID for the site command tables. The values for *m* and *n* must be in the range 1 to 8, where *m* is less than or equal to *n* and the difference in their values is no more than 3.

The default value for *m* is 1. The default value for *n* is *m*+3, to a maximum value of 8.

#### **Compatibility Issue:**

For compatibility of the ISPF configuration options with previous

releases, do not specify any of the special formats (\*, \*m, or \*m:n) as either the first user or site application ID.

#### **SITE\_COMMAND\_TABLE\_SEARCH\_ORDER**

Determines whether the site-wide command tables are searched before or after the default ISP command table. Valid values are AFTER and BEFORE.

The default is BEFORE.

Depending on this setting, the search order will be:

Search Order = BEFORE

1. Application
2. USER (1 to 3)
3. SITE (1 to 3)
4. System

Search Order = AFTER

1. Application
2. USER (1 to 3)
3. System
4. SITE (1 to 3)

#### **YEAR\_2000\_SLIDING\_RULE**

The cutoff value used by ISPF to determine whether a 2-character year date specified to ISPF should be considered a 19xx or 20xx date. Values less than or equal to this date will be considered 20xx, values greater will be considered 19xx. Value can be an absolute number or a number preceded by a minus sign to indicate the cutoff should be the specified number of years before the current year. For example, specifying 72 indicates that any 2-character year less than or equal to 72 should be considered 20xx, anything greater should be 19xx. Specifying -40 (assuming the current year is 1999) will yield a cutoff value of 59.

The default is 65.

#### **SHOW\_ENQ\_DISPLAYS**

Used to indicate that users should not be able to see who has existing data set ENQs when they press the help key or when they use the ISRDDN utility.

The default is blank.

#### **DEFAULT\_SESSION\_LANGUAGE**

Selects the language to use as the default language for ISPF. The value chosen will be the language used by ISPF if no language is specified on invocation. Valid values are:

1. English
2. Uppercase English
3. German
4. Swiss German
5. Japanese

The default is 1 (English).

#### **PATHNAME\_SUBSTITUTION\_CHARACTER**

This field defines the character which can be used to represent the full path name of a selected file. This character shows the position of the file name when it is specified as an argument in a z/OS UNIX command, TSO command, CLIST, or REXX exec. The substitution character can be used with commands that are invoked through the z/OS UNIX Directory List utility (ISPF option 3.17).

The default character is ! (exclamation point).

---

## Values formerly in ISPDFLTS

These fields represent values in the ISPF Configuration table that in previous releases of ISPF were set through ISPDFLTS.

### **NUMBER\_OF\_ROWS\_FOR\_TBADD**

The number of rows to be used in calculating the amount of storage required when a TBADD service is invoked. The value can be an integer from 1 to 1000.

The default is 1.

### **RETRIEVE\_COMMAND\_STACK\_SIZE**

The size, in bytes, of the command stack that RETRIEVE command uses. ISPF uses the command stack to hold commands (stripped of leading and trailing blanks) and some ISPF internal information. A stack of 512 bytes holds approximately 20 commands with a length of 10 characters each. The value can be an integer from 312 to 4096.

The default is 512.

### **ENABLE\_ISPF\_EXITS**

Indicates whether exit routines are available:

**YES** Indicates that installation-written routines are provided or planned, and that the ISPEXITS load module is to be loaded at ISPF initialization. If you plan to use exit routines but those routines have not been written, you can code YES for this field, causing the IBM-provided defaults module, ISPEXITS, to be loaded. Later, you can replace ISPEXITS without having to repeat this part of the installation process.

**NO** Indicates that exit routines are not provided or planned, and that ISPEXITS is not to be loaded, reducing startup time.

The default is NO.

---

## VSAM data set restrictions

### **VSAM\_RESTRICTED\_EDIT\_DATASET**

The names of the data sets that are restricted from use in Edit. Wildcards may be used in the data set name (as in option 3.4) to specify sets of restricted data sets.

The default is NONE (all data sets allowed).

### **VSAM\_RESTRICTED\_BROWSE\_DATASET**

The names of the data sets that are restricted from use in Browse. Wildcards may be used in the data set name (as in option 3.4) to specify sets of restricted data sets.

The default is NONE (all data sets allowed).

### **VSAM\_RESTRICTED\_VIEW\_DATASET**

The names of the data sets that are restricted from use in View. Wildcards may be used in the data set name (as in option 3.4) to specify sets of restricted data sets.

The default is NONE (all data sets allowed).

---

## ISPSPROF general values

ISPF site-wide defaults are not enabled until the `VERSION_LEVEL_OF_SITEWIDE_DEFAULTS` field is modified. See “ISPF site-wide profile customizations” on page 217 for more details. If you select any RESET fields here, you *must* increment the value in the `VERSION_LEVEL_OF_SITEWIDE_DEFAULTS`.

The RESET fields here mean that ISPF resets the site-wide defaults for each user. This is done once each time the `VERSION_LEVEL_OF_SITEWIDE_DEFAULTS` field is incremented. Users can change the values of their fields after the incrementation has caused the reset.

### **LOG/LIST\_JOB\_CARD1**

First job card for log/list.

The default is NONE (blank).

### **LOG/LIST\_JOB\_CARD2**

Second job card for log/list.

The default is NONE (blank).

### **LOG/LIST\_JOB\_CARD3**

Third job card for log/list.

The default is NONE (blank).

### **LOG/LIST\_JOB\_CARD4**

Fourth job card for log/list.

The default is NONE (blank).

### **LOG\_DATA\_SET\_UNIQUE\_CHARACTER**

This unique character is used as the default only when a first-time user profile is created.

The default is 1.

### **LOG\_SYSOUT\_CLASS**

Log data set sysout class.

The default is NONE (blank).

### **RESET\_LOG\_SYSOUT\_CLASS**

Reset the Log data set sysout class.

The default is NO.

### **LOCAL\_LOG\_SYSOUT\_CLASS**

Local log data set sysout class.

The default is NONE (blank).

### **RESET\_LOCAL\_LOG\_SYSOUT\_CLASS**

Reset the Local log data set sysout class.

The default is NO.

### **LOG\_DISPLAY\_REQUIRED**

Log panel display required on termination. Valid values are YES or NO.

The default is NO.

### **LOG\_KEPT**

The log data set is to be kept. Valid values are YES or NO.

The default is NO.

**LOG\_LINES\_PER\_PAGE**

The number of log lines per page.

The default is 60.

**LOG\_MESSAGE\_ID**

The log message ID. Valid values are YES or NO.

The default is NO.

**LOG\_LOCAL\_PRINTER\_ID**

The local printer ID (CHAR(17)) for the log.

The default is NONE (blank).

**RESET\_LOG\_LOCAL\_PRINTER\_ID**

Reset the log local printer ID for the log.

The default is NO.

**LOG\_PAGES\_PRIMARY\_QUANTITY**

The primary quantity of log pages.

The default is 10.

**LOG\_PAGES\_SECONDARY\_QUANTITY**

The secondary quantity of log pages.

The default is 10.

**LIST\_DATA\_SET\_UNIQUE\_CHARACTER**

This unique character is used as the default only when a first-time user profile is created.

The default is 1.

**LIST\_SYSOUT\_CLASS**

List data set sysout class.

The default is NONE (blank).

**RESET\_LIST\_SYSOUT\_CLASS**

Reset the List data set sysout class.

The default is NO.

**LOCAL\_LIST\_SYSOUT\_CLASS**

Local list data set sysout class.

The default is NONE (blank).

**RESET\_LOCAL\_LIST\_SYSOUT\_CLASS**

Reset Local list data set sysout class.

The default is NO.

**LIST\_DISPLAY\_REQUIRED**

List panel display required on termination. Valid values are YES or NO.

The default is NO.

**LIST\_KEPT**

The list data set is to be kept. Valid values are YES or NO.

The default is NO.

**LIST\_LINES\_PER\_PAGE**

The number of list lines per page.

The default is 60.

**LIST\_LINE\_LENGTH**

The line length of the list data set.

The default is 120.

**LIST\_LOCAL\_PRINTER\_ID**

The local printer ID (CHAR(17)) for the list.

The default is NONE (blank).

**RESET\_LIST\_LOCAL\_PRINTER\_ID**

Reset the list local printer ID for the list. Valid values are YES or NO.

The default is NO.

**LIST\_RECORD\_FORMAT**

The record format for the list.

The default is FBA.

**LIST\_LOGICAL\_RECORD\_LENGTH**

The logical record length for the list.

The default is 121.

**RESET\_LIST\_LRECL\_AND\_RECFCM**

Reset the LIST\_LOGICAL\_RECORD\_LENGTH and LIST\_RECORD\_FORMAT.

The default is NO.

**LIST\_PAGES\_PRIMARY\_QUANTITY**

The primary quantity of list pages.

The default is 100.

**LIST\_PAGES\_SECONDARY\_QUANTITY**

The secondary quantity of list pages.

The default is 200.

**UNIQUE\_JOB\_CHARACTER**

The unique job character.

The default is NONE (blank).

**SCREEN\_FORMAT**

The screen format. The screen format you choose depends on the type of terminal you are using or the type of terminal your emulator is emulating. These formats are available:

**DATA** Format based on data width (only 3278 model 5 terminal)

**STD** Format 24 lines by 80 characters

**MAX** Format 27 lines by 132 characters

**PART** Format using hardware partitions (only 3290 terminal)

The default is STD.

**TERMINAL\_TYPE**

The terminal type depends on the type of terminal you are using or the type of terminal your emulator is emulating.

The default is 3278.

**FAMILY\_PRINTER**

The printer type of the destination device. Two (2) is the only valid value and represents a QUEUED printer.

**DEVICE\_NAME**

The device name is the destination of printed output. On MVS, this is the VTAM<sup>®</sup> node name for the printer and is installation-dependent.

The default is NONE.

**RESET\_DEVICE\_NAME**

Reset the device name specified.

The default is NO.

**ASPECT\_RATIO**

Allows the user to preserve the Graphics Aspect Ratio for a "true" picture (0) or to preserve the positional relationship between the graphics and alphanumerics (1).

The default is zero (0).

**PAD\_CHARACTER**

The character entered here will be used to fill input fields on a panel. It must be different than the command delimiter and it cannot be a-z, A-Z, 0-9 or /, except N and B can be used to indicate nulls and blanks respectively.

The default is B.

**DELIMITER**

The character entered here will be used to separate multiple commands entered on a command line. Alphabetic and alphanumeric characters as well as = (equal sign) and . (period) are not valid.

The default is ; (semicolon).

**RETRIEVE\_MINIMUM\_LENGTH**

The minimum number of characters ISPF should save in the retrieve stack. Valid values are 1 through 99.

The default is 1.

**RETRIEVE\_CURSOR\_POSITION**

Cursor position relative to the retrieved command. Valid values are:

- 1 Place cursor at the beginning of the string.
- 2 Place cursor at the end of the string.

The default is 1.

**ENABLE\_DUMP**

Enable a dump for a subtask ABEND when not in ISPF TEST mode. Valid values are ON and OFF.

The default is OFF.

**TERMTRAC\_DD\_NAME**

The terminal tracing (TERMTRAC) ddname.

The default is ISPSNAP.

**RESET\_TERMTRAC\_DD\_NAME**

Reset the TERMTRAC ddname specified. Valid values are YES and NO.

The default is NO.

#### **ENVIRON\_TERMTRAC\_VALUE**

Enable terminal tracing. Valid values are ON, OFF and ERROR.

The default is OFF.

#### **PFKEY\_PREFIX**

Select pfkey prefix. Valid values are YES and NO.

The default is YES.

#### **FKA\_SETTING**

Current state of the function key form. Valid values are LONG, SHORT and OFF (no display).

The default is LONG.

#### **NUMBER\_OF\_PFKEYS**

Number of function keys. Valid values are 12 or 24.

The default is 12.

#### **CHARACTER\_SET\_LOAD\_MODULE**

Character set load module name.

The default is ISP3278.

#### **RESET\_CHARACTER\_SET\_LOAD\_MODULE**

Reset the character set load module.

The default is NO.

#### **FRAME\_COLOR**

The color for window frames. Valid values are:

- 1 Blue
- 2 Red
- 3 Pink
- 4 Green
- 5 Turquoise
- 6 Yellow
- 7 White

The default is 6 (Yellow).

#### **FRAME\_INTENSITY**

The intensity for window frames. Valid values are 0 (Low) and 2 (High).

The default is 2 (High).

#### **TPUT\_BUFFER\_BLOCKSIZE**

The TPUT buffer block size can range from 512 to 32767. If defined as 0, ISPF will use its own calculations to determine the TPUT buffer size. If defined within the range 512 to 32767, ISPF will use the defined value as the buffer block size.

The default is 0.

#### **GLOBAL\_COLORS**

Determines the colors that are displayed while running ISPF in host mode. The string is a 7 digit number, each digit specifying the color to be substituted for another host color. The order of the digits in the string, and the number that represents each color is:

1. Blue
2. Red

3. Pink
4. Green
5. Turquoise
6. Yellow
7. White

For example, to substitute blue for green and accept all other default colors, specify the number for blue (1) in the position for green (fourth position): 1231567.

The default is 1234567.

### GUI\_GLOBAL\_COLORS

The global colors supported by the workstation. In GUI mode you can map each of the 7 host colors to one of the 16 workstation colors. The value is specified as a string of seven 2-digit values mapping the workstation color to the host color. The order of the host colors is the same as for the GLOBAL\_COLORS keyword. The 16 workstation colors are:

- |    |               |
|----|---------------|
| 0  | Black         |
| 1  | Dark Blue     |
| 2  | Dark Green    |
| 3  | Dark Cyan     |
| 4  | Dark Red      |
| 5  | Dark Magenta  |
| 6  | Brown         |
| 7  | Light Gray    |
| 8  | Dark Gray     |
| 9  | Light Blue    |
| 10 | Light Green   |
| 11 | Light Cyan    |
| 12 | Light Red     |
| 13 | Light Magenta |
| 14 | Yellow        |
| 15 | White         |

The default value is 01041302030600. Each 2-digit pair corresponds to a workstation color, in a host color position. The 01 in the first position tells the workstation to display the color Dark Blue wherever the host displays the color Blue. The default sequence maps the colors as:

Host color	Workstation color
<b>Blue</b>	Dark Blue
<b>Red</b>	Dark Red
<b>Pink</b>	Light Magenta
<b>Green</b>	Dark Green
<b>Turquoise</b>	Dark Cyan
<b>Yellow</b>	Brown
<b>White</b>	Black

### HOST\_COLORS

Select host colors for panel fields when on the workstation. Valid values are ON (selected) or OFF (not selected).

The default is OFF (not selected).

**Note:** HOST\_COLORS and PC\_COLORS are mutually exclusive. If one is selected, DO NOT select the other.

**PC\_COLORS**

Select personal computer colors for panel fields when on the workstation. Valid values are ON (selected) or OFF (not selected).

The default is OFF (not selected).

**Note:** HOST\_COLORS and PC\_COLORS are mutually exclusive. If one is selected, DO NOT select the other.

**DEFAULT\_MESSAGE\_ID**

Select to display the message identifier. The valid values are OFF or ON, the default is OFF.

**DEFAULT\_PANEL\_ID**

Select to display the panel identifier. The valid values are OFF or ON, the default is OFF.

**DEFAULT\_SCREEN\_NAME**

Select to display the screen name. The valid values are OFF or ON.

The default is OFF.

**DEFAULT\_SYSTEM\_NAME**

Select to display the system name. The valid values are OFF or ON.

The default is OFF.

**DEFAULT\_USERID\_DISP**

Select to display the user identifier. The valid values are OFF or ON.

The default is OFF.

**DEFAULT\_PRIMARY\_PANEL**

Enter a panel name for the default primary panel.

The default is ISP@MSTR.

**DEFAULT\_LIBDEF\_PROCESSING\_OPTION**

Enter the default option for processing LIBDEF requests. The valid values are COND, UNCOND, STACK, or STKADD.

The default is UNCOND.

**ENABLE\_EURO\_SYMBOL**

Enable the Euro currency symbol. Valid values are YES or NO.

The default is NO.

**RESET\_ENABLE\_EURO\_SYMBOL**

Reset enable the Euro currency symbol field.

The default is NO.

**DATE\_FORMAT\_ZDATEFD**

The date format using the national language convention for the day, month, year and the national language separator. If the value is DEFAULT ISPF will use the value defined in the ISPF Literal Load Module.

The default is DEFAULT.

**DATE\_FORMAT\_ZDATEF**

The date format using the characters DD for day, MM for month, and YY for year and the national language separator. The order of year month and day may change. If the value is DEFAULT, ISPF uses the value defined in the ISPF Literal Load Module.

The default is DEFAULT.

#### **DEFAULT\_TIME\_SEPARATOR**

The separator used in the time of day format. For example, the colon (:) in hh:mm:ss. If the value is D, ISPF uses the value defined in the ISPF Literal Load Module.

The default is the D.

---

## **ISPSPROF workstation defaults**

#### **GUI\_APPC\_ADDRESS**

The workstation's APPC network name in fully qualified LU name format or in symbolic destination name format.

The default is NONE.

#### **GUI\_APPC\_ADDRESS2**

The workstation's APPC network name in fully qualified LU name format or in symbolic destination name format.

The default is NONE.

#### **GUI\_APPC\_ADDRESS3**

The workstation's APPC network name in fully qualified LU name format or in symbolic destination name format.

The default is NONE.

#### **GUI\_APPC\_ADDRESS4**

The workstation's APPC network name in fully qualified LU name format or in symbolic destination name format.

The default is NONE.

#### **GUI\_APPC\_ADDRESS5**

The workstation's APPC network name in fully qualified LU name format or in symbolic destination name format.

The default is NONE.

#### **GUI\_CLOSE\_WINDOW**

The command to be processed when a GUI window is closed. Valid values are:

- 1 CANCEL
- 2 END
- 3 EXIT
- 4 RETURN

The default value is 1 (CANCEL).

#### **GUI\_DISPLAY\_ENTER\_KEY**

If YES the enter key button will be displayed at the bottom of GUI panels. Valid values are YES or NO.

The default is YES.

#### **GUI\_WINDOW\_FRAME**

Select the type of GUI window frame. Valid values are:

- 1 **Standard** a GUI window frame that can be resized and has max/min buttons
- 2 **Fixed** a GUI window frame that has max/min buttons, but cannot be resized

**3 Dialog** a GUI window frame that cannot be resized and does not have max/min buttons

The default is 1.

#### **GUI\_NETWORK**

Select the network protocol to be used for the GUI session. Valid values are:

**1 TCP/IP** Transmission Control Protocol/Internet Protocol

**2 APPC** Advanced Program-to-Program Communication

#### **3 ISPDTPRF file**

This option tells ISPF to look for the file allocated to DD ISPDTPRF. ISPF will search this file and use the network protocol and address specified for your userid.

The default is 1.

#### **GUI\_SWITCH**

A connection to the workstation will be established and subsequent panels will be displayed in GUI mode after the ENTER key is pressed. Valid values are:

**1** GUI display

**2** Non-GUI display

The default is 1, GUI display.

#### **SAVE\_GUI\_VALUES**

Save the values entered on ISPF Settings Initiate the Workstation Connection panel in the system profile. Valid values are YES or NO.

The default value is YES.

#### **GUI\_TCP\_ADDRESS**

The TCP/IP address of the workstation in which the GUI display or connection should be made. The TCP/IP address can be in dotted decimal format or in domain name format.

The default is NONE.

#### **GUI\_TCP\_ADDRESS2**

The TCP/IP address of the workstation in which the GUI display or connection should be made. The TCP/IP address can be in dotted decimal format or in domain name format.

The default is NONE.

#### **GUI\_TCP\_ADDRESS3**

The TCP/IP address of the workstation in which the GUI display or connection should be made. The TCP/IP address can be in dotted decimal format or in domain name format.

The default is NONE.

#### **GUI\_TCP\_ADDRESS4**

The TCP/IP address of the workstation in which the GUI display or connection should be made. The TCP/IP address can be in dotted decimal format or in domain name format.

The default is NONE.

### **GUI\_TCP\_ADDRESS5**

The TCP/IP address of the workstation in which the GUI display or connection should be made. The TCP/IP address can be in dotted decimal format or in domain name format.

The default is NONE.

### **GUI\_TITLE**

The GUI title is the default value to be used for the title bar. If the GUI title is blank the title bar will be set to the userid from where the dialog was initiated.

The default is NONE (blank).

### **GUI\_ACCELERATOR\_SUPPORT**

When running in GUI mode and Accelerator Support is YES, any accelerator keys that are defined for the pulldown menus will be appended to the pulldown choice text and the accelerator keys will be functional. Valid values are YES or NO.

The default is YES.

### **HOST\_CODE\_PAGE and HOST\_CHARACTER\_SET**

Both the Code Page (CP) and the Character Set (CS) must be specified in order to be used. If specified they are used in translating the data from host to workstation. If the code page and character set are specified as 0000, values from the terminal query will be used. If your terminal/emulator does not support code pages, the CODEPAGE and CHARSET parameter values on ISPSTART will be used. The default will be English (U.S.) if these ISPSTART parameters are not specified.

The default is 0000 for both.

For more information about code pages and character sets refer to the *z/OS ISPF Dialog Developer's Guide and Reference*.

<b>CP &amp; CS</b>	<b>Used by countries</b>
0037 0697	United States, Canada, Netherlands, Portugal, Brazil, Australia, New Zealand
0273 0697	Austria, Germany
0280 0697	Italy
0277 0697	Denmark, Norway
0278 0697	Finland, Sweden
0284 0697	Spain, Latin America (Spanish)
0285 0697	United Kingdom
0297 0697	France
0500 0697	Belgium, Switzerland
0870 0959	Latin 2
0880 0960	Cyrillic
1025 1150	Cyrillic
290 1172	Japanese (Katakana)
1027 1172	Japanese (Latin)
833 1173	Korean

836 1174 Simplified Chinese

037 1175 Traditional Chinese

These Code Pages (CP) and the Character Sets (CS) support the Euro currency sign.

**CP & CS Used by countries**

1140 0695 United States, Canada, Netherlands, Portugal, Brazil, Australia, New Zealand

1141 0695 Austria, Germany

1144 0695 Italy

1145 0695 Spain, Latin America (Spanish)

1146 0695 United Kingdom

1147 0695 France

1148 0695 Belgium, Canada, Switzerland

1149 0695 Iceland

**RESET\_HOST\_CODE\_PAGE**

Reset the host code page value.

The default is NO.

**RESET\_HOST\_CHARACTER\_SET**

Reset the host character set value.

The default is NO.

**GUI\_DEFAULT\_BACKGROUND\_COLOR**

Allows you to specify the background color for the display of GUI panels. The valid values are: 1 (Dialog) and 2 (Standard).

The default is 1.

**GUI\_DOWNLOAD\_IMAGES**

When running in GUI mode and Download Images is YES, ISPF will attempt to download images from the host libraries defined to ddname ISPILIB. When Download Images is NO, ISPF will not download images from the host. Valid values are YES or NO.

The default is YES.

**GUI\_MAKEPATH\_FOR\_IMAGES**

When running in GUI mode and Make Path for Images is YES and the image path you have specified does not exist, ISPF will attempt to create the path. Valid values are YES or NO.

The default is NO.

**CONTINUE\_3270\_AFTER\_LOSS\_OF\_WS\_CONNECTION**

If the "3270 Mode after losing workstation connection" setting is selected, the ISPF session continues in the 3270 emulator session rather than abend with code 989 if the workstation connection is lost while running in GUI mode. Valid values are YES and NO.

The default is YES.

---

## ISPSPROF workstation download defaults

### **WSA\_DOWNLOAD\_OPTION**

Workstation agent download option. The valid values are:

- 1 Download now
- 2 Do not download
- 3 Cease warnings

The default is 1.

### **WSA\_DOWN\_LEVEL\_WARNING**

Warn when a user's workstation agent is at an earlier level than the host's version. Valid values are YES and NO.

The default is YES.(issue a warning).

### **WSA\_DOWNLOAD\_FTP\_ADDRESS**

The IP address of the workstation agent FTP server.

The default is NONE.

### **BROWSE\_FTP\_ERRORS**

Browse FTP output only if an error occurs in workstation agent download. Valid values are YES or NO.

The default is YES.

### **CREATE\_DIRECTORY\_ON\_WSA\_DOWNLOAD**

Create a directory on the workstation agent download. Valid values are YES or NO.

The default is NO.

### **WSA\_DOWNLOAD\_DIRECTORY**

The directory to copy the file to on a workstation agent download.

The default is NONE (blank).

### **WSA\_DOWNLOAD\_METHOD**

There are three methods for downloading the workstation agent, and four values that can be placed in this keyword:

- 1 Download using FTP
- 2 Download using ISPF/Client
- 3 Download manually
- NONE Not using the Workstation Agent

The default is NONE.

### **WSA\_DOWNLOAD\_DATASET**

The data set to copy the file from on the workstation agent download. The host data set that contains the workstation agent install programs.

The default is 'ISP.SISPGUI'.

### **RESET\_WSA\_DOWNLOAD\_DATASET**

Reset the workstation agent download data set.

The default is NO.

### **WSA\_DOWNLOAD\_PLATFORM**

The workstation agent runs on different workstation platforms. Indicate which platform you are using. Valid values are:

- NONE Workstation agent not being used
- 1 Windows 2000/NT

- 2 Reserved
- 3 AIX
- 4 Solaris
- 5 HP UX

The default is NONE.

**WSA\_DOWNLOAD\_FTP\_USER**

The name of the workstation agent download FTP user.

The default is NONE.

---

## Chapter 10. Exits

This section lists the exit macros and installation-wide exits.

---

### Exit macros

You define the ISPF installation-wide exits and installation-written exit routines you want to use by placing entries in the exit definition table, ISPXDT. ISPXDT consists of two sections:

#### Exit entry definitions

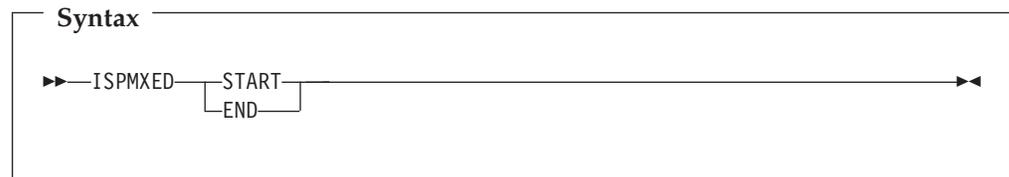
Defines the installation-wide exits and their associated exit routines

#### Exit data area definitions

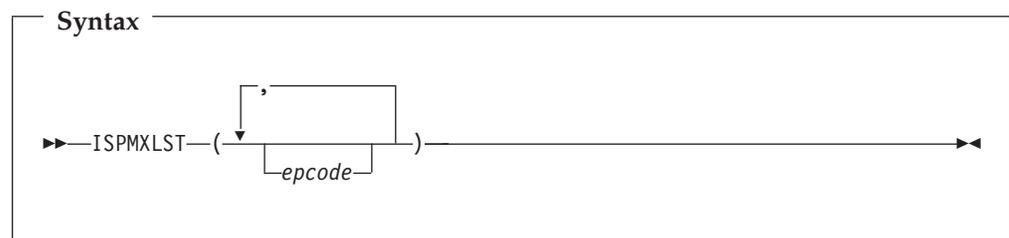
Defines the data areas that are used by the exit routines and their size specifications

The ISPMXED, ISPMXLST, ISPMXDEF, ISPMEXT, and ISPMXEND macros define the exit entry definition section of the table, and the ISPMXDD and ISPMDDAD macros define the exit data area definitions.

**ISPMXED** Defines the start or the end of the exit entry definition section.



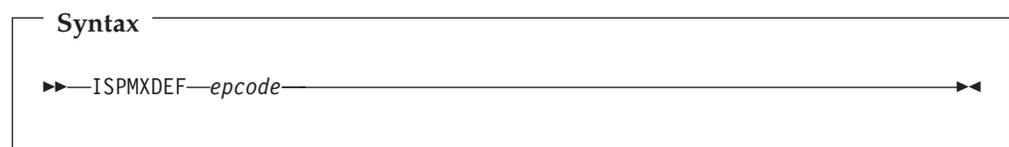
**ISPMXLST** Defines the installation-wide exits where you provide exit routines.



where:

*epcode* The numeric code for a user exit provided by ISPF. The list of codes must be enclosed in parentheses, and must be in ascending order. See Table 25 on page 113 for a list of the numeric codes.

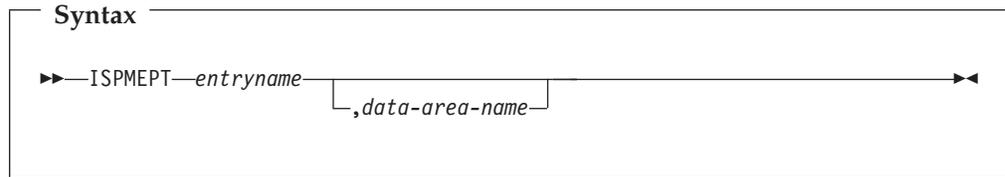
**ISPMXDEF** Begins the definition of exit routines for a particular user exit.



where:

*opcode* The numeric code for an exit point provided by ISPF. You must include this code as an operand for the ISPMXLST macro in order for the user exit to be defined. See Table 25 on page 113 for a list of the numeric codes.

**ISPMEPT** Identifies an exit routine to call at a particular user exit.



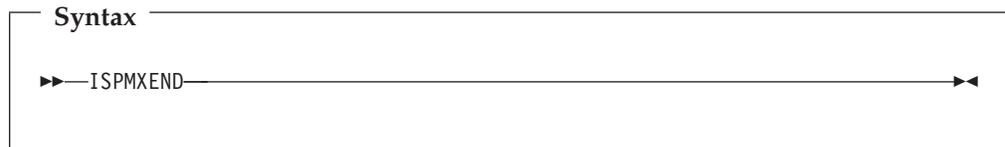
where:

*entryname* Identifies the entry point of the exit routine. This is usually a CSECT name.

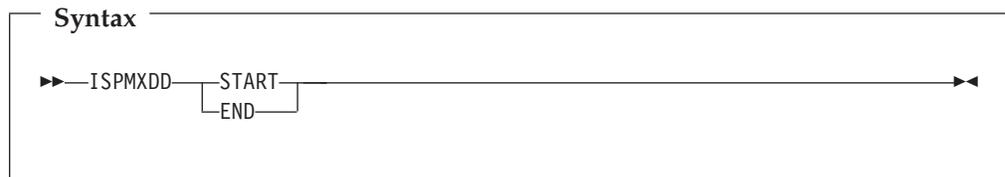
*data-area-name* Identifies the name of the data area the exit routine uses. The name can be up to 8 characters long. If you list the data area on an ISPMEPT macro, you must define it using the ISPMDDAD macro. If you do not specify a data area on the ISPMEPT macro, ISPF does not provide a data area for the exit routine being defined.

**Note:** Do not use 'NULLAREA' as a *data-area-name*.

**ISPMXEND** Ends the definition of routines for a particular user exit. This macro explicitly ends the ISPMXDEF macro and must be included.



**ISPMXDD** Indicates the start or the end of the data-area definition section of ISPXDT. Even if an exit routine does not require a data area, you must code the ISPMXDD START and ISPMXDD END macros.



**ISPMDDAD** Defines a data area and its size. The data area can be used by one or more exit routines.

### Syntax

▶▶ ISPMAD *data-area-name, size* ▶▶

where:

*data-area-name* Identifies the name of the data area an exit routine uses. The name can be up to 8 characters long.

*size* Specifies, in bytes, the size of the data area. If the size is not a multiple of eight, ISPF rounds it up to the next multiple of eight.

---

## Exit 1: ISPF session initialization exit

The ISPF session initialization exit provides accounting and monitoring capabilities. ISPF gives control to the exit routine after successfully opening all required data sets, determining session language, and before the first logical screen is started.

### Exit parameters

In addition to the standard exit parameters described in “Exit parameter list” on page 118, the exit routines at this user exit receive these parameters.

**Flags** 4 bytes of bit flags defined as follows:

- 0 1 = TEST option specified
- 1 1 = TESTX option specified
- 2 1 = TRACE option specified
- 3 1 = TRACEX option specified
- 4-31 Reserved.

**CPPL** Pointer to the address of the TSO command processor parameter list (CPPL) that was passed to ISPF.

**Lang** An 8-character field that contains the name of the national language for this ISPF session. The value is left-justified and padded with blanks.

### Return codes

No return codes are acknowledged at this user exit. When this user exit returns to ISPF, normal processing continues.

**Note:** For multiple exit routines, return codes still affect the processing flow.

---

## Exit 2: ISPF session termination exit

This user exit also provides accounting and monitoring capabilities. ISPF invokes routines at this exit after the last logical screen terminates and just before ISPF terminates.

If the ISPF main task terminates abnormally, routines at this user exit are not invoked. You can create an ESTAE exit in the ISPF initialization exit if you want to detect abnormal product termination. However, this applies only to product abnormal termination and not to abends of the logical screen task.

## Exit parameters

In addition to the standard exit parameters described in “Exit parameter list” on page 118, the exit routines at this user exit receive these parameters.

### ISPF Return Code:

A fullword binary number that contains the value of ZISPFRC as set by the application on the ISPSTART command or by ISPF. See the *z/OS ISPF Dialog Developer's Guide and Reference* for return codes set by ISPF. Any change made to this parameter by the exit routine is ignored.

## Return codes

No return codes are acknowledged at this user exit. ISPF termination continues upon return from the exit routine.

**Note:** For multiple exit routines, return codes still affect the processing flow.

---

## Exit 3: SELECT service start exit

In addition to providing information you can use to monitor ISPF, this user exit lets you restrict access to applications selected through ISPF. ISPF invokes the exit routines when the SELECT service is invoked. The SELECT service is invoked by:

- ISPSTART command
- SPLIT command (not when already in split-screen mode)
- SPLIT NEW command
- START command
- ISPSTRT program interface
- Selection menu entry
- Command table entry
- SELECT dialog service

The exit routines are given control after the SELECT request has been parsed and syntax checking is complete.

Changes you make to the Screen name are **not** reflected in the SCRNAME value passed in this exit.

## Exit parameters

In addition to the standard exit parameters described in “Exit parameter list” on page 118, the exit routines at this user exit receive these parameters. ISPF ignores any changes the exit routines make to these parameters.

**Flags** 4 bytes of bit flags defined as follows:

- |     |  |
|-----|--|
| 0   | 1 = PGM keyword specified                |
| 1   | 1 = CMD keyword specified                |
| 2   | 1 = PANEL keyword specified              |
| 3-4 | Reserved                                 |
| 5   | 1 = ADDPOP keyword specified             |
| 6   | 1 = BARRIER keyword specified            |
| 7   | 1 = NEST keyword specified               |
| 8   | 1 = NEWAPPL keyword specified            |
| 9   | 1 = NEWPOOL or NEWAPPL keyword specified |

- 10 1 = PASSLIB keyword specified
- 11 1 = Lang (CREX) keyword specified
- 12 1 = Lang (APL) keyword specified
- 13 1 = MODE(FSCR) specified
- 14 1 = MODE(LINE) specified
- 15 Reserved
- 16 1 = PARM value passed to the exit was truncated to 258 bytes, which include a halfword length field and 256 bytes of data.
- 17 1 = SUSPEND specified
- 18 1 = LOGO keyword specified
- 19-31 Reserved.

**elemname**

An 8-character field that contains the name of the element (PGM, CMD, or PANEL) to be selected.

**APPLID**

A 4-character field that contains the current application ID. If NEWAPPL was specified on the SELECT request, this field contains the new APPLID as specified. If APPLID field is blank, the previously specified application ID is implied. The value is left justified and padded with blanks.

**PARM** Input parameters being passed to a program dialog in the same format as when passed to the program; single character string preceded by a halfword containing its length (the length value does not include itself). FLAGS bit 16 is set when this parameter represents a truncated PARM value. The Length field is not affected by truncation.

**logoname**

An 8-character field that contains the name of the LOGO panel if it was specified.

**Screen name**

Screen name as set by the SCRNAME command and shown by the SCRNAME ON function.

## Return codes

- 0 Normal completion; ISPF continues processing.
- 8 Authorization failure; The SELECT request is not processed. Instead, ISPF issues a line mode message stating that the user exit indicated an authorization failure and terminates the select service, but allows the active application to continue.
- 16 Authorization failure; the SELECT request is not processed. Instead, ISPF issues a message indicating that the user exit indicated an authorization failure and terminates the SELECT service processing with a severe error (RC=20).
- Other** ISPF treats the error as severe and issues a message indicating that the exit routine returned an incorrect return code.

---

## Exit 4: select service end exit

You can use this user exit to mark the end of a program, command, or menu invoked through any of the SELECT services, with or without the SELECT service start exit.

Changes you make to the screen name are not reflected in the SCRNAME value passed in this exit.

### Exit parameters

In addition to the standard exit parameters described in “Exit parameter list” on page 118, the exit routines at this user exit receive these parameters. ISPF ignores any changes the exit routines make to the parameters.

**Flags** 4 bytes of bit flags defined as follows:

- 0 1 = PGM keyword specified
- 1 1 = CMD keyword specified
- 2 1 = PANEL keyword specified
- 3-4 Reserved
- 5 1 = ADDPOP keyword specified
- 6 1 = BARRIER keyword specified
- 7 1 = NEST keyword specified
- 8 1 = new application being invoked
- 9 1 = new shared pool is to be created
- 10 1 = PASSLIB keyword specified
- 11 1 = Lang (CREX) keyword specified
- 12 1 = APL keyword specified
- 13 1 = MODE(FSCR) specified
- 14 1 = MODE(LINE) specified
- 15-16 Reserved
- 17 1 = SUSPEND specified
- 18 1 = LOGO keyword specified
- 19-31 Reserved.

**Note:** If no parameters were passed, the flag bytes will be zero.

#### **elemname**

An 8-character field that contains the name of the element (PGM, CMD, or PANEL) to be selected. In case of CMD, if the element is a CLIST prefixed with %, the % symbol is removed from the elemname.

#### **APPLID**

A 4-character field that contains the application ID of the element that just terminated.

#### **STATS**

Two contiguous fullwords containing:

Total “think time” in hundredths of a second for this SELECT level only. “Think time” means all of the time intervals between the time when ISPF unlocked the keyboard for input and the time when input was received by ISPF. If there are other SELECTs nested within this one, their statistics are given at the corresponding select end exits, but not included here.

The number of times the screen was read by ISPF in this SELECT level.

**logoname**

An 8-character field that contains the name of the LOGO panel if it was specified.

**Screen name**

Screen name as set by the SCRNAME command and shown by the SCRNAME ON function.

## Return codes

No return codes are acknowledged at this user exit. ISPF termination continues upon return from the exit routine.

**Note:** For multiple exit routines, return codes still affect the processing flow.

---

## Exit 5: TSO command start exit

You can use this user exit to monitor and restrict commands invoked through ISPF. In addition, you can also use the user exit to allow TSO commands newly added to the system to be invoked from within ISPF without updating the ISPF TSO command table (ISPTCM). If the invoked command is not an implicit CLIST (not prefixed by %), the TCM is searched. This exit is called immediately after searching the TCM. For implicit CLISTs, the exit is called before attaching the exec processor.

**Note:** This user exit will not be invoked for TSO commands issued from REXX execs.

## Exit parameters

In addition to the standard exit parameters described in “Exit parameter list” on page 118, the exit routines at this user exit receive these parameters:

- |                   |   |
|-------------------|---|
| <b>cmdname</b>    | An 8-character field that contains the name of the command that was invoked. The name is left-justified within the field and padded with blanks.<br><br>If the command is a CLIST, this field contains the CLIST name. The CLIST name will not include the % prefix if one was used.  |
| <b>user flags</b> | 1 or more bytes of installation data as defined when ISPTCM is generated. See “Customizing the ISPF TSO command table (ISPTCM)” on page 106 for more information.   |
| <b>TCM flags</b>  | The flag byte in ISPTCM that ISPF uses to determine what processing should be done for the command. If the current command is not in ISPTCM, the default flag byte is provided. The exit routine can change this value and ISPF uses the value if it receives a return code of 4. See “Customizing the ISPF TSO command table (ISPTCM)” on page 106 and “ISPTCM usage notes” on page 109 for information about TCM flags. |
| <b>CPPL</b>       | The address of the TSO command processor parameter list (CPPL) that was passed to ISPF.   |

When the command is an implicit CLIST (prefixed by %), the TCM is not searched. In those cases, the user flags and TCM flags parameters will be binary zeros.

## Return codes

- 0** Normal completion; ISPF continues processing.
- 4** The exit routine has changed the value of the TCM flag byte. In this case, ISPF uses the value of the flag to process the command. However, changes to the TCM flag bit indicating whether the command should be logged are ignored.
- 16** Authorization failure. The exit routine indicated that this command should not be attached. ISPF issues an error message indicating that the exit routine rejected the command and returns to the caller.
- Other** ISPF treats the error as severe and issues a message indicating that the exit routine returned an incorrect return code.

---

## Exit 6: TSO command end exit

You can use this user exit to monitor TSO commands invoked from within ISPF. This exit is called after the attached command completes (or if attach fails).

**Note:** This user exit will not be invoked for TSO commands issued from REXX execs.

## Exit parameters

In addition to the standard exit parameters described in “Exit parameter list” on page 118, the exit routines at this user exit receive these parameters. ISPF ignores any changes made to these parameters.

- cmdname** An 8-character field that contains the name of the command to be attached. The name is left-justified within the field.
- CLIST name** An 8-character field that contains the name of the CLIST if the command was a CLIST. In that case, cmdname is 'EXEC'. If the command is not a CLIST, this field is the same as cmdname.
- user flags** 1 or more bytes of installation data as defined when ISPTCM is generated. See “Customizing the ISPF TSO command table (ISPTCM)” on page 106 for more information.
- TCM flags** The flag byte in ISPTCM that ISPF uses to determine what processing should be done for the command. If the current command is not in ISPTCM, the default flag byte is provided.
- CPPL** The address of the TSO command processor parameter list (CPPL) that was passed to ISPF.

When the command is an implicit CLIST (prefixed by %), the TCM is not searched. In those cases, the user flags and TCM flags parameters will be binary zeros.

## Return codes

No return codes are acknowledged at this user exit. When this exit returns to ISPF, normal processing continues.

**Note:** For multiple exit routines, return codes still affect the processing flow.

---

## Exit 7: LIBDEF service exit

This user exit lets you restrict the use of the LIBDEF service. ISPF passes control to the exit routines at this user exit after determining that a syntactically valid call has been made and before allocating and opening the alternate library.

### Exit parameters

In addition to the standard exit parameters described in “Exit parameter list” on page 118, the exit routines at this user exit receive these parameters. ISPF ignores any changes the exit routines make to the parameters.

#### libtype

An 8-character field that contains the library type as specified on the LIBDEF request. The value is left-justified and padded with blanks.

**Flags** 4 bytes of bit flags defined as follows:

- 0 1 = DATASET keyword specified
- 1 1 = LIBRARY keyword specified
- 2 1 = EXCLDATA keyword specified
- 3 1 = EXCLLIBR keyword specified
- 4 1 = COND request
- 5 1 = STACK request. If bit 4 and 5 are both 0, an UNCOND request was done
- 6 1 = STKADD request. Valid if bit 0 equals 1
- 7-31 Reserved

#### dsname#

A fullword binary number indicating the number of elements in the lengths and names arrays that follow. The number will not exceed 15.

#### lengths

An array of fullwords indicating the lengths of the corresponding elements of the names array. The maximum length of a dsname is 44. If a libname is provided using the LIBRARY or EXCLLIBR keyword, the length will be eight.

**names** An array containing the data set names or library name as specified in the ID parameter of the LIBDEF service. Data set names are fully qualified without quotes. Each element can be up to 44 characters long and those names less than 44 characters are padded on the right with blanks.

### Return codes

- 0 Normal completion; ISPF continues processing.
- 16 Authorization failure. ISPF issues a message indicating an authorization failure has occurred and LIBDEF terminates with a severe error.
- Other LIBDEF treats return codes other than 0 or 16 as a severe error and returns a code of 20 to the caller.

---

## Exit 8: RESERVE exit

This user exit lets you use your own method for serializing resources in addition to the RESERVE done by ISPF. See Appendix A, "ISPF enqueue processing for data integrity," on page 305 for information about ISPF's use of ENQ and RESERVE. ISPF gives control to the exit routine before it does the RESERVE. If your serialization mechanism cannot acquire the resource, or if the exit routine returns a code greater than 0, ISPF does not attempt to get it.

### Exit parameters

In addition to the standard exit parameters described in "Exit parameter list" on page 118, the exit routines at this user exit receive these parameters. ISPF ignores any changes made to these parameters.

**qname**

An 8-character field that contains the queue name against which the RESERVE is done. (ISPF uses a qname of 'SPFEDIT'.)

**rname** The 44-character resource name (data set name) that is reserved.

**ucbaddr**

A fullword field containing the address of the unit control block (UCB) for the reserved device.

**Flags** 4 bytes of bit flags defined as follows:

- 0        1 = Exclusive reserve requested. 0 = Shared reserve requested.  
(ISPF always requests exclusive use of the resource at RESERVE.)
- 1        1 = A RESERVE or ENQ will be done on qname 'SYSIEWLP' (the linkage editor qname).
- 2        1 = The resource being reserved is a partitioned data set.
- 3        1 = The resource being reserved is a sequential data set.
- 4-31    Reserved.

**Note:** The exit routine should not use the same (qname, rname) combination (in a RESERVE macro call) that ISPF uses. If that happens, an abend occurs when ISPF attempts to do its own RESERVE.

### Return codes

- 0        Normal completion; ISPF does its own RESERVE and continues processing.
- 16       Resource not available. ISPF issues a message showing that the exit routine indicated the resource is not available. This results in a failure of the service requesting the RESERVE.
- Other** ISPF treats other return codes similar to a return code of 16 and issues a message indicating that the exit routine returned an incorrect return code.

---

## Exit 9: RELEASE exit

Use this user exit to release any resources acquired at the RESERVE user exit. ISPF relies on task termination to release any resources reserved by the abending logical screen task. This user exit does not get control in the case of abnormal termination of the logical screen.

If your serialization mechanism is such that task termination will not release the reserved resources, an exit routine is provided at the logical screen end user exit to clean up any unreleased resources for that task.

The RESERVE done by ISPF is released before the exit routine is given control.

## Exit parameters

In addition to the standard exit parameters described in “Exit parameter list” on page 118, the exit routines at this user exit receive these parameters. ISPF ignores any changes made to these parameters.

### qname

An 8-character field that contains the queue name against which the RESERVE is done. (ISPF uses a qname of 'SPFEDIT'.)

**rname** The 44-character resource name (data set name) that is reserved.

## Return codes

No return codes are acknowledged at this user exit. When this exit returns to ISPF, normal processing continues.

**Note:** For multiple exit routines, return codes still affect the processing flow.

---

## Exit 10: logical screen start exit

You can use this user exit to gather accounting and monitoring information for each ISPF logical screen. The exit routine is given control just before the logical screen task is attached.

## Exit parameters

This user exit uses only the standard exit parameters described in “Exit parameter list” on page 118, which includes the logical screen identifier (screenid).

## Return codes

No return codes are acknowledged at this user exit. When this exit returns to ISPF, normal processing continues.

**Note:** For multiple exit routines, return codes still affect the processing flow.

---

## Exit 11: logical screen end exit

This user exit, which is similar to the logical screen start user exit, lets you gather accounting and monitoring information for each ISPF logical screen. It gives you control for both normal and abnormal termination of a logical screen. You can use it to perform necessary cleanup required as a result of exits being bypassed because of abnormal termination. Specifically, these end user exits cannot gain control because of a logical screen abend even though the corresponding start installation exits did get control:

- SELECT service end exit
- RELEASE exit
- ISPF service end exit.

**Note:** The TSO command end exit gets control for both normal and abnormal termination of attached commands.

## Exit parameters

In addition to the standard exit parameters described in “Exit parameter list” on page 118, the exit routines at this user exit receive these parameters:

**Flags** 4 bytes of bit flags defined as follows:

0 0 = Normal termination 1 = Abnormal termination.

1-31 Reserved

**Next logical screen**

An 8-character field that identifies the next active ISPF logical screen. In the case of exiting ISPF, the next logical screen ID will be 0.

## Return codes

No return codes are acknowledged at this user exit. When this exit returns to ISPF, normal processing continues.

**Note:** For multiple exit routines, return codes still affect the processing flow.

---

## Exit 12: ISPF service start exit

The ISPF exit for the service start lets you monitor all external ISPF service requests. Exit routines at this user exit are notified of all service requests (including PDF services) made through the ISPLINK or ISPEXEC interfaces. The exit routines are not notified for ISPF internal service requests (those that do not use the ISPLINK or ISPEXEC interfaces.)

After initial verification of parameters and syntax, ISPF calls the exit routines at this user exit before the requested service is performed.

## Exit parameters

In addition to the standard exit parameters described in “Exit parameter list” on page 118, the exit routines at this user exit receive these parameters. ISPF ignores any changes the routines make to these parameters.

**servname**

An 8-character field containing the name of the ISPF service being invoked. The name is left-justified within the field.

**Flags** 4 bytes of bit flags defined as follows:

0 1 = ISPF service 0 = PDF service

1-31 Reserved.

## Return codes

No return codes are acknowledged at this user exit. When this exit returns to ISPF, normal processing continues.

**Note:** For multiple exit routines, return codes still affect the processing flow.

---

## Exit 13: ISPF service end exit

You can use this user exit to mark the termination of ISPF dialog services invoked through the ISPLINK or ISPEXEC interfaces.

If a severe error occurs causing a logical screen abend, ISPF does not give control to the exit routines at this user exit. You can also stack exit routines at the logical screen end user exit to ensure that service termination is correctly recorded.

## Exit parameters

In addition to the standard exit parameters described in “Exit parameter list” on page 118, the exit routines at this user exit receive these parameters. ISPF ignores any changes the exit routines make to the parameters.

### servname

An 8-character field containing the name of the ISPF or PDF service being invoked. The name is left-justified within the field.

**Flags** 4 bytes of bit flags defined as follows:

0        1 = ISPF service 0 = PDF service

1-31    Reserved.

## Return codes

No return codes are acknowledged at this user exit. When this exit returns to ISPF, normal processing continues.

**Note:** For multiple exit routines, return codes still affect the processing flow.

---

## Exit 14: SWAP exit

Using this user exit, you can indicate a change of the active logical screen. Together with the logical screen start and end user exits, it allows resource use to be monitored for each ISPF logical screen.

ISPF calls the exit routines at this user exit just after the logical screen to be given control is activated. In addition, ISPF calls the exit routines if the SPLIT command is entered when the screen is already split.

## Exit parameters

This user exit uses only the standard exit parameters described in “Exit parameter list” on page 118, which includes the logical screen identifier (screenid). The screen ID identifies the logical screen activated as a result of the SWAP command.

## Return codes

No return codes are acknowledged at this user exit. When this exit returns to ISPF, normal processing continues.

**Note:** For multiple exit routines, return codes still affect the processing flow.

---

## Exit 15: DISPLAY service exit

This user exit is provided for installation tailoring purposes. This allows you to selectively replace ISPF, PDF, or other ISPF-based panels with your own versions of the panels. You can control the amount of information presented to your users based on their experience with the panels. The exit routines can change a number of parameters at this user exit.

ISPF calls the exit routines at this user exit before the display of all the panels (as a result of internal or external display requests) except for the display of severe error

or abend panels or display requests where a panel ID is not specified. This includes displays caused by the DISPLAY, TBDISPL, EDIT, SELECT, and BROWSE services.

Changes you make to the Screen name are **not** reflected in the SCRNAME value passed in this exit.

## Exit parameters

In addition to the standard exit parameters described in “Exit parameter list” on page 118, the exit routines at this user exit receive these parameters.

- panel-id** An 8-character field that contains the name of the panel to be displayed. If the display request did not specify a panel-id (indicating that the previously displayed panel is to be redisplayed), the exit routine is not invoked.
- This parameter can be changed. If the parameter is changed, the name of the panel in the field must be left-justified.
- message-id** An 8-character field that contains the message-id to be displayed on the panel, as specified on the display request. Messages displayed as a result of the SETMSG service are not identified in this field. If a message-id is not specified, the parameter contains blanks.
- This parameter can be changed. If the parameter is changed, the message-id in the field must be left-justified.
- cursor-field** An 8-character field that contains the field name on which the cursor is to be positioned. This field contains blanks if the cursor-field is not explicitly specified (the cursor is placed by defaults).
- This parameter can be changed. If the parameter is changed, the field name must be left-justified.
- cursor-offset** A fullword binary number that contains the offset within the cursor-field where the cursor is to be positioned. If not explicitly specified on the display request, this parameter has a value of zero.
- This parameter can be changed.
- table-name** An 8-character field that contains the name of the table to be displayed if the display request is a result of the TBDISPL service. Otherwise, the field contains blanks.
- This parameter *cannot* be changed.
- Flags** 4 bytes of bit flags defined as follows:
- |      |   |
|------|---|
| 0    | 1 = Non-display mode is active; the panel is processed but not displayed. |
| 1    | 1 = COMMAND option was specified on the DISPLAY request.                  |
| 2-31 | Reserved.   |
- message-field** An 8-character field that contains the name of the panel field the message pop-up window is to be positioned adjacent to. This field contains blanks if the MSGLOC parameter is not specified.
- This parameter can be changed by the exit routine.

**Screen name** Screen name as set by the SCRNAME command and shown by the SCRNAME ON function.

## Return codes

- 0** Normal completion; ISPF continues processing.
- 4** The exit routine changed one or more of the parameters. ISPF continues processing using the changed parameter values.
- Other** ISPF treats the error as severe and issues a message indicating that the exit routine issued an incorrect return code.

**Note:** Panel functions, such as RESP, that are coded in the panel )INIT section are not processed before the exit is entered. This could prevent a display although non-display is not indicated by the exit 15 parameter.

---

## Exit 16: log, list, trace, and temporary data set allocation exit

This user exit lets you maintain your own data set naming conventions. ISPF calls the routines at this user exit before allocating the log, list, or temporary control data sets. As a result, you can provide a prefix for the name of the data set to be allocated. However, if the data set has been preallocated, ISPF does not use this prefix.

The exit routine can provide a prefix up to 26 characters long. A zero length prefix is flagged as an error. ISPF reserves the remaining 18 characters of the data set name for its own use.

ISPF builds the names of the log, list, and temporary control data sets according to these rules:

1. If a data set prefix is specified in the TSO user profile table (UPT) and it is different from the user ID, the data set name is of the form:  
uotpfx.userid.ISPF-specific-suffix
2. If a prefix is not specified in the UPT or if it is the same as the TSO user ID, the data set name is:  
userid.ISPF-specific-suffix
3. If a user ID is not available (executing ISPF in BATCH), ISPF recommends that the TSO PROFILE command be used to place the user ID in the UPT prefix field. In that case, the data set name has the this form:  
uotpfx.ISPF-specific-suffix

Note that UPTPFX is assumed to be the user ID in this case.

If you provide an exit routine at this user exit, ISPF allows the data set names to be of the form:

Exit-provided-prefix.ISPF-specific-suffix

The 18 characters reserved by ISPF begin with the period separator.

Because the user ID is also passed to the exit routine as part of the prefix, the exit routine is responsible for maintaining unique data set names.

## Exit parameters

In addition to the standard exit parameters described in “Exit parameter list” on page 118, the exit routines at this user exit receive these parameters.

**prefix-len**

A fullword binary number that identifies the length of the prefix field. On entry to the exit routine, it is the length of the prefix including the UPT prefix, the user ID or both. On return to ISPF, it should contain the length of the prefix provided by the exit routine.

The value of this parameter must be in the range 1 to 26, inclusive.

**prefix** A 26-character field that contains the data set name prefix used by ISPF. On entry to the exit routine, it contains the UPT prefix, the user ID, or both left-justified within the field.

On return to ISPF, it can contain any prefix (up to 26 characters) chosen by the exit. ISPF does not do any validity checking of the specified prefix. This prefix must be left-justified. If an incorrect prefix is provided, allocation of the data set fails.

**suffix-type**

A fullword of bit flags indicating the type of data set that ISPF allocates:

- 0        1 = List data set
- 1        1 = Log data set
- 2        1 = Temporary listing data set
- 3        1 = Temporary control data set
- 4        1 = Temporary work data set
- 5        1 = ISPVCALL trace data set
- 6        1 = ISPDPTRC trace data set
- 7        1 = ISPFTRC trace data set
- 8-31    Reserved.

**suffix-len**

A fullword binary number containing the length of the value within the suffix field. ISPF ignores any changes the exit routine makes to this field.

**suffix** An 18-character field containing the name of the data set name suffix that ISPF uses. ISPF ignores any changes the exit routine makes to this field.

## Return codes

No return codes are acknowledged. Upon return to ISPF, the data set name is generated using the prefix provided by the exit routine (if any) and normal processing continues.

**Note:** For multiple exit routines, return codes still affect the processing flow.

---

## Chapter 11. PDF translation tables

---

### Translation table for valid data set name characters

The translation table for valid data set name characters (Table 38) specifies which characters are allowed in a data set name, as follows:

- Valid characters are represented with X'00'
- Invalid characters are represented with X'FF'.

*Table 38. Example of translation table for valid data set characters*

Table	Hexadecimal Code	Position
TTVDSN	DC X'FFFFFFFFFFFFFFFF'	(X'00' to X'07')
	DC X'FFFFFFFFFFFFFFFF'	(X'08' to X'0F')
	DC X'FFFFFFFFFFFFFFFF'	(X'10' to X'17')
	...	
	DC X'FFF0000000000000'	(X'78' to X'7F')
	DC X'FF00000000000000'	(X'80' to X'87')
	...	
	DC X'0000FFFFFFFFFFFF'	(X'E8' to X'EF')
	DC X'0000000000000000'	(X'F0' to X'F7')
	DC X'0000FFFFFFFFFFFF'	(X'F8' to X'FF')

---

### Translation table for invalid data set name characters

The translation table for invalid data set name characters (Table 39) specifies which characters are not allowed in a data set name, as follows:

- Valid characters are represented by their EBCDIC hexadecimal code
- Invalid characters are represented with X'00'.

*Table 39. Example of translation table for invalid data set name characters*

Table	Hexadecimal Code	Position
TTIDSN	DC X'0000000000000000'	(X'00' to X'07')
	DC X'0000000000000000'	(X'08' to X'0F')
	DC X'0000000000000000'	(X'10' to X'17')
	...	
	DC X'00007A7B7C000000'	(X'78' to X'7F')
	DC X'0081828384858687'	(X'80' to X'87')
	...	
	DC X'E8E9000000000000'	(X'E8' to X'EF')
	DC X'F1F2F3F4F5F6F700'	(X'F0' to X'F7')
	DC X'F8F9000000000000'	(X'F8' to X'FF')

---

### Translation table for hexadecimal characters

The translation table for hexadecimal characters (Table 40 on page 266) specifies the valid hexadecimal characters as follows:

- Valid characters are represented with X'00'
- Invalid characters are represented with X'FF'.

Table 40. Example of translation table for hexadecimal characters

Table	Hexadecimal Code	Position
TTHex	DC X'FFFFFFFFFFFFFFFF'	(X'00' to X'07')
	DC X'FFFFFFFFFFFFFFFF'	(X'08' to X'0F')
	DC X'FFFFFFFFFFFFFFFF'	(X'10' to X'17')
	...	
	DC X'FFFFFFFFFFFFFFFF'	(X'78' to X'7F')
	DC X'FF000000000000FF'	(X'80' to X'87')
	...	
	DC X'FFFFFFFFFFFFFFFF'	(X'E8' to X'EF')
	DC X'0000000000000000'	(X'F0' to X'F7')
	DC X'0000FFFFFFFFFFFF'	(X'F8' to X'FF')

## Translation table for numeric characters

The translation table for numeric characters (Table 41) specifies the valid numeric characters as follows:

- Valid characters are represented with X'00'.
- Invalid characters are represented with X'FF'.

Table 41. Example of translation table for numeric characters

Table	Hexadecimal Code	Position
TTNUM	DC X'FFFFFFFFFFFFFFFF'	(X'00' to X'07')
	DC X'FFFFFFFFFFFFFFFF'	(X'08' to X'0F')
	DC X'FFFFFFFFFFFFFFFF'	(X'10' to X'17')
	...	
	DC X'FFFFFFFFFFFFFFFF'	(X'78' to X'7F')
	DC X'FFFFFFFFFFFFFFFF'	(X'80' to X'87')
	...	
	DC X'FFFFFFFFFFFFFFFF'	(X'E8' to X'EF')
	DC X'0000000000000000'	(X'F0' to X'F7')
	DC X'0000FFFFFFFFFFFF'	(X'F8' to X'FF')

## Translation table for alphanumeric characters

The translation table for alphanumeric characters (Table 42) specifies the valid alphanumeric characters as follows:

- Valid characters are represented with X'00'.
- Invalid characters are represented with X'FF'.

Table 42. Example of translation table for alphanumeric characters

Table	Hexadecimal Code	Position
TTALN	DC X'FFFFFFFFFFFFFFFF'	(X'00' to X'07')
	DC X'FFFFFFFFFFFFFFFF'	(X'08' to X'0F')
	DC X'FFFFFFFFFFFFFFFF'	(X'10' to X'17')
	...	
	DC X'FFFFFFF0000FFFFFF'	(X'78' to X'7F')
	DC X'FF00000000000000'	(X'80' to X'87')
	...	
	DC X'0000FFFFFFFFFFFF'	(X'E8' to X'EF')
	DC X'0000000000000000'	(X'F0' to X'F7')
	DC X'0000FFFFFFFFFFFF'	(X'F8' to X'FF')

---

## Translation table for edit terminal output characters

The translation table for edit terminal output characters (Table 43) is used to translate invalid edit display characters to an attribute byte as follows:

- Valid characters are represented by their EBCDIC value
- Invalid characters are represented with X'15'.

Table 43. Example of translation table for edit terminal output character

Table	Hexadecimal Code	Position
TTETO	DC X'1515151515151515'	(X'00' to X'07')
	DC X'1515151515151515'	(X'08' to X'0F')
	DC X'1515151515151515'	(X'10' to X'17')
	...	
	DC X'15797A7B7C7D7E7F'	(X'78' to X'7F')
	DC X'1581828384858687'	(X'80' to X'87')
	...	
	DC X'E8E9151515151515'	(X'E8' to X'EF')
	DC X'F0F1F2F3F4F5F6F7'	(X'F0' to X'F7')
	DC X'F8F9151515151515'	(X'F8' to X'FF')

---

## Translation table for generic string characters

The translation table for generic string characters (Table 44) is used to assign a mask value to a character representing a subset of characters. The characters are defined in the Generic String Special Character (GSS) table, Table 45 on page 268.

Table 44. Example of translation table for generic string characters

Table	Hexadecimal Code (Mask Value)	Code Mask Offset
TTGSC	DC X'00'	X'00' Any character
	DC X'01'	X'01' Invalid characters
	DC X'02'	X'02' Special characters
	DC X'04'	X'03' APL/TEXT Special
	DC X'08'	X'04' APL/TEXT Alpha
	DC X'10'	X'05' Lower alpha
	DC X'20'	X'06' Upper alpha
	DC X'40'	X'07' Numeric
	DC X'80'	X'08' User defined character set
	DC X'FF'	X'09' (RESERVED)
	DC X'30'	X'0A' Alpha
	DC X'7F'	X'0B' Nonblank
	DC X'FF'	X'0C' (RESERVED)
	DC X'FF'	X'0D' (RESERVED)
	DC X'FF'	X'0E' (RESERVED)
	DC X'FF'	X'0F' (RESERVED)
	DC X'40'	X'10' Not numeric
	DC X'20'	X'11' Not upper
	DC X'10'	X'12' Not lower
	DC X'30'	X'13' Not alpha
	DC X'FF'	X'14' (RESERVED)
	DC X'FF'	X'15' (RESERVED)
	DC X'FF'	X'16' (RESERVED)

---

## Translation table for generic string special characters

The translation table for generic string special characters (Table 45 on page 268) is used to assign a code of X'01' to X'16' to generic string special characters according to the generic string character (GSC) table (Table 44) as follows:

- Numbers and letters translate to their EBCDIC hexadecimal codes.

- Other valid characters, that is, characters used to represent a character subset for Edit and Browse picture strings, are represented by the offset from the generic string character table corresponding to the subset they represent. In Table 45, (X'7B') has a value of '07' because '07' is the offset in the generic string character table for the subset of numeric characters, which the # is used to represent.
- Characters that are invalid in a generic string are represented with X'FF'.

Table 45. Example of translation table for generic string special characters

Table	Hexadecimal Code	Position
TTGSS	DC X'FFFFFFFFFFFFFFFF'	(X'00' to X'07')
	DC X'FFFFFFFFFFFFFFFF'	(X'08' to X'0F')
	DC X'FFFFFFFFFFFFFFFF'	(X'10' to X'17')
	...	
	DC X'FFFFFF070AFF00FF'	(X'78' to X'7F')
	DC X'FF81828384858687'	(X'80' to X'87')
	...	
	DC X'E8E9FFFFFFFFFFFF'	(X'E8' to X'EF')
	DC X'F0F1F2F3F4F5F6F7'	(X'F0' to X'F7')
	DC X'F8F9FFFFFFFFFFFF'	(X'F8' to X'FF')

## Usage notes for the GSC and GSS tables

The generic string special (GSS) table is used to determine if a character is valid in a picture string and, if so, what offset into the generic string character (GSC) table describes its subset. The mask value from the GSC table is used to determine which characters satisfy the subset. Each character in the text being scanned is represented in the generic string master (GSM) table by a hexadecimal code that indicates which subsets that character belongs to (see "Creating ISPF terminal translation tables" on page 86). The hexadecimal code in the GSM is ANDed with the mask value from the GSC. Any nonzero result is considered a match.

In Table 45, a # ('7B') has a value in the GSS of '07'. At offset '07' in the GSC (actually the eighth entry in the GSC, the first being offset '00'), the mask value is a X'40'. This means any character that is a member of the numeric subset, when ANDed with a X'40', will produce a nonzero result.

## Translation table for uppercase characters

The translation table for uppercase characters (Table 46) translates data as follows:

- Lowercase alphabetic characters translate to uppercase
- All other characters translate to themselves.

Table 46. Example of translation table for uppercase characters

Table	Hexadecimal Code	Position
TTUPP	DC X'0001020304050607'	(X'00' to X'07')
	DC X'08090A0B0C0D0E0F'	(X'08' to X'0F')
	DC X'1011121314151617'	(X'10' to X'17')
	...	
	DC X'78797A7B7C7D7E7F'	(X'78' to X'7F')
	DC X'80C1C2C3C4C5C6C7'	(X'80' to X'87')
	...	
	DC X'E8E9EAEBECEDEEEF'	(X'E8' to X'EF')
	DC X'F0F1F2F3F4F5F6F7'	(X'F0' to X'F7')
	DC X'F8F9FAFBFCFDFEFF'	(X'F8' to X'FF')

## Translation table for lowercase characters

The translation table for lowercase characters (Table 47) translates data as follows:

- Uppercase alphabetic characters translate to lowercase
- All other characters translate to themselves.

Table 47. Example of translation table for lowercase characters

Table	Hexadecimal Code	Position
TTLOW	DC X'0001020304050607'	(X'00' to X'07')
	DC X'08090A0B0C0D0E0F'	(X'08' to X'0F')
	DC X'1011121314151617'	(X'10' to X'17')
	...	
	DC X'78797A7B7C7D7E7F'	(X'78' to X'7F')
	DC X'8081828384858687'	(X'80' to X'87')
	...	
	DC X'A8A9EAECEDEEEF'	(X'E8' to X'EF')
	DC X'F0F1F2F3F4F5F6F7'	(X'F0' to X'F7')
	DC X'F8F9FAFBFCFDFEFF'	(X'F8' to X'FF')

## Modifying the GSM to use the user character subset

The generic string master (GSM) translation table (Table 48) and its related tables can be modified to add an additional character subset to be used in Edit picture string processing for the FIND and CHANGE commands. The GSM table is found in the ISPF translation tables, member ISPOWNTT in ISP.SISPSAMP.

## Generic string master translation table

The positions in the generic string master translation table (Table 48) are filled in as follows:

- X'00' Blank character
- X'01' Invalid character
- X'02' Special character
- X'04' APL/TEXT special characters (only for APL and TEXT keyboards)
- X'08' APL/TEXT alphabetic characters (only for APL and TEXT keyboards)
- X'10' Lowercase alphabetic character
- X'20' Uppercase alphabetic character
- X'40' Numeric character
- X'80' User character subset.

Table 48. Example of Generic String Master translation table

Table	Hexadecimal Code	Position
TTGSM	DC X'0101010101010101'	(X'00' to X'07')
	DC X'0101010101010101'	(X'08' to X'0F')
	DC X'0101010101010101'	(X'10' to X'17')
	...	
	DC X'0102020202020202'	(X'78' to X'7F')
	DC X'0110101010101010'	(X'80' to X'87')
	...	
	DC X'2020010101010101'	(X'E8' to X'EF')
	DC X'4040404040404040'	(X'F0' to X'F7')
	DC X'4040010101010101'	(X'F8' to X'FF')

To modify the GSM table to use a user character subset, follow these steps:

1. Choose a character to represent your subset. For example, Edit uses an @ to stand for alphabetic.

2. Modify the entry in the generic string special character (GSS) table that corresponds to the character you want to use so that it has a value of X'08'. This indicates where in the generic string character (GSC) table the mask for your character is located. The GSC does not need to be changed. It is initially set for user character sets.
3. Modify the GSM entries of those characters you want to include in your special character set so the high order bit is on.

For example:

If you want to define a character set of special attribute characters consisting of hexadecimal codes X'10' through X'17', and you want to use a ("") as the picture string identifier for them, you would:

1. Modify the entry for X'4F' in the GSS table so it has a value of X'08' as shown in Table 49. Compare to Table 45 on page 268.

*Table 49. Example of translation table for modified Generic String Special Characters*

Table	Hexadecimal Code	Position
TTGSS	DC X'FFFFFFFFFFFFFFFF'	(X'00' to X'07')
	DC X'FFFFFFFFFFFFFFFF'	(X'08' to X'0F')
	DC X'FFFFFFFFFFFFFFFF'	(X'10' to X'17')
	...	
	DC X'FFFFFFF0105FFFF08'	(X'78' to X'7F')
	DC X'FFFFFFFFFFFFFFFF'	(X'80' to X'87')
	...	
	DC X'E8E9FFFFFFFFFFFF'	(X'E8' to X'EF')
	DC X'F0F1F2F3F4F5F6F7'	(X'F0' to X'F7')
	DC X'F8F9FFFFFFFFFFFF'	(X'F8' to X'FF')

2. Modify the GSM entries for hexadecimal codes X'10' through X'17' to turn the high order bit on as shown in Table 50. Compare to Table 48 on page 269.

*Table 50. Example of modified Generic String Master translation table*

Table	Hexadecimal Code	Position
TTGSM	DC X'0101010101010101'	(X'00' to X'07')
	DC X'0101010101010101'	(X'08' to X'0F')
	DC X'8181818181818181'	(X'10' to X'17')
	...	
	DC X'0102020202020202'	(X'78' to X'7F')
	DC X'0110101010101010'	(X'80' to X'87')
	...	
	DC X'2020010101010101'	(X'E8' to X'EF')
	DC X'4040404040404040'	(X'F0' to X'F7')
	DC X'4040010101010101'	(X'F8' to X'FF')

You could locate the special attribute bytes by issuing the edit command FIND P"". If you do not want these bytes to be found under any other picture string, set the hexadecimal value to X'80'. These characters can be included in multiple character sets by setting the appropriate bits to on, according to the GSM table.

---

## ISPF and PDF terminal translation table relationship

The terminal translation table relationship for ISPF and PDF is shown in Table 51.

*Table 51. ISPF terminal translation table relationship*

---

<b>ISPF</b>	<b>PDF</b>
ISP3277 ISP3277A ISP3277K ISP3278	ISR3277 ISR3277A ISR3277K ISR3278
ISP3278A ISP3278C ISP3278K ISP3278T	ISR3278A ISR3278C ISR3278K ISR3278T
NEW32TBL	ISR32TBL

---

The delivered ISPF terminal table names start with the prefix "ISP". ISPF does not require that user-defined terminal table names begin with the prefix "ISP"; however, PDF terminal names require the "ISR" prefix. PDF searches for the load module beginning with the fourth position of the actual table name and prefixes it with "ISR".



---

## Chapter 12. Dialog development model listings

This topic lists all of the models included with PDF. The table shows the external model name, any qualifiers, a short description, and the internal member name in the SKELS library provided with PDF. See *z/OS ISPF Edit and Edit Macros* for more information about adding or changing a model.

---

### DM and PDF services in CLIST commands

Model Name		Description	Member Name
<b>Display</b>			
ADDFPOP		Display pop-up window	ISREMCD5
DISPLAY		Display option	ISREMCD1
PQUERY		Get panel information	ISREMCD4
REMPPOP		Remove pop-up window	ISREMCD6
SETMSG		Set message display	ISREMCD3
TBDISPL		Table display information	ISREMCD2
<b>File Tailoring</b>			
FTCLOSE		End file tailoring	ISREMCF3
FTERASE		File tailor erase	ISREMCF4
FTINCL		File tailor include skeleton	ISREMCF2
FTOPEN		File tailor open	ISREMCF1
<b>Library Access</b>			
DIRLIST		Displays z/OS UNIX directory list	ISREMCLD
LMCLOSE		Close a data set	ISREMCL1
LMCOMP		Compress a data set	ISREMCLU
LMCOPY		Copy a data set	ISREMCLQ
LMDDISP		Data set list	ISREMCLZ
LMDFREE		Release a data set list	ISREMCLW
LMDINIT		Establish a data set ID	ISREMCLX
LMDLIST		Obtain a list of data sets	ISREMCLV
LMERASE		Erase a data set or library	ISREMCL2
LMFREE		Release a data set	ISREMCL3
LMGET		Read a record	ISREMCL4
LMINIT		Establish a data ID	ISREMCL5
LMMADD		Add a member	ISREMCL6
LMMDEL		Delete a member	ISREMCL7
LMMDISP		Display member list	ISREMCL0
LMMFIND		Find a member	ISREMCL8
LMMLIST		Create a member list	ISREMCL9
LMMOVE		Move a data set or member	ISREMCLP

Model Name		Description	Member Name
LMMREN		Rename a member	ISREMCLA
LMMREP		Replace a member	ISREMCLB
LMMSTATS		Set member statistics	ISREMCLR
LMOPEN		Open a data set	ISREMCLC
LMPRINT		Write member to list data set	ISREMCLT
LMPUT		Write a record	ISREMCLF
LMQUERY		Provide data set information	ISREMCLG
LMRENAME		Rename a library	ISREMCLH
MEMLIST		Displays Option 3.1 member list	
<b>Miscellaneous</b>			
BROWSE		Browse service	ISREMCM3
CONTROL		Control service	ISREMCM2
DSINFO		Returns data set information	ISREMCM4
EDIT		Edit service	ISREMCM7
EDREC		Edit recovery services	ISREMCM6
GETMSG		Get message service	ISREMCM8
LIBDEF		LIBDEF service	ISREMCM5
LIST		Write data to list data set	ISREMCMA
LOG		Write message or log data set	ISRECMC
QBASELIB		Query base library information	ISREMCMD
QLIBDEF		Query LIBDEF library information	ISREMCMD
SELECT		Select service	ISREMCMD
<b>Table Functions (General)</b>			
TBCLOSE		Table close	ISREMC5
TBCREATE		Table create	ISREMC1
TBEND		Table end	ISREMC6
TBERASE		Table erase	ISREMC7
TBOPEN		Table open	ISREMC2
TBQUERY		Table query	ISREMC3
TBSAVE		Table save	ISREMC4
TBSTATS		Table statistics	ISREMC8
<b>Table Functions (Row)</b>			
TBADD		Table row add	ISREMC1
TBBOTTOM		Table row pointer to bottom	ISREMCRA
TBDELETE		Table delete	ISREMC2
TBEXIST		Table exist	ISREMC6
TBGET		Table get	ISREMC3
TBMOD		Table modify	ISREMC5
TBPUT		Table put	ISREMC4
TBSARG		Table search parameter	ISREMC7
TBSCAN		Table scan	ISREMC8

Model Name		Description	Member Name
TBSKIP		Table skip	ISREMCRB
TBSORT		Table sort	ISREMCRD
TBTOP		Table top	ISREMCRC9
TBVCLEAR		Table variable clear	ISREMCRC
<b>Variables</b>			
VERASE		Variable erase	ISREMCV8
VGET		Variable get	ISREMCV1
VPUT		Variable put	ISREMCV2
<b>Workstation</b>			
FILESTAT		Statistics for a file	ISREMCX1
FILEXFER		Upload or download a file	ISREMCX2
WSCON		Workstation connection	ISREMCX3
WSDISCON		Workstation disconnect	ISREMCX4

---

## DM and PDF services in COBOL programs

Model Name		Description	Member Name
<b>Display</b>			
ADDPPOP		Display pop-up window	ISREMBD5
DISPLAY		Display option	ISREMBD1
PQUERY		Get panel information	ISREMBD4
REMPPOP		Remove pop-up window	ISREMBD6
SETMSG		Set message display	ISREMBD3
TBDISPL		Table display information	ISREMBD2
<b>File Tailoring</b>			
FTCLOSE		End file tailoring	ISREMBF3
FTERASE		File tailor erase	ISREMBF4
FTINCL		File tailor include skeleton	ISREMBF2
FTOPEN		File tailor open	ISREMBF1
<b>Graphics</b>			
GRERROR		Graphics error block service	ISREMB3
GRINIT		Graphics initialization	ISREMB1
GRTERM		Graphics completion service	ISREMB2
<b>Library Access</b>			
DIRLIST		Displays z/OS UNIX directory list	ISREMBLD
LMCLOSE		Close a data set	ISREMBL1
LMCOMP		Compress a data set	ISREMBLS
LMCOPY		Copy a data set	ISREMBLQ
LMDDISP		Data set list	ISREMBLZ
LMDFREE		Release a data set list	ISREMBLW
LMDINIT		Establish a data set ID	ISREMBLU

Model Name		Description	Member Name
LMDLIST		Obtain a list of data sets	ISREMBLV
LMERASE		Erase a data set or library	ISREMBL2
LMFREE		Release a data set	ISREMBL3
LMGET		Read a record	ISREMBL4
LMINIT		Establish a data ID	ISREMBL5
LMMADD		Add a member	ISREMBL6
LMMDEL		Delete a member	ISREMBL7
LMMDISP		Display member list	ISREMBLO
LMMFIND		Find a member	ISREMBL8
LMMLIST		Create a member list	ISREMBL9
LMMOVE		Move a data set or member	ISREMBLP
LMMREN		Rename a member	ISREMBLA
LMMREP		Replace a member	ISREMBLB
LMMSTATS		Set member statistics	ISREMBLR
LMOPEN		Open a data set	ISREMBLC
LMPRINT		Write member to list data set	ISREMBLT
LMPUT		Write a record	ISREMBLE
LMQUERY		Provide data set information	ISREMBLF
LMRENAME		Rename a library	ISREMBLG
MEMLIST		Displays Option 3.1 member list	ISREMBLH
<b>Miscellaneous</b>			
BRIF		Browse interface service	ISREMCB4
BROWSE	MVS	Browse service (MVS)	ISREMCB1
BROWSE	VM	Browse service (VM)	ISREMCB2
BROWSE	VSE	Browse service (VSE)	ISREMCB3
CONTROL		Control service	ISREMBM2
DSINFO		Returns data set information	ISREMBME
EDIF		Edit interface service	ISREMCCE4
EDIREC		Edit recovery for EDIF	ISREMBM9
EDIT	MVS	Edit service (MVS)	ISREMCCE1
EDIT	VM	Edit service (VM)	ISREMCCE2
EDIT	VSE	Edit service (VSE)	ISREMCCE3
EDREC		Edit recovery services	ISREMBM7
GETMSG		Get message service	ISREMBM6
LIBDEF		LIBDEF service	ISREMBM8
LIST		Write to list data set	ISREMBMA
LOG		Write message or log data set	ISREMBM5
QBASELIB		Query base library information	ISREMBMC
QLIBDEF		Query LIBDEF library information	ISREMBMD
SELECT		Select service	ISREMBM1
VIIF		View Interface service	ISREMCCE6

Model Name		Description	Member Name
<b>Table Functions (General)</b>			
TBCLOSE		Table close	ISREMBG5
TBCREATE		Table create	ISREMBG1
TBEND		Table end	ISREMBG6
TBERASE		Table erase	ISREMBG7
TBOPEN		Table open	ISREMBG2
TBQUERY		Table query	ISREMBG3
TBSAVE		Table save	ISREMBG4
TBSTATS		Table statistics	ISREMBG8
<b>Table Functions (Row)</b>			
TBADD		Table row add	ISREMBR1
TBBOTTOM		Table row pointer to bottom	ISREMBRA
TBDELETE		Table delete	ISREMBR2
TBEXIST		Table exist	ISREMBR6
TBGET		Table get	ISREMBR3
TBMOD		Table modify	ISREMBR5
TBPUT		Table put	ISREMBR4
TBSARG		Table search parameter	ISREMBR7
TBSCAN		Table scan	ISREMBR8
TBSKIP		Table skip	ISREMBRB
TBSORT		Table sort	ISREMBRD
TBTOP		Table top	ISREMBR9
TBVCLEAR		Table variable clear	ISREMBRC
<b>Variables</b>			
VCOPY		Copy variable	ISREMBV5
VDEFINE		Variable define	ISREMBV3
VDELETE		Variable delete	ISREMBV4
VERASE		Variable erase	ISREMBV8
VGET		Variable get	ISREMBV1
VMASK		Variable mask	ISREMBV9
VPUT		Variable put	ISREMBV2
VREPLACE		Variable replace	ISREMBV6
VRESET		Variable reset	ISREMBV7
<b>Working Storage</b>			
WORKSTOR		Working storage definition	ISREMBW1
<b>Workstation</b>			
FILESTAT		Statistics for a file	ISREMBX1
FILEXFER		Upload or download a file	ISREMBX2
WSCON		Workstation connection	ISREMBX3
WSDISCON		Workstation disconnect	ISREMBX4

## DM and PDF services in EXEC commands

Model Name		Description	Member Name
<b>Display</b>			
ADDDPOP		Display pop-up window	ISREMED5
DISPLAY		Display option	ISREMED1
PQUERY		Get panel information	ISREMED4
REMPPOP		Remove pop-up window	ISREMED6
SETMSG		Set message display	ISREMED3
TBDISPL		Table display information	ISREMED2
<b>File Tailoring</b>			
FTCLOSE		End file tailoring	ISREMEF3
FTERASE		File tailor erase	ISREMEF4
FTINCL		File tailor include skeleton	ISREMEF2
FTOPEN		File tailor open	ISREMEF1
<b>Library Access</b>			
LMCLOSE		Close a data set	ISREMEL1
LMERASE		Erase a data set or library	ISREMEL2
LMFREE		Release a data set	ISREMEL3
LMGET		Read a record	ISREMEL4
LMINIT		Establish a data ID	ISREMEL5
LMMADD		Add a member	ISREMEL6
LMMDEL		Delete a member	ISREMEL7
LMMFIND		Find a member	ISREMEL8
LMMLIST		Create a member list	ISREMEL9
LMMREN		Rename a member	ISREMELA
LMMREP		Replace a member	ISREMELB
LMOPEN		Open a data set	ISREMELC
LMPUT		Write a record	ISREMELE
LMQUERY		Provide data set information	ISREMELF
LMRENAME		Rename a library	ISREMELG
LMSPEC		Specify a new ISPF library	ISREMELH
LMUNSPEC		Unspecify an ISPF library	ISREMELI
<b>Miscellaneous</b>			
BROWSE		Browse service	ISREMEM3
CONTROL		Control service	ISREMEM2
EDIT		Edit service	ISREMEM4
EDREC		Edit recovery services	ISREMEM7
GETMSG		Get message service	ISREMEM6
LIBDEF		LIBDEF service	ISREMEM8
LIST		Write list data set	ISREMEMA
LOG		Write message or log data set	ISREMEM5

Model Name		Description	Member Name
QBASLIB		Query base library information	ISREMEMC
QLIBDEF		Query LIBDEF library information	ISREMEMD
SELECT		Select service	ISREMEM1
<b>Table Functions (General)</b>			
TBCLOSE		Table close	ISREMEG5
TBCREATE		Table create	ISREMEG1
TBEND		Table end	ISREMEG6
TBERASE		Table erase	ISREMEG7
TBOPEN		Table open	ISREMEG2
TBQUERY		Table query	ISREMEG3
TBSAVE		Table save	ISREMEG4
TBSTATS		Table statistics	ISREMEG8
<b>Table Functions (Row)</b>			
TBADD		Table row add	ISREMER1
TBBOTTOM		Table row pointer to bottom	ISREMERA
TBDELETE		Table delete	ISREMER2
TBEXIST		Table exist	ISREMER6
TBGET		Table get	ISREMER3
TBMOD		Table modify	ISREMER5
TBPUT		Table put	ISREMER4
TBSARG		Table search parameter	ISREMER7
TBSCAN		Table scan	ISREMER8
TBSKIP		Table skip	ISREMERB
TBSORT		Table sort	ISREMERD
TBTOP		Table top	ISREMER9
TBVCLEAR		Table variable clear	ISREMERC
<b>Variables</b>			
VGET		Variable get	ISREMEV1
VPUT		Variable put	ISREMEV2
VERASE		Variable erase	ISREMEV8

---

## DM and PDF services in FORTRAN programs

Model Name		Description	Member Name
<b>Display</b>			
ADDDPOP		Display pop-up window	ISREMGD5
DISPLAY		Display option	ISREMGD1
PQUERY		Get panel information	ISREMGD4
REMPPOP		Remove pop-up window	ISREMGD6
SETMSG		Set message display	ISREMGD3
TBDISPL		Table display information	ISREMGD2

Model Name		Description	Member Name
<b>File Tailoring</b>			
FTCLOSE		End file tailoring	ISREMF3
FTERASE		File tailor erase	ISREMF4
FTINCL		File tailor include skeleton	ISREMF2
FTOPEN		File tailor open	ISREMF1
<b>Graphics</b>			
GRERROR		Graphics error block service	ISREMF3
GRINIT		Graphics initialization	ISREMF1
GRTERM		Graphics completion service	ISREMF2
<b>Library Access</b>			
DIRLIST		Displays z/OS UNIX directory list	ISREMLD
LMCLOSE		Close a data set	ISREML1
LMCOMP		Compress a data set	ISREML5
LMCOPY		Copy a data set	ISREMLQ
LMDDISP		Data set list	ISREMLZ
LMDFREE		Release a data set list	ISREMLW
LMDINIT		Establish a data set ID	ISREMLU
LMDLIST		Obtain a list of data sets	ISREMLV
LMERASE		Erase a data set or library	ISREML2
LMFREE		Release a data set	ISREML3
LMGET		Read a record	ISREML4
LMINIT		Establish a data ID	ISREML5
LMMADD		Add a member	ISREML6
LMMDEL		Delete a member	ISREML7
LMMDISP		Display member list	ISREML0
LMMFIND		Find a member	ISREML8
LMMLIST		Create a member list	ISREML9
LMMOVE		Move a data set or member	ISREMLP
LMMREN		Rename a member	ISREMLA
LMMREP		Replace a member	ISREMLB
LMMSTATS		Set member statistics	ISREMLR
LMOPEN		Open a data set	ISREMLC
LMPRINT		Write member to list data set	ISREMLT
LMPUT		Write a record	ISREMLE
LMQUERY		Provide data set information	ISREMLF
LMRENAME		Rename a library	ISREMLG
MEMLIST		Displays Option 3.1 member list	ISREMLH
<b>Miscellaneous</b>			
BRIF		Browse interface service	ISREMB4
BROWSE	MVS	Browse service (MVS)	ISREMB1
BROWSE	VM	Browse service (VM)	ISREMB2

Model Name		Description	Member Name
BROWSE	VSE	Browse service (VSE)	ISREMF3
CONTROL		Control service	ISREMF2
DSINFO		Returns data set information	ISREMFME
EDIF		Edit interface service	ISREMF4
EDIREC		Edit recovery for EDIF	ISREMF9
EDIT	MVS	Edit service (MVS)	ISREMF1
EDIT	VM	Edit service (VM)	ISREMF2
EDIT	VSE	Edit service (VSE)	ISREMF3
EDREC		Edit recovery services	ISREMF7
GETMSG		Get message service	ISREMF6
LIBDEF		LIBDEF service	ISREMF8
LIST		Write to list data set	ISREMFMA
LOG		Write message or log data set	ISREMF5
QBASELIB		Query base library information	ISREMFMC
QLIBDEF		Query LIBDEF library information	ISREMFMD
SELECT		Select service	ISREMF1
VIIF		View Interface service	ISREMF6

#### Table Functions (General)

TBCLOSE		Table close	ISREMFZ5
TBCREATE		Table create	ISREMFZ1
TBEND		Table end	ISREMFZ6
TBERASE		Table erase	ISREMFZ7
TBOPEN		Table open	ISREMFZ2
TBQUERY		Table query	ISREMFZ3
TBSAVE		Table save	ISREMFZ4
TBSTATS		Table statistics	ISREMFZ8

#### Table Functions (Row)

TBADD		Table row add	ISREMF1
TBBOTTOM		Table row pointer to bottom	ISREMFRA
TBDELETE		Table delete	ISREMF2
TBEXIST		Table exist	ISREMF6
TBGET		Table get	ISREMF3
TBMOD		Table modify	ISREMF5
TBPUT		Table put	ISREMF4
TBSARG		Table search parameter	ISREMF7
TBSCAN		Table scan	ISREMF8
TBSKIP		Table skip	ISREMF8B
TBSORT		Table sort	ISREMF8D
TBTOP		Table top	ISREMF9
TBVCLEAR		Table variable clear	ISREMF8C

#### Variables

Model Name		Description	Member Name
VCOPY		Copy variable	ISREMFV5
VDEFINE		Variable define	ISREMFV3
VDELETE		Variable delete	ISREMFV4
VERASE		Variable erase	ISREMFV8
VGET		Variable get	ISREMFV1
VMASK		Variable mask	ISREMBV9
VPUT		Variable put	ISREMFV2
VREPLACE		Variable replace	ISREMFV6
VRESET		Variable reset	ISREMFV7
<b>Workstation</b>			
FILESTAT		Statistics for a file	ISREMFV1
FILEXFER		Upload or download a file	ISREMFV2
WSCON		Workstation connection	ISREMFV3
WSDISCON		Workstation disconnect	ISREMFV4

## Message format

Model Name		Description	Member Name
MSG		Message member selection	ISREMSG

## Panel formats and statements

Model Name	Qualifier	Description	Member Name
<b>Panel Formats</b>			
ACTION		'Panel with action bar'	ISREMMF6
ENTRY		'Data entry panel'	ISREMMF1
HELPSCR		'Help panel with scrollable area'	ISREMMF8
MULTIPLE	SELECT2	'Double-column selection panel'	ISREMSE1
MULTIPLE	ENTRY2	'Double-column entry panel'	ISREMSE2
SCROLL		'Panel with scrollable area'	ISREMMF7
SELECTION		'Choice panel'	ISREMMF3
SELECTION	CUA	'Choice panel'	ISREMMF9
TBDISPL		'Table display table'	ISREMMF4
TUTORIAL		'Help panel'	ISREMMF5
<b>Panel Statements</b>			
ABC		'Action bar'	ISREMMSE
AREA		'AREA section header'	ISREMMPI
AREA	DYNAMIC	'Dynamic area attribute'	ISREMMMA1
AREA	GRAPHIC	'Graphic area attribute'	ISREMMMA2
AREA	SCRL	'Scrollable area attribute'	ISREMMMA3
ASSIGN	SIMPLE	'Simple assignment statement'	ISREMAS1

Model Name	Qualifier	Description	Member Name
ASSIGN	TRANS	'Trans assignment statement'	ISREMAS2
ASSIGN	TRUNC	'Trunc assignment statement'	ISREMAS3
ASSIGN	TRANSTRU	'Nested translate truncate statement'	ISREMAS4
ASSIGN	PFKEY	'Function key built-in function'	ISREMAS5
ASSIGN	LVLIN	'Last visible line built-in function'	ISREMAS6
ATTR		'Attribute section header'	ISREMMS2
ATTRIB		'New attribute character definition'	ISREMMS3
ATTRIBA		'New attrib char definition for area'	ISREMMSB
BODY		'Body section header'	ISREMMS4
CCSID		'CCSID section header'	ISREMMPA
CONTROL	CURSOR	'Control first cursor placement'	ISREMCN1
CONTROL	HELP	'Establish a tutorial panel'	ISREMCN2
CONTROL	MSG	'Identify message to be displayed'	ISREMCN3
CONTROL	RESP	'Show user response to panel'	ISREMCN4
CONTROL	TRAIL	'Contain remainder from TRUNC function'	ISREMCN5
CONTROL	ALARM	'Shows the alarm is to be sounded'	ISREMCN6
CONTROL	ATTR	'Override field attr by field name'	ISREMCN7
CONTROL	ATTRCHAR	'Override field attr by character'	ISREMCN8
CONTROL	AUTOSEL	'Control table display row selection'	ISREMCN9
CONTROL	CSRPOS	'Shows position of cursor in field'	ISREMCNA
CONTROL	CSRROW	'Row where cursor is positioned'	ISREMCNB
CONTROL	PFKEY	'Function key pressed by user'	ISREMCND
CONTROL	ZVARS	'Define names of placeholder fields'	ISREMCNC
CUAATTR		'CUA attributes'	ISREMMSJ
END		'END section header'	ISREMMPP
GROUPBOX		'TYPE(GRPBOX) attribute'	ISREMMMA4
HELP		'HELP section'	ISREMMPM
IF		'IF statement'	ISREMMS6
INIT		'INIT section header'	ISREMPJ
KEYLIST		'Keylist specification'	ISREMMSF
LIST		'LIST section'	ISREMMPO
MODEL		'Model section header'	ISREMMS7
PANEXIT		'Panel language exit'	ISREMMSD
PDC		'Action bar pull-down'	ISREMMSH
PNTS		'Point-and-shoot section'	ISREMPN
PROC		'PROC section header'	ISREMMPL
REFRESH		'Retrieve variables before redisplay'	ISREMMSA
REINIT		'REINIT section header'	ISREMPK
SC	ATTR	'TYPE(SC) attribute'	ISREMMMA5
VEDIT		'Validate available'	ISREMMSI

Model Name	Qualifier	Description	Member Name
VERIFY	ALPHA	'Alphabetic or special characters'	ISREMVE1
VERIFY	ALPHAB	'Alphabetic characters'	ISREMVEE
VERIFY	BIT	'Binary characters'	ISREMVE2
VERIFY	DSNAME	'TSO data set name'	ISREMVE3
VERIFY	DSNAMEF	'TSO data set name with filters'	ISREMVEL
VERIFY	DSNAMEFM	'TSO data set name with filter member name only'	ISREMVEO
VERIFY	DSNAMEPQ	'TSO data set name' (adds missing end parenthesis and quote)	ISREMVEx
VERIFY	DSNAMEQ	'TSO data set name' (adds missing end quote)	ISREMVEJ
VERIFY	ENUM	'Extended numeric'	ISREMOVED
VERIFY	FILEID	'CMS file ID'	ISREMVE4
VERIFY	IPADDR4	IP Version 4 address	ISREMVEV
VERIFY	HEX	'Hexadecimal characters'	ISREMVE5
VERIFY	IDATE	'International date'	ISREMVEP
VERIFY	INCLUDE	'Specify list of types'	ISREMVEF
VERIFY	ITIME	'International time'	ISREMVET
VERIFY	JDATE	'Julian date'	ISREMVER
VERIFY	JSTD	'Julian standard date'	ISREMVES
VERIFY	LEN	'Length of data stored in variable'	ISREMVEC
VERIFY	LIST	'List of valid values'	ISREMVE6
VERIFY	LISTV	'Specify list of values'	ISREMVEG
VERIFY	LISTVX	'Specify list of excluded values'	ISREMVEI
VERIFY	LISTX	'Specify list of excluded values'	ISREMVEH
VERIFY	NAME	'Data set member name'	ISREMVE7
VERIFY	NAMEF	'Data set member name with filters'	ISREMVEM
VERIFY	NONBLANK	'Verify nonblank field'	ISREMVE8
VERIFY	NUM	'Numeric characters'	ISREMVE9
VERIFY	PICT	'Mixed characters matching picture'	ISREMVEA
VERIFY	PICTCN	'Constants and mixed characters matching picture'	ISREMVEK
VERIFY	RANGE	'Numeric value within specified limits'	ISREMVEB
VERIFY	STDDATE	'Standard date'	ISREMVEQ
VERIFY	STDTIME	'Standard time'	ISREMVEU
VGET		'Variable get statement'	ISREMMSC
VPUT		'Variable put statement'	ISREMMS9

## DM and PDF services in PL/I programs

Model Name		Description	Member Name
<b>Display</b>			
ADDDPOP		'Display pop-up window'	ISREMPD5
DISPLAY		'Display option'	ISREMPD1
PQUERY		'Get panel information'	ISREMPD4
REMPPOP		'Remove pop-up window'	ISREMPD6
SETMSG		'Set message display'	ISREMPD3
TBDISPL		'Table display information'	ISREMPD2
<b>File Tailoring</b>			
FTCLOSE		'End file tailoring'	ISREMPF3
FTERASE		'File tailor erase'	ISREMPF4
FTINCL		'File tailor include skeleton'	ISREMPF2
FTOPEN		'File tailor open'	ISREMPF1
<b>Graphics</b>			
GRERROR		'Graphics error block service'	ISREMP3
GRINIT		'Graphics initialization'	ISREMP31
GRTERM		'Graphics completion service'	ISREMP32
<b>Library Access</b>			
DIRLIST		Displays z/OS UNIX directory list	ISREMPLD
LMCLOSE		'Close a data set'	ISREMP11
LMCOMP		'Compress a data set'	ISREMP15
LMCOPY		'Copy a data set'	ISREMP14
LMDDISP		Data set list	ISREMP13
LMDFREE		Release a data set list	ISREMP12
LMDINIT		Establish a data set ID	ISREMP10
LMDLIST		Obtain a list of data sets	ISREMP16
LMERASE		'Erase a data set or library'	ISREMP17
LMFREE		'Release a data set'	ISREMP18
LMGET		'Read a record'	ISREMP19
LMINIT		'Establish a data ID'	ISREMP20
LMMADD		'Add a member'	ISREMP21
LMMDEL		'Delete a member'	ISREMP22
LMMDISP		'Display member list'	ISREMP23
LMMFIND		'Find a member'	ISREMP24
LMMLIST		'Create a member list'	ISREMP25
LMMOVE		'Move a data set or member'	ISREMP26
LMMREN		'Rename a member'	ISREMP27
LMMREP		'Replace a member'	ISREMP28
LMMSTATS		'Set member statistics'	ISREMP29
LMOPEN		'Open a data set'	ISREMP30

Model Name		Description	Member Name
LMPRINT		'Write member to list data set'	ISREMPLT
LMPUT		'Write a record'	ISREMPLE
LMQUERY		'Provide data set information'	ISREMPLF
LMRENAME		'Rename a library'	ISREMPLG
MEMLIST		Displays Option 3.1 member list	ISREMP LH
<b>Miscellaneous</b>			
BRIF		'Browse interface service'	ISREMPB4
BROWSE	MVS	'Browse service (MVS)'	ISREMPB1
BROWSE	VM	'Browse service (VM)'	ISREMPB2
BROWSE	VSE	'Browse service (VSE)'	ISREMPB3
CONTROL		'Control service'	ISREMPM2
DSINFO		Returns data set information	ISREMPME
EDIF		'Edit interface service'	ISREMPE4
EDIREC		'EDIF recovery service'	ISREMPM9
EDIT	MVS	'Edit service (MVS)'	ISREMPE1
EDIT	VM	'Edit service (VM)'	ISREMPE2
EDIT	VSE	'Edit service (VSE)'	ISREMPE3
EDREC		'Edit recovery services'	ISREMPM7
GETMSG		'Get message service'	ISREMPM6
LIBDEF		'LIBDEF service'	ISREMPM8
LIST		'Write to list data set'	ISREMPMA
LOG		'Write message or log data set'	ISREMPM5
QBASELIB		Query base library information	ISREMPMC
QLIBDEF		Query LIBDEF library information	ISREMPMD
SELECT		'SELECT service'	ISREMPM1
VIIF		'View Interface service'	ISREMPE6
<b>Table Functions (General)</b>			
TBCLOSE		'Table close'	ISREMPZ5
TBCREATE		'Table create'	ISREMPZ1
TBEND		'Table end'	ISREMPZ6
TBERASE		'Table erase'	ISREMPZ7
TBOPEN		'Table open'	ISREMPZ2
TBQUERY		'Table query'	ISREMPZ3
TBSAVE		'Table save'	ISREMPZ4
TBSTATS		'Table statistics'	ISREMPZ8
<b>Table Functions (Row)</b>			
TBADD		'Table row add'	ISREMPR1
TBBOTTOM		'Table row pointer to bottom'	ISREMPRA
TBDELETE		'Table delete'	ISREMPR2
TBEXIST		'Table exist'	ISREMPR6
TBGET		'Table get'	ISREMPR3

Model Name		Description	Member Name
TBMOD		'Table modify'	ISREMPR5
TBPUT		'Table put'	ISREMPR4
TBSARG		'Table search parameter'	ISREMPR7
TBSCAN		'Table scan'	ISREMPR8
TBSKIP		'Table skip'	ISREMPRB
TBSORT		'Table sort'	ISREMPRD
TBTOP		'Table top'	ISREMPR9
TBVCLEAR		'Table variable clear'	ISREMPRC
<b>Variables</b>			
VCOPY		'Copy variable'	ISREMPV5
VDEFINE		'Variable define'	ISREMPV3
VDELETE		'Variable delete'	ISREMPV4
VERASE		'Variable erase'	ISREMPV8
VGET		'Variable get'	ISREMPV1
VMASK		'Variable mask'	ISREMPV9
VPUT		'Variable put'	ISREMPV2
VREPLACE		'Variable replace'	ISREMPV6
VRESET		'Variable reset'	ISREMPV7
<b>Workstation</b>			
FILESTAT		Statistics for a file	ISREMPX1
FILEXFER		Upload or download a file	ISREMPX2
WSCON		Workstation connection	ISREMPX3
WSDISCON		Workstation disconnect	ISREMPX4

---

## File tailoring control statements

Model Name		Description	Member Name
BLANK		Create blank lines	ISREMSK8
CM		Define comment statement	ISREMSK1
DEFAULT		Change control character defaults	ISREMSK2
DOT		Define DO group for table row	ISREMSK3
IM		Imbed specified data set skeleton	ISREMSK4
SEL		Conditional processing definition	ISREMSK5
SET		Set dialog variable value	ISREMSK6
TB		Set tab stop position	ISREMSK7

---

## DM and PDF services in Pascal programs

Model Name		Description	Member Name
<b>Display</b>			
ADDFPOP		'Display pop-up window'	ISREMQD5

Model Name		Description	Member Name
DISPLAY		'Display option'	ISREMQD1
PQUERY		'Get panel information'	ISREMQD4
REMPPOP		'Remove pop-up window'	ISREMQD6
SETMSG		'Set message display'	ISREMQD3
TBDISPL		'Table display information'	ISREMQD2
<b>File Tailoring</b>			
FTCLOSE		'End file tailoring'	ISREMQF3
FTERASE		'File tailor erase'	ISREMQF4
FTINCL		'File tailor include skeleton'	ISREMQF2
FTOPEN		'File tailor open'	ISREMQF1
<b>Graphics</b>			
GRERROR		'Graphics error block service'	ISREMQS3
GRINIT		'Graphics initialization'	ISREMQS1
GRTERM		'Graphics completion service'	ISREMQS2
<b>Library Access</b>			
DIRLIST		Displays z/OS UNIX directory list	ISREMQLD
LMCLOSE		'Close a data set'	ISREMQL1
LMCOMP		'Compress a data set'	ISREMQLS
LMCOPY		'Copy a data set'	ISREMQLQ
LMDDISP		Data set list	ISREMQLZ
LMDFREE		Release a data set list	ISREMQLW
LMDINIT		Establish a data set ID	ISREMQLU
LMDLIST		Obtain a list of data sets	ISREMQLV
LMERASE		'Erase a data set or library'	ISREMQL2
LMFREE		'Release a data set'	ISREMQL3
LMGET		'Read a record'	ISREMQL4
LMINIT		'Establish a data ID'	ISREMQL5
LMMADD		'Add a member'	ISREMQL6
LMMDEL		'Delete a member'	ISREMQL7
LMMDISP		'Display member list'	ISREMQLO
LMMFIND		'Find a member'	ISREMQL8
LMMLIST		'Create a member list'	ISREMQL9
LMMOVE		'Move a data set or member'	ISREMQLP
LMMREN		'Rename a member'	ISREMQLA
LMMREP		'Replace a member'	ISREMQLB
LMMSTATS		'Set member statistics'	ISREMQLR
LMOPEN		'Open a data set'	ISREMQLC
LMPRINT		'Write member to list data set'	ISREMQLT
LMPUT		'Write a record'	ISREMQLE
LMQUERY		'Provide data set information'	ISREMQLF
LMRENAME		'Rename a library'	ISREMQLG

Model Name		Description	Member Name
MEMLIST		'Displays Option 3.1 member list'	ISREMLH
<b>Miscellaneous</b>			
BRIF		'Browse interface service'	ISREMQB4
BROWSE	MVS	'Browse service (MVS)'	ISREMQB1
BROWSE	VM	'Browse service (VM)'	ISREMQB2
BROWSE	VSE	'Browse service (VSE)'	ISREMQB3
CONTROL		'Control service'	ISREMQM2
DSINFO		'Returns data set information'	ISREMQME
EDIF		'Edit interface service'	ISREMQE4
EDIREC		'Edit recovery for EDIF'	ISREMQM9
EDIT	MVS	'Edit service (MVS)'	ISREMQE1
EDIT	VM	'Edit service (VM)'	ISREMQE2
EDIT	VSE	'Edit service (VSE)'	ISREMQE3
EDREC		'Edit recovery services'	ISREMQM7
GETMSG		'Get message service'	ISREMQM6
LIBDEF		'LIBDEF service'	ISREMQM8
LIST		'Write to list data set'	ISREMQMA
LOG		'Write message or log data set'	ISREMQM5
QBASELIB		Query base library information	ISREMQMC
QLIBDEF		Query LIBDEF library information	ISREMQMD
SELECT		'Select service'	ISREMQM1
VIIF		'View Interface service'	ISREMQE6
<b>Pascal Definitions</b>			
PASDEFS		'Pascal Definitions'	ISREMQPD
<b>Table Functions (General)</b>			
TBCLOSE		'Table close'	ISREMQZ5
TBCREATE		'Table create'	ISREMQZ1
TBEND		'Table end'	ISREMQZ6
TBERASE		'Table erase'	ISREMQZ7
TBOPEN		'Table open'	ISREMQZ2
TBQUERY		'Table query'	ISREMQZ3
TBSAVE		'Table save'	ISREMQZ4
TBSTATS		'Table statistics'	ISREMQZ8
<b>Table Functions (Row)</b>			
TBADD		'Table row add'	ISREMQR1
TBBOTTOM		'Table row pointer to bottom'	ISREMQRA
TBDELETE		'Table delete'	ISREMQR2
TBEXIST		'Table exist'	ISREMQR6
TBGET		'Table get'	ISREMQR3
TBMOD		'Table modify'	ISREMQR5
TBPUT		'Table put'	ISREMQR4

Model Name		Description	Member Name
TBSARG		'Table search parameter'	ISREMQR7
TBSCAN		'Table scan'	ISREMQR8
TBSKIP		'Table skip'	ISREMQRB
TBSORT		'Table sort'	ISREMQRD
TBTOP		'Table top'	ISREMQR9
TBVCLEAR		'Table variable clear'	ISREMQRC
<b>Variables</b>			
VCOPY		'Copy variable'	ISREMQV5
VDEFINE		'Variable define'	ISREMQV3
VDELETE		'Variable delete'	ISREMQV4
VERASE		'Variable erase'	ISREMQV8
VGET		'Variable get'	ISREMQV1
VMASK		'Variable mask'	ISREMQV9
VPUT		'Variable put'	ISREMQV2
VREPLACE		'Variable replace'	ISREMQV6
VRESET		'Variable reset'	ISREMQV7
<b>Workstation</b>			
FILESTAT		Statistics for a file	ISREMQX1
FILEXFER		Upload or download a file	ISREMQX2
WSCON		Workstation connection	ISREMQX3
WSDISCON		Workstation disconnect	ISREMQX4

## DM and PDF services in TSO/REXX commands

Model Name		Description	Member Name
<b>Display</b>			
ADDDPOP		'Display pop-up window'	ISREMRD5
DISPLAY		'Display option'	ISREMRD1
PQUERY		'Get panel information'	ISREMRD4
REMPPOP		'Remove pop-up window'	ISREMRD6
SETMSG		'Set message display'	ISREMRD3
TBDISPL		'Table display information'	ISREMRD2
<b>File Tailoring</b>			
FTCLOSE		'End file tailoring'	ISREMRF3
FTERASE		'File tailor erase'	ISREMRF4
FTINCL		'File tailor include skeleton'	ISREMRF2
FTOPEN		'File tailor open'	ISREMRF1
<b>Library Access</b>			
DIRLIST		Displays z/OS UNIX directory list	ISREMRLD
LMCLOSE		'Close a data set'	ISREMR1
LMCOMP		'Compress a data set'	ISREMR1U

Model Name		Description	Member Name
LMCOPY		'Copy a data set'	ISREMLRQ
LMDDISP		'Data set list'	ISREMLRZ
LMDFREE		Release a data set list	ISREMLRW
LMDINIT		Establish a data set ID	ISREMLRX
LMDLIST		Obtain a list of data sets	ISREMLRV
LMERASE		'Erase a data set or library'	ISREMLR2
LMFREE		'Release a data set'	ISREMLR3
LMGET		'Read a record'	ISREMLR4
LMINIT		'Establish a data ID'	ISREMLR5
LMMADD		'Add a member'	ISREMLR6
LMMDEL		'Delete a member'	ISREMLR7
LMMDISP		'Display member list'	ISREMLR0
LMMFIND		'Find a member'	ISREMLR8
LMMLIST		'Create a member list'	ISREMLR9
LMMOVE		'Move a data set or member'	ISREMLRP
LMMREN		'Rename a member'	ISREMLRA
LMMREP		'Replace a member'	ISREMLRB
LMMSTATS		'Set member statistics'	ISREMLRr
LMOPEN		'Open a data set'	ISREMLRC
LMPRINT		'Write member to list data set'	ISREMLRT
LMPUT		'Write a record'	ISREMLRE
LMQUERY		'Provide data set information'	ISREMLRF
LMRENAME		'Rename a library'	ISREMLRG
MEMLIST		'Displays Option 3.1 member list'	ISREMLRH
<b>Miscellaneous</b>			
BROWSE		'Browse service'	ISREMRM3
CONTROL		'Control service'	ISREMRM2
DSINFO		'Returns data set information'	ISREMRME
EDIT		'Edit service'	ISREMRM4
EDREC		'Edit recovery services'	ISREMRM7
GETMSG		'Get message service'	ISREMRM6
LIBDEF		'LIBDEF service'	ISREMRM8
LIST		'Write to list data set'	ISREMRMA
LOG		'Write message or log data set'	ISREMRM5
QBASELIB		Query base library information	ISREMRMC
QLIBDEF		Query LIBDEF library information	ISREMRMD
SELECT		'SELECT service'	ISREMRM1
<b>Table Functions (General)</b>			
TBCLOSE		'Table close'	ISREMRG5
TBCREATE		'Table create'	ISREMRG1
TBEND		'Table end'	ISREMRG6

Model Name		Description	Member Name
TBERASE		'Table erase'	ISREMRG7
TBOPEN		'Table open'	ISREMRG2
TBQUERY		'Table query'	ISREMRG3
TBSAVE		'Table save'	ISREMRG4
TBSTATS		'Table statistics'	ISREMRG8
<b>Table Functions (Row)</b>			
TBADD		'Table row add'	ISREMR1
TBBOTTOM		'Table row pointer to bottom'	ISREMRRA
TBDELETE		'Table delete'	ISREMR2
TBEXIST		'Table exist'	ISREMR6
TBGET		'Table get'	ISREMR3
TBMOD		'Table modify'	ISREMR5
TBPUT		'Table put'	ISREMR4
TBSARG		'Table search parameter'	ISREMR7
TBSCAN		'Table scan'	ISREMR8
TBSKIP		'Table skip'	ISREMRB
TBSORT		'Table sort'	ISREMRD
TBTOP		'Table top'	ISREMR9
TBVCLEAR		'Table variable clear'	ISREMRRC
<b>Variables</b>			
VERASE		'Variable erase'	ISREMRV8
VGET		'Variable get'	ISREMRV1
VPUT		'Variable put'	ISREMRV2
<b>Workstation</b>			
FILESTAT		Statistics for a file	ISREMRX1
FILEXFER		Upload or download a file	ISREMRX2
WSCON		Workstation connection	ISREMRX3
WSDISCON		Workstation disconnect	ISREMRX4

---

## SCLM architecture definition formats

Model Name		Description	Member Name
<b>Architecture Definition Formats</b>			
'CC'		'Compilation Control'	ISREMHAC
'LEC'		'Linkage Editor Control'	ISREMHAL
'HL'		'High Level Definition'	ISREMHAAH
'GENERIC'		'Special processing control'	ISREMHAG

## SCLM project definition macros and templates

Model Name	Description	Member Name
<b>Macros</b>		
'FLMABEG'	'Define project name'	ISREMGAB
'FLMAEND'	'End project definition'	ISREMGAE
'FLMAGRP'	'Define authcode group'	ISREMGAG
'FLMALLOC'	'Define ddname'	ISREMGAL
'FLMALTC'	'Define alternate control set'	ISREMGAC
'FLMATVER'	'Enable audit tracking and versioning'	ISREMGAV
'FLMCMPLB'	'Name compool library'	ISREMGCM
'FLMCNTRL'	'Specify project controls'	ISREMGCN
'FLMCPYLB'	'Name data set for allocation'	ISREMGCP
'FLMEXLIB'	'Define external library name'	ISREMGEX
'FLMGROUP'	'Define group of libraries'	ISREMGGR
'FLMLANGL'	'Define a language'	ISREMGLA
'FLMSYSLB'	'Name system library'	ISREMGSY
'FLMTRNSL'	'Define language translator'	ISREMGTR
'FLMTYPE'	'Define a library type'	ISREMGTY
<b>Templates</b>		
'PROJDEF'	'Project definition'	ISREMGT1
'LANGUAGE'	'Language definition'	ISREMGT2

## ISPF Dialog Tag Language models

Model Name	Description	Member Name
<b>Panel Formats</b>		
'ACTION BAR'	'Action bar panel'	ISREMDP2
ENTRY	'Data entry panel'	ISREMDP1
SELECTION	'Choice panel'	ISREMDP3
'TABLE DISPLAY'	'Scrollable list'	ISREMDP5
TUTORIAL	'Help/Tutorial panel'	ISREMDP4
<b>Command Table Format</b>		
'COMMAND TABLE'	'Command table application'	ISREMDC1

## DM and PDF services in C/370 programs

Model Name	Description	Member Name
<b>Display</b>		
ADDPPOP	'Display pop-up window'	ISREMDW5
DISPLAY	'Display option'	ISREMDW1

Model Name		Description	Member Name
PQUERY		'Get panel information'	ISREMWD4
REMPop		'Remove pop-up window'	ISREMWD6
SETMSG		'Set message display'	ISREMWD3
TBDISPL		'Table display information'	ISREMWD2
<b>File Tailoring</b>			
FTCLOSE		'End file tailoring'	ISREMWF3
FTERASE		'File tailor erase'	ISREMWF4
FTINCL		'File tailor include skeleton'	ISREMWF2
FTOPEN		'File tailor open'	ISREMWF1
<b>Graphics</b>			
GRERROR		'Graphics error block service'	ISREMWS3
GRINIT		'Graphics initialization'	ISREMWS1
GRTERM		'Graphics completion service'	ISREMWS2
<b>Library Access</b>			
DIRLIST		Displays z/OS UNIX directory list	ISREMWLD
LMCLOSE		'Close a data set'	ISREMWL1
LMCOMP		'Compress a data set'	ISREMWL0
LMCOPY		'Copy a data set'	ISREMWLY
LMDDISP		Data set list	ISREMWLZ
LMDFREE		Release a data set list	ISREMWLW
LMDINIT		Establish a data set ID	ISREMWLU
LMDLIST		Obtain a list of data sets	ISREMWLV
LMERASE		'Erase a data set or library'	ISREMWL2
LMFREE		'Release a data set'	ISREMWL3
LMGET		'Read a record'	ISREMWL4
LMINIT		'Establish a data ID'	ISREMWL5
LMMADD		'Add a member'	ISREMWL6
LMMDEL		'Delete a member'	ISREMWL7
LMMDISP		'Display member list'	ISREMWLL
LMMFIND		'Find a member'	ISREMWL8
LMMLIST		'Create a member list'	ISREMWL9
LMMOVE		'Move a data set or member'	ISREMWLM
LMMREN		'Rename a member'	ISREMWL0
LMMREP		'Replace a member'	ISREMWLB
LMMSTATS		'Set member statistics'	ISREMWLP
LMOPEN		'Open a data set'	ISREMWLC
LMPRINT		'Write member to list data set'	ISREMWLQ
LMPUT		'Write a record'	ISREMWLE
LMQUERY		'Provide data set information'	ISREMWLF
LMRENAME		'Rename a library'	ISREMWLG
MEMLIST		'Displays Option 3.1 member list'	ISREMWLH

Model Name		Description	Member Name
<b>Miscellaneous</b>			
BRIF		'Browse interface service'	ISREMWM9
BROWSE	MVS	'Browse service'	ISREMWM3
CONTROL		'Control service'	ISREMWM2
DSINFO		'Returns data set information'	ISREMWME
EDIF		'Edit interface service'	ISREMWMA
EDIREC		'Edit recovery for EDIF'	ISREMWMB
EDIT	MVS	'Edit service'	ISREMW4
EDREC		'Edit recovery services'	ISREMW7
GETMSG		'Get message service'	ISREMW6
LIBDEF		'LIBDEF service'	ISREWM8
LIST		'Write to list data set'	ISREWMC
LOG		'Write message or log data set'	ISREWM5
QBASELIB		Query base library information	ISREMWQ2
QLIBDEF		Query LIBDEF library information	ISREMWQ1
SELECT		'Select service'	ISREWM1
VIIF		'View Interface service'	ISREWMF
<b>Table Functions (General)</b>			
TBCLOSE		'Table close'	ISREMWG5
TBCREATE		'Table create'	ISREMWG1
TBEND		'Table end'	ISREMWG6
TBERASE		'Table erase'	ISREMWG7
TBOPEN		'Table open'	ISREMWG2
TBQUERY		'Table query'	ISREMWG3
TBSAVE		'Table save'	ISREMWG4
TBSTATS		'Table statistics'	ISREMWG8
<b>Table Functions (Row)</b>			
TBADD		'Table row add'	ISREMR1
TBBOTTOM		'Table row pointer to bottom'	ISREMR8
TBDELETE		'Table delete'	ISREMR2
TBEXIST		'Table exist'	ISREMR6
TBGET		'Table get'	ISREMR3
TBMOD		'Table modify'	ISREMR5
TBPUT		'Table put'	ISREMR4
TBSARG		'Table search parameter'	ISREMR7
TBSCAN		'Table scan'	ISREMR8
TBSKIP		'Table skip'	ISREMRB
TBSORT		'Table sort'	ISREMRD
TBTOP		'Table top'	ISREMR9
TBVCLEAR		'Table variable clear'	ISREMR9C

#### Variables

Model Name		Description	Member Name
VCOPY		'Copy variable'	ISREMWV5
VDEFINE		'Variable define'	ISREMWV3
VDELETE		'Variable delete'	ISREMWV4
VERASE		'Variable erase'	ISREMWV8
VGET		'Variable get'	ISREMWV1
VMASK		'Variable mask'	ISREMWV9
VPUT		'Variable put'	ISREMWV2
VREPLACE		'Variable replace'	ISREMWV6
VRESET		'Variable reset'	ISREMWV7
<b>Workstation</b>			
FILESTAT		'Statistics for a file'	ISREMWX1
FILEXFER		'Upload or download a file'	ISREMWX2
WSCON		'Workstation connection'	ISREMWX3
WSDISCON		'Workstation disconnect'	ISREMWX4

## Dialog Tag Language models

Model Name		Description	Member Name
AB		Action bar	ISREMDAB
AB (example)		Action bar with pull-down choice	ISREMDAG
ABC		Action bar choice	ISREMDA2
ACTION		Action	ISREMDAC
AREA		Area	ISREMDAR
ASSIGNI		Assignment List Item	ISREMDAI
ASSIGNL		Assignment list	ISREMDAL
ATTENTION		Attention	ISREMDAN
ATTR		Attribute	ISREMDAT
BOTINST		Bottom instruction	ISREMDBI
CAUTION		Caution	ISREMDCU
CHDIV		Choice divider	ISREMDCD
AB		Action bar	ISREMDAB
AB		Action bar	ISREMDAB
CHECKI		Validity Check Item	ISREMDCI
CHECKL		Check validity list	ISREMDCL
CHOFLD		Choice data field	ISREMDCF
CHOICE		Selection choice	ISREMDCH
CMD		Command definition	ISREMDCM
CMDACT		Command action	ISREMDCC
CMDAREA		Command area	ISREMDCA
CMDTBL		Command table	ISREMDCT
CMDTBL example		Command table example	ISREMDC1

Model Name		Description	Member Name
COMMENT		Comment	ISREMDCN
COMPOPT		Compiler options	ISREMDCO
COPYR		Copyright	ISREMDCR
DA		Dynamic Area	ISREMDDA
DD		Definition description	ISREMDDD
DDHD		Definition description header	ISREMDDD1
DIVIDER		Area divider	ISREMDDI
DL		Definition list	ISREMDDL
DOCTYPE		Document type	ISREMDTP
DT		Definition term	ISREMDDT
DTACOL		Data column	ISREMDDC
DTAFLD		Data field	ISREMDDF
DTAFLDD		Data field description	ISREMDDX
DTHD		Definition term header	ISREMDD2
ENTITY		Entity	ISREMDEN
FIG		Figure	ISREMDFI
FIGCAP		Figure caption	ISREMDFC
GA		Graphic area	ISREMDGA
GRPHDR		Group header	ISREMDGH
HELP		Help panel	ISREMDHE
HELPDEF		Help default	ISREMDHD
HP		Highlighted phrase	ISREMDHP
H1		Heading 1	ISREMDH1
Hn		Heading (H2, H3, H4)	ISREMDHN
INFO		Information region	ISREMDIN
KEYI		Key item	ISREMDKI
KEYL		Key list	ISREMDKL
LI		List item	ISREMDLI
LINES		Lines	ISREMDLN
LIT		Literal	ISREMDLT
LP		List part	ISREMDLP
LSTCOL		List column	ISREMDLC
LSTFLD		List field	ISREMDLF
LSTGRP		List group	ISREMDLG
LSTVAR		List variable	ISREMDLV
M		Mnemonic	ISREMDMN
MSG		Message	ISREMDMS
MSGMBR		Message member	ISREMDMM
NOTE		Note	ISREMDNO
NOTEL		Note list	ISREMDNL
NT		Note	ISREMDNT

<b>Model Name</b>		<b>Description</b>	<b>Member Name</b>
OL		Ordered list	ISREMDOL
Panel example		Panel example	ISREMDP1
Panel example		Panel example	ISREMDP2
Panel example		Panel example	ISREMDP3
Panel example		Panel example	ISREMDP4
Panel example		Panel example	ISREMDP5
PANDEF		Panel default	ISREMDPB
PANEL		Panel	ISREMDPA
PARML		Parameter list	ISREMDPL
PD		Parameter description	ISREMDPD
PDC		Pull-down choice	ISREMDA3
PDSEP		Pull-down separator	ISREMDA4
PG		Paragraph	ISREMDPG
PNLINST		Panel instruction	ISREMDPI
PS		Point-and-shoot	ISREMDPS
PT		Parameter term	ISREMDPT
Region		Region	ISREMDRE
RP		Reference phrase	ISREMDRP
SF		Selection field	ISREMDSF
SL		Simple list	ISREMDSL
SOURCE		Source	ISREMDSO
T		Truncation	ISREMDTR
TOPINST		Top instruction	ISREMDTI
UL		Unordered list	ISREMDUL
VARCLASS		Variable class	ISREMDVC
VARDCL		Variable declaration	ISREMDVD
VARLIST		Variable list	ISREMDVL
VARSUB		Variable substitution	ISREMDVS
WARNING		Warning	ISREMDWA
XLATI		Translate item	ISREMDXI
XLATL		Translate list	ISREMDXL
XMP		Example	ISREMDXM

---

## Chapter 13. Programming interface macros for customers

This topic lists the macros ISPF and SCLM provide as programming interfaces.

**Attention:** Do not use any ISPF macros other than those listed in this topic as programming interfaces.

*Table 52. DM component general-use programming macros*

---

ISPCCSID	ISPMXDEF
ISPMDAD	ISPMXED
ISPMEPT	ISPMXEND
ISPMTCM	ISPMXLST
ISPMXDD	ISPDSTAT

---

See Chapter 5, “Customizing DM,” on page 79 for further information about DM component macros.

*Table 53. SCLM general-use programming macros*

---

FLMABEG	FLMGROUP
FLMAEND	FLMINCLS
FLMAGRP	FLMLANGL
FLMALLOC	FLMSYSLB
FLMALTC	FLMTRNSL
FLMATVER	FLMTSEXT
FLMCNTRL	FLMTYPE
FLMCPYLB	

---

See the SCLM Macros chapter in *z/OS ISPF Software Configuration and Library Manager Guide and Reference* for more information.



## Chapter 14. ISPF data set descriptions

These figures list the target and distribution libraries (data sets) used by ISPF, and their contents.

Table 54. Target data set descriptions

Library	Description
SISPALIB	APL2 workspace library
SISPCLIB	CLIST library
SISPEXEC	REXX exec library
SISPGDES	Swiss German Language-specific panel source library
SISPGDEU	German Language-specific panel source library
SISPGENP	Uppercase English Language-specific panel source library
SISPGENU	English Language-specific panel source library
SISPGJPN	Japanese Language-specific panel source library
SISPGMLI	Non-language-specific panel source library
SISPGUI	ISPF Client/Server workstation code library
SISPHelp	Help library
SISpload	Linklist load library
SISPLPA	LPA load library
SISPMACS	ISPF Macro library
SISPMDES	Swiss German Language-specific message library
SISPMDEU	German Language-specific message library
SISPMENP	Uppercase English Language-specific message library
SISPMENU	English Language-specific message library
SISPMJPN	Japanese Language-specific message library
SISPPDES	Swiss German Language-specific panel library
SISPPDEU	German Language-specific panel library
SISPPENP	Uppercase English Language-specific panel library
SISPPENU	English Language-specific panel library
SISPPJPN	Japanese Language-specific panel library
SISPSAMP	Sample library
SISPSDES	Swiss German Language-specific skeleton library
SISPSDEU	German Language-specific skeleton library
SISPSENP	Uppercase English Language-specific skeleton library
SISPSENU	English Language-specific skeleton library
SISPSJPN	Japanese Language-specific skeleton library
SISPSLIB	Non-language-specific skeleton library
SISPTDES	Swiss German Language-specific table library
SISPTDEU	German Language-specific table library
SISPTENP	Uppercase English Language-specific table library

Table 54. Target data set descriptions (continued)

Library	Description
SISPTENU	English Language-specific table library
SISPTJPN	Japanese Language-specific table library

Table 55. Distribution data set descriptions

Library	Description
AISPALIB	APL2 workspace library
AISPCLIB	CLIST library
AISPEXEC	REXX exec library
AISPGDES	Swiss German Language-specific panel source library
AISPGDEU	German Language-specific panel source library
AISPGENP	Uppercase English Language-specific panel source library
AISPGENU	English Language-specific panel source library
AISPGJPN	Japanese Language-specific panel source library
AISPGMLI	Non-language-specific panel source library
AISPGUI	ISPF Client/Server workstation code library
AISPHelp	Help library
AISPMACS	ISPF Macro library
AISPMDES	Swiss German Language-specific message library
AISPMDEU	German Language-specific message library
AISPMENP	Uppercase English Language-specific message library
AISPMENU	English Language-specific message library
AISPMJPN	Japanese Language-specific message library
AISPMOD1	Executable modules library
AISPPDES	Swiss German Language-specific panel library
AISPPDEU	German Language-specific panel library
AISPPENP	Uppercase English Language-specific panel library
AISPPENU	English Language-specific panel library
AISPPJPN	Japanese Language-specific panel library
AISPSAMP	Sample library
AISPSDES	Swiss German Language-specific skeleton library
AISPSDEU	German Language-specific skeleton library
AISPSENP	Uppercase English Language-specific skeleton library
AISPSENU	English Language-specific skeleton library
AISPSJPN	Japanese Language-specific skeleton library
AISPSLIB	Non-language-specific skeleton library
AISPTDES	Swiss German Language-specific table library
AISPTDEU	German Language-specific table library
AISPTENP	Uppercase English Language-specific table library
AISPTENU	English Language-specific table library
AISPTJPN	Japanese Language-specific table library

---

## Part 3. Appendixes



---

## Appendix A. ISPF enqueue processing for data integrity

**Note:** The enqueue processing performed by ISPF as described in this topic does not apply (that is, ISPF takes no action) when the data set is being accessed with the VSAM utility.

This topic describes the enqueue processing ISPF performs to maintain the integrity of data sets, PDS members, member queues, workstation files, and the SCLM VSAM database.

---

### Serializing with non-ISPF TSO and BATCH

ISPF relies on MVS allocation to serialize access to resources with concurrent batch or non-ISPF TSO users. ISPF uses dynamic allocation and allocates partitioned data sets with DISP=SHR before any ISPF generated ENQUEUE. To ensure integrity when batch or TSO users are not using ISPF services, if you are updating a data set, you must allocate the data set with DISP=OLD.

**Note:** ISPF's allocation with DISP=SHR causes MVS allocation to issue a shared ENQUEUE on Qname SYSDSN as follows:

```
ENQ SYSDSN,dsname,S,44,SYSTEM
```

Non-ISPF TSO and Batch allocation with DISP=OLD causes MVS allocation to issue an exclusive ENQUEUE on Qname SYSDSN as follows:

```
ENQ SYSDSN,dsname,E,44,SYSTEM
```

ISPF also issues ENQ, DEQ, and RESERVE macro instructions to serialize access to resources among multiple ISPF users. It is possible for the LMMOVE service or option 3.3 Move function to be used such that each of 2 users is holding a RESERVE (therefore an exclusive ENQ is held) that is being waited on by the other user. This occurs when the user 1 is moving from data set A to data set B, while at the same time, user 2 is moving from data set B to data set A. Each user holds a reserve on the output data set for the move and is requesting a reserve on the input for purposes of deleting the moved member. If this situation occurs, cancel one user off TSO, log back on, and reissue the move request.

---

### ISPF data set integrity enqueue

To allow you to update a data set on DASD (EDIT's SAVE, PDF option 3 COPY/MOVE/DELETE/RENAME, TBSAVE, and so on), ISPF issues this macro to protect the entire partitioned data set:

```
RESERVE SPFEDIT,dsname,E,44,SYSTEMS
```

Also, when writing to a data set on a DASD (COPY/MOVE) that has a RECFM of 'U', ISPF serializes with the linkage editor using this macro to protect the entire partitioned data set:

On a shared volume:

```
RESERVE SYSIEWLP,dsname,E,44,SYSTEMS
```

On a volume that is not shared:

```
ENQ SYSIEWLP,dsname,E,44,SYSTEM
```

**Attention:** Do not install SPF and ISPF on the same system. There is a danger of destroying PDSs that are being updated by SPF and ISPF at the same time because SPF uses a different Qname (SPFDSN) than ISPF.

---

## Member name enqueue

To restrict concurrent use of a member of a partitioned data set, while still allowing ISPF users to use different members of the same data set (PDF EDIT, Table Processing, File Tailoring), ISPF issues this ENQ macro for the member:

```
ENQ SPFEDIT,rname,E,52,SYSTEMS
```

where

**rname** the data set name, length of 44, padded with blanks, followed by the member name, length of 8, padded with blanks

---

## z/OS UNIX file Enqueue

To restrict concurrent use of a z/OS UNIX file, ISPF edit issues this ENQ macro for the file:

```
ENQ SPFEDIT,rname,E,12,SYSTEMS (z/OS UNIX in sysplex; OextSysplexActv = '1'b)
```

or

```
ENQ SPFEDIT,rname,E,12,SYSTEM (z/OS UNIX not in sysplex)
```

where

**rname** Is 12 bytes in length comprising these 4-byte binary integers:

- The inode number of the file.
- The device number of the file.
- Sysplex indicator:
  - 1 z/OS UNIX is in a sysplex (OextSysplexActv is set ON).
  - 0 z/OS UNIX is not in a sysplex (OextSysplexActv is set OFF).

This enqueue is compatible with the enqueue issued by the z/OS UNIX OEDIT command.

---

## ISPF GUI TSO line mode message queue enqueue

Running ISPF GUI with TSO line mode support (that is, displaying TSO line mode and tputs/tgets in the ISPF/TSO window) requires a messaging queue for each user (TSO address space). This is used to coordinate the incoming TSO line mode and TPUT/TGET messages so that they will be synchronized correctly. Thus, ISPF issues this enqueue before placing messages on or retrieving messages from this message queue:

```
ENQ SYSZDTSK,rname,E,38,SYSTEMS
```

where

**rname** 'ISPF/TSO\_WINDOW\_SERIALIZATION' followed by a 64 bit unique ID for the TSO address space

**Note:** In addition to issuing this enqueue while running in GUI mode, this enqueue on SYSZDTSK is issued from the ISPF SVC 93 exit ISPSC93X when processing a cross-memory message even if the user is not running in GUI

mode. This is because at the point the enqueue must be issued, the exit is not running in the user's address space and therefore does not know if the user is running in ISPF GUI mode.

---

## Workstation file name enqueue

To restrict the concurrent use of a workstation file through ISPF's distributed editing or FILEXFER service, ISPF issues this ENQ macro for the workstation file:

```
ENQ SPFEDIT,wsname,E,255,SYSTEMS
```

where

**wsname**

The workstation address (IP:*tcpip-address* or LU:*appc-address*), padded with blanks to 68 characters, followed by the workstation file name identifier, padded with blanks to 187 characters. The entire string is padded to 255 characters with blanks.

---

## SCLM VSAM enqueue

To restrict the concurrent use of the SCLM VSAM data base, ISPF issues this ENQ macro for the VSAM data set:

```
ENQ SLMVSAM,vsam-dsn,E,44,SYSTEMS
```

where

**vsam-dsn**

The VSAM data set name, padded to 44 characters with blanks.



---

## Appendix B. Accessibility

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products successfully. The major accessibility features in z/OS® enable users to:

- Use assistive technologies such as screen readers and screen magnifier software
- Operate specific or equivalent features using only the keyboard
- Customize display attributes such as color, contrast, and font size

---

### Using assistive technologies

Assistive technology products, such as screen readers, function with the user interfaces found in z/OS. Consult the assistive technology documentation for specific information when using such products to access z/OS interfaces.

---

### Keyboard navigation of the user interface

Users can access z/OS user interfaces using TSO/E or ISPF. Refer to *z/OS TSO/E Primer*, *z/OS TSO/E User's Guide*, and *z/OS ISPF User's Guide Vol I* for information about accessing TSO/E and ISPF interfaces. These guides describe how to use TSO/E and ISPF, including the use of keyboard shortcuts or function keys (PF keys). Each guide includes the default settings for the PF keys and explains how to modify their functions.

---

### z/OS information

z/OS information is accessible using screen readers with the BookServer/Library Server versions of z/OS books in the Internet library at:

<http://www.ibm.com/systems/z/os/zos/bkserv/>



---

## Notices

This information was developed for products and services offered in the USA.

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  
USA

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation  
Licensing  
2-31 Roppongi 3-chome, Minato-ku  
Tokyo 106, Japan

**The following paragraph does not apply to 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 information 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.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

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 Corporation  
Mail Station P300  
2455 South Road  
Poughkeepsie, NY 12601-5400  
USA

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 information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

#### COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

---

## Programming Interface Information

This publication primarily documents information that is NOT intended to be used as Programming Interfaces of ISPF.

This publication also documents intended Programming Interfaces that allow the customer to write programs to obtain the services of ISPF. This information is identified where it occurs, either by an introductory statement to a chapter or section or by the following marking:

+-----Programming Interface information-----+

+-----End of Programming Interface information-----+

---

## Trademarks

A current list of IBM trademarks is available on the Web at "Copyright and trademark information", <http://www.ibm.com/legal/copytrade.shtml>.

# Index

## A

accessibility 309  
action bars  
    customizing 39  
    extended color in Edit 123  
    removing from ISPF panels 68  
alternate option 7.1 panels 110  
APL2 workspace, loading 105  
application command table 85

## B

BACKUP temporary data sets,  
    preallocating 76  
backup, Edit 123  
base code pages for terminals 97  
batch  
    CLIST 161  
    modifications 161  
    options  
        adding to PDF 161  
        customizing 130  
    processing  
        control program (ISRJB1) 159  
        flow of control 158  
        generated JCL 160  
        panels, CLIST, and skeletons 145  
        required variables 160  
    SCLM 125  
    skeletons 161  
    suboptions, adding to PDF 162  
    using to serialize 305  
block size, edit recovery data set 67  
Browse  
    automatic 142  
    customizing 167  
    panel  
        customizing 168  
        output-only variables 171  
    PDF invocation of 143  
    providing Unicode support 172  
build  
    Configuration Table Load Module 33  
    SMP/E USERMOD 35

## C

CCSID  
    coded character set identifier 95  
    SBCS/DBCS 96  
    Unicode support in Browse 172  
changing DTL source  
    ISPISMMN 91  
    ISPOPTxx 93  
CLIST  
    batch 161  
    batch processing 145  
    foreground 143  
    foreground processing 141  
    ISRFPFR 143

CLIST (*continued*)  
    panel names 138  
    with foreground processing  
        panels 130  
CNTL temporary data sets,  
    preallocating 77  
code page translation tables, creating 95  
coding errors, skeleton, debugging 161  
collating sequence translation table  
    (TTCOL) 89  
command accounting 80  
commands  
    reading syntax diagrams viii  
compress request exit 196  
compressing listings 78  
configuration  
    table  
        example 9  
        utility, ISPF 12  
        utility, using before installation  
            complete 10  
configuration table  
    keyword file 203  
configuration table fields, ISPF 203  
    Data Set Allocation Settings 204  
    Default CUA Color Settings 227  
edit recovery data set 206  
Edit Site-wide Profile  
    Customizations 212  
Edit-related Settings 209  
EDIT/BROWSE/VIEW 235  
ISPF Site-wide Profile  
    Customizations 217  
ISPSPROF General Values 236  
ISPSPROF Workstation Defaults 243  
ISPSPROF Workstation Download  
    Defaults 247  
LMF 206  
Miscellaneous Settings 231  
Move/Copy Settings 207  
Outlist data set specifications 205  
PDF Exits 203  
Removable media interface  
    settings 207  
SuperC Data set specifications 205  
Values Formerly in ISPDFLTS 235  
configuration table, ISPF 9  
CONVACTB exec 44  
Convert Assembler Configuration Table  
    to Keyword File 35  
converting user profile data 95  
Create/Modify Settings and Regenerate  
    Keyword File 14  
customization options  
    ISPF 79  
    PDF 121  
customizing  
    browse and edit 167  
    ISPF/PDF product options 175  
    ISPTCM 106  
    member list panels 172

customizing (*continued*)  
    z/OS UNIX directory list panel 175

## D

daemon name, TCP/IP 106  
data set  
    allocation exit 182  
    allocation settings 204  
    integrity enqueue  
        DASD 305  
        destroying PDSs 305  
        member name enqueue 305  
        updating data sets 305  
    list  
        exit 181  
        line command exit 190  
        utility 190  
    name change exit 197  
    print 124  
    temporary sort data sets 166  
Data Set Allocation Settings 204  
default CUA Color Settings 227  
defaults, ISPF site-wide 84  
dialog development model listings  
    file tailoring control statements 287  
    ISPF Dialog Tag Language panel  
        models 293  
    ISPF services  
        in CLIST commands 273  
        in CLIST programs 293  
        in COBOL programs 275  
        in EXEC commands 278  
        in FORTRAN programs 279  
        in Pascal programs 287  
        in PLI programs 285  
        in TSO/REXX commands 290  
    message format 282  
    panel formats and statements 282  
Dialog Tag Language Models 296  
dialogs, converting LMF to SCLM 129  
disability 309  
disabling high-level qualifiers 74  
DISPLAY service exit 261  
double-byte character set (DBCS) 125  
DSLISL removable media interface  
    settings 207  
dynamic area, Edit and Browse  
    panels 171

## E

edit  
    backup and recovery 123  
    customizing 167  
    defaults 121  
    extended highlighting, disabling 68  
    functions, removing or tailoring 67  
    keyword file configuration table 31

edit (*continued*)

- panel
  - customizing 168
  - output-only variables 171
- profile 121
- recovery data set
  - changing the block size 67
  - ISPF Configuration Table fields 206
  - restrictions on 124
  - related settings 209
  - site-wide profile customizations 212
- terminal output character translation table 267
- error processing for exit routines 119
- execution data sets, allocating 73
- exit macros
  - exit data area definitions 249
  - exit entry definitions 249
  - ISPMDDAD 250
  - ISPMDEPT 250
  - ISPMXDD 250
  - ISPMXDEF 249
  - ISPMXED 249
  - ISPMXEND 250
  - ISPMXLST 249
- exit parameter list 118
- exits 113, 181
- expansion triggers 166
- extended
  - code pages
    - adding translation tables for 98
    - supported by ISPF 95
    - translation tables 99
  - color in Edit 123

**F**

- file tailoring
  - allocating 39
  - control statements, model listings for 287
  - image library 39
- FMID for USERMOD 36
- foreground
  - CLIST 143
  - ISRFPR 143
  - modifications 143
  - options
    - adding new option to PDF 144
    - adding new suboption to PDF 144
    - customizing 130
    - invoking ISRFPR 140
    - unauthorized commands 144
  - processing
    - flow of control 139
    - panel and CLIST names 138
    - required variables 141
- FORTTRAN and COBOL interactive CLIST 143
- fragments, syntax diagrams viii

**G**

- gateway, TSO/ISPF
  - customizing 49
  - description 46
  - HTTP server, customizing 49
  - installing 48
  - other environments, configuration for 50
  - Remote Systems Explorer, customizing 50
  - using 51
- Generalized Markup Language (GML) 39
- generic string
  - character translation table 267
  - master translation table 88, 267, 269
  - special character translation table 267

**H**

- hard copy utility, using DBCS data sets 125
- hexadecimal character translation table 265
- high-level qualifiers, disabling 74
- HILITE 68

**I**

- IEEMB846 module 80
- image library, allocating 39
- initialization, site-wide edit profile 122
- installation exits
  - DISPLAY service 256, 261
  - ISPF session
    - initialization 251
    - termination 251
  - LIBDEF service 257
  - log, list, and temporary data set allocation 263
  - logical screen
    - end 259
    - start 259
  - PDF service
    - end 260
    - start 260
  - RELEASE 258
  - RESERVE 258
  - SELECT service end 254
  - SWAP 261
  - TSO command
    - end 256
    - start 255
- installation-wide exits
  - activity monitoring exits 186, 258
  - calling the exit routine 197
  - common parameter list 118
  - compress request exit 196
  - data set exits
    - allocation 182
    - list 181
    - list line 190, 192
    - name change 197
  - error processing 119

- installation-wide exits (*continued*)
  - exit
    - definition table 115
    - macros 249
    - parameters 119, 197
  - installing 114
  - possible uses for the routine 113
  - print utility exit 192
  - SELECT service start 252
  - TSO/E information center
    - facility 192
    - variables 192, 196
- interactive CLIST, FORTRAN, and COBOL 143
- introduction to ISPF/PDF 3
- invalid data set name character translation table 265
- IPV6 3
- ISPAPTT, APL2 terminal type dialog 105
- ISPCCSID translation load module
  - description 98
  - generation macro 100
- ISPCFIG 9
- ISPDFLTS 235
- ISPEXEC, external entry point in ISPLINK 104
- ISPEXITS, sample JCL 115
- ISPF
  - APL2 terminal type mappings 105
  - customizing
    - DM 79
    - ISPTCM 106
  - data set integrity enqueue 305
  - default table, generating 106
  - installation-wide exits 113
  - introduction 3
  - ISPPREP preprocessed panel utility 69
  - list data set allocation 83
  - load modules 6
  - log data set allocation 83
  - module names 80
  - preprocessed panel utility 69
  - session initialization exit 251
  - session termination exit 251
  - system command table, overriding
    - commands 84
  - tailoring defaults 106
  - terminal translation table, ISPF/PDF 251, 271
  - translation tables, creating 86
  - using with modified batch options 130
  - using with modified foreground options 130
- ISPF client/server
  - changing default name 79
- ISPF client/server feature 5
- ISPF configuration table 84
- ISPF Configuration Table 9
- ISPF configuration table fields 203
  - Data Set Allocation Settings 204
  - Default CUA Color Settings 227
  - DSLIT interface settings 207
- edit recovery data set 206
- Edit Site-wide Profile
  - Customizations 212

ISPF configuration table fields (*continued*)

- Edit-related Settings 209
- EDIT/BROWSE/VIEW 235
- ISPF Site-wide Profile
  - Customizations 217
- ISPSPROF General Values 236
- ISPSPROF Workstation Defaults 243
- ISPSPROF Workstation Download Defaults 247
- LMF 206
- Miscellaneous Settings 231
- Move/Copy Settings 207
- Outlist data set specifications 205
- PDF Exits 203
- Removable media interface settings 207
- SuperC Data set specifications 205
- Values Formerly in ISPDFLTS 235

ISPF configuration utility 12

- Build Configuration Table Load Module 33
- Build SMP/E USERMOD 35
- Convert Assembler Configuration Table to Keyword File 35
- Create/Modify Settings and Regenerate Keyword File 14
- Edit Keyword File Configuration Table 31
- using before installation complete 10
- Verify Keyword Table Contents 33

ISPF Dialog Tag Language panel models 293

ISPF services

- in C/370 programs 293
- in CLIST commands 273
- in COBOL programs 275
- in EXEC commands 278
- in FORTRAN programs 279
- in Pascal programs 287
- in PLI programs 285
- in TSO/REXX commands 290
- specific to LMF 129

ISPF Site-wide Profile

- Customizations 217

ISPF termination and LOG/LIST commands, Print utility exit 193

ISPF/PDF

- installation-wide exits 181
- interfaces 3
- licensed processing programs 3
- Product Option customizing 175

ISPILIB 8

ISPILIB image input library 39

ISPILIB in logon proc 8

ISPLIST, default values for 83

ISPMDAD macro 250

ISPMEPT macro 250

ISPMTCM macro

- how to use 106
- sample definition 109
- syntax 106

ISPMXDD macro 250

ISPMXDEF macro 249

ISPMXED macro 249

ISPMXEND macro 250

ISPMXLST macro 249

ISPPREP preprocessed panel utility 69

ISPMTCM (TSO command table)

- ISPMTCM macro 106
- sample definition 109
- usage notes 109

ISPTTDEF 94

ISPXDT (exit definition table)

- how to define 115
- main entry point of ISPEXITS 113
- sample definition 118

ISR@PRIM, PDF Primary Option Menu 130

ISRBROB, distributed Browse panel 168

ISREDIT, edit profiles set 122

ISREDRT, Edit backup and recovery table 123

ISREFR01, distributed Edit panel 168

ISRFRP 140

ISRJB1

- batch processing control program 159
- jobnames algorithm 160

ISRJB2 159

ISRLEMX

- customizing 163
- job steps 164
- parameters 164
- return codes 164

ISRLWKnn, temporary data sets 166

ISRSCAN

- customizing 163
- job steps 163
- return codes 164

ISRUOL, print data set attributes 124

**J**

JCL generated by batch processing 160

jobnames, algorithm for uniqueness 160

**K**

keyboard 309

Keyword File, Configuration Table 203

keywords, syntax diagrams viii

**L**

language default value, changing for session 111

Language Environment 5

LIBDEF service exit 110, 257

line mode support, setup 79

link-pack area 71

LINKLST 71

list data set

- default values 83
- preallocating to SYSOUT 83

LIST temporary data sets, preallocating 76, 78

listings, compressing 78

LMACT service 129

LMDEACT service 129

LMF

- converting to SCLM 127
- dialogs, converting to SCLM 129

LMHIER service 129

LMPROM service 129

LMREVIEW service 129

load modules

- adding to the LPA 71
- descriptions 72
- moving 71
- recommended for the LPA 72
- search order 6

loading APL2 workspace 105

log

- data set, preallocating to SYSOUT 83
- list, and temporary data set allocation exit 263

logical screen exit

- end 259
- start 259

LOGON procedure, TSO

- list and log data sets, preallocating to SYSOUT 83
- preprocessed utility, modifications for 70

lowercase character translation table (TTLOW) 87, 269

LPA 71

LPALST 71

**M**

macro names, complete list 299

mainframe

- education viii

mappings, ISPF-to-APL2 terminal type 105

member

- expansion function, adding user-defined trigger 166, 167
- list built-in line command exit 201
- list filter exit
  - analyze member names 200
  - analyze patterns 199
- list panels, customizing 172
- parts list function, adding user-defined trigger 167

menu

- PDF Primary Option Menu 130

message format, model listings for 282

migrating to ISPF Version 4.1

- from ISPF Version 2 or 3 3

Miscellaneous Settings 231

model listings, complete list of file tailoring control statements 287

ISPF Dialog Tag Language panel models 293

ISPF services

- in C/370 programs 293
- in CLIST commands 273
- in COBOL programs 275
- in EXEC commands 278
- in FORTRAN programs 279
- in Pascal programs 287
- in PLI programs 285
- in TSO/REXX commands 290

message format 282

panel formats and statements 282

SCLM

- architecture definition formats 292

- model listings, complete list of  
(*continued*)
  - SCLM (*continued*)
    - project definition macros and templates 293
- model names, complete list 273
- Models, Dialog Tag Language 296
- Move/Copy Settings 207
- multicultural support 6
  - installation 111
  - maintenance considerations 6
- multivolume data sets
  - edit recovery data set 124
  - temporary data set 78
- MVS environments, ISPF Version 3.3 support 3

## N

- National Language Support
  - See* multicultural support
- NLS
  - See* multicultural support
- non-ISPF TSO and BATCH 305
- Notices 311
- numeric character translation table 266

## O

- option
  - IBM products 175
  - panel, redisplay of in foreground processing 143
- optional ISPF libraries 39
- Outlist data set specifications 205
- OUTLIST temporary data sets, preallocating 76
- outlist utility, allocation defaults for 124

## P

- panels
  - batch processing 145
  - display sequence, changing 143, 145
  - Edit and Browse, customizing 168
  - foreground processing 130
  - formats and statements, model listings for 282
- parameter list for exit routines 118
- PDF
  - automatic browse and display, bypassing 143
  - batch, customizing 130
  - customizing 121
  - FORTRAN Interactive Debug option 141
  - installation-wide exits 181
  - introduction 3
  - Primary Option Menu 130
  - sample assembler modules 129
  - service end exit 260
  - service start exit 260
  - terminal translation table, ISPF/PDF 271
  - translation tables, creating 129
  - VS COBOL II Interactive Debug 141

- PDF Exits 203
- performance, improving
  - adding load modules to the LPA 71
  - changing the block size of the edit recovery data set 67
  - disabling edit extended highlighting 68
  - disabling SETUNDO storage 68
  - preprocessing ISPF panels 69
  - removing action bars from ISPF panels 68
  - removing scrollable areas from ISPF panels 69
  - VIO 67
- planning
  - ISPF installation 3
  - preprocessed panel utility 69
  - preprocessing ISPF panels 69
- Primary Option Menu
  - PDF 130
- print
  - data set attributes 124
  - function 142
  - utility exit 192
- Print utility exit
  - ISPF termination and LOG/LIST commands 193
- product option 175
- profile
  - Edit 121
  - sharing, customizing 45
- programming
  - interface macros for customers 299

## R

- recovery
  - Edit 123
- redisplay of option panel, modifying 143
- RELEASE exit 258
- removing edit functions 67
- repeatable items, syntax diagrams viii
- requirements
  - system 3
- RESERVE exit 258
- Resource Access Control Facility (RACF)
  - protects SCLM project data sets 6
- return codes
  - ISRLEMX 164
  - ISRSCAN 164
- RTNPNL, PDF batch processing variable 159

## S

- SBCS/DBCS CCSIDs 96
- SCEERUN run-time library data sets 5
- SCLM
  - architecture definition formats 292
  - batch considerations 125
  - converting from LMF 127
  - listings, compressing 78
  - project definition macros and templates 293

- scrollable areas, removing from ISPF panels 69
- search sequence for attaching commands 110
- SELECT service
  - end exit 254
  - start exit 252
- serializing 305
- services
  - ISPF services specific to LMF 129
- SETUNDO STORAGE, disabling 68
- shared variable pool, foreground processing 142
- shortcut keys 309
- site command tables 85
- site-wide
  - defaults, setting
    - CUA panel elements 84
    - KEYLIST 84
    - PFSHOW 84
    - SCROLL AREA 84
    - STATUS 84
  - edit profile initialization 122
- skeletons, batch processing 145, 161
- SMF 80
- SMP/E
  - ++MOVE statements 71
  - data set 36
  - USERMOD 35
- software components used by SCLM 5
- Software Configuration and Library Manager (SCLM)
  - migrating from LMF
    - data changes 5
    - library structure changes 5
  - project
    - accounting data 293
    - database 5
    - definition and control, contained in PROJDEFS 6
    - project accounting data 6
    - project cross-reference 6
    - RACF to protect SCLM project data sets 6
    - SCLM project definition macros and templates 293
    - started task 6
    - VSAM clusters 6
  - SPFTEMP temporary data sets, preallocating 76
  - split screens, specifying number of 83
  - square brackets, displaying in C programs 104
  - STATUS AREA 84
  - SUBMIT command 159
  - suboption panel, redisplay of, in batch processing 160
  - suboptions
    - PDF batch option 152
    - PDF foreground option 134
  - SuperC Data set specifications 205
  - SVCs
    - SVC 109 80
  - SWAP exit 261
  - syntax diagrams, how to read viii
  - SYSLIB 36
  - SYSMOD Identifier for USERMOD 36

system  
  command table 86  
  requirements 3  
System Management Facility (SMF)  
  command accounting 80  
  description 80

**T**

tailoring  
  edit functions 67  
  ISPF defaults 106  
  PDF defaults 9

tape data sets, configure support for 207  
TCP/IP daemon name 106

temporary data sets  
  default names 76  
  preallocating 76  
  restrictions on 78

terminal type  
  associating with translation tables 90  
  mappings, ISPF-to-APL2 105

terminals, base code pages for 97

trademarks 312

transition to ISPF Version 4.1 3

translation load  
  module generation macro,  
  ISPCCSID 100  
  modules, ISPCCSID 98

translation tables  
  alphabetic character (TTALB) 89  
  alphabetic character (TTALP) 89  
  alphanumeric character 266  
  collating sequence (TTCOL) 89  
  creating for ISPF 86  
  creating for ISPF/PDF 129  
  edit terminal output character 267  
  generic string character  
  translation table 267  
  usage notes 268  
  generic string master (TTGSM) 88  
  generic string special character  
  translation table 267  
  usage notes 268  
  hexadecimal character 265  
  invalid data set name character 265  
  lowercase character (TTLOW) 87, 269  
  master, generic string 269  
  modifying GSM 269  
  numeric character 266  
  sample assembler modules 129  
  uppercase character (TTUPP) 87, 268  
  valid data set name character 265  
  valid terminal output (TTVAL) 87

trigger, user-defined  
  adding to member expansion  
  function 166  
  adding to member parts list  
  function 167

TSO  
  command 255  
  end exit 256  
  table (ISPTCM) 106  
  line mode 8  
  line mode support setup 79  
  logon procedure example 7, 8  
  SUBMIT command 159

TSO (*continued*)  
  TSO line mode restriction 8  
TSO/ISPF client gateway,  
  description 46  
TSO/ISPF gateway  
  customizing 49  
  HTTP server, customizing 49  
  installing 48  
  other environments, configuration  
  for 50  
  Remote Systems Explorer,  
  customizing 50  
  using 51

**U**

Unicode  
  providing support in Browse 172

uppercase character translation table  
  ISPF 87  
  ISPF/PDF 268

user command tables 85

user-defined trigger  
  batch options 166  
  foreground options 166  
  ISRLEMX parameter description 165  
  member expansion function 166  
  member parts list function 167

usermod 71

utility  
  hard copy, with DBCS support 125  
  ISPF configuration 12  
  ISPF configuration, using before  
  installation complete 10  
  outlist (3.8), changing data set  
  allocation defaults 124  
  preprocessed panel utility  
  (ISPPREP) 69

**V**

valid  
  data set name character translation  
  table 265  
  terminal output translation table  
  (TTVAL) 87

Values Formerly in ISPDFLTS 235

variables, syntax diagrams viii

Verify Keyword Table Contents 33

VIO 67

virtual I/O 67

VSAM clusters  
  SCLM accounting data sets 6  
  SCLM project cross-reference 6

**W**

wildcard characters 74

WORK temporary data sets,  
  preallocating 77

**Z**

z/OS Basic Skills information center viii

z/OS UNIX directory list panel,  
  customizing 175

z/OS UNIX Directory List utility 5

ZDEFAULT  
  default edit profile 122  
  profile, creating 122

ZDSCKO, PDF batch processing  
  variable 159

ZFBROWS, foreground automatic browse  
  variable 142

ZFPRINT, foreground automatic print  
  variable 142

ZNEXTPN, controlling panel display  
  sequence  
  batch 161  
  foreground 143

ZSEL, foreground processing restriction  
  for 140



---

## Readers' Comments — We'd Like to Hear from You

Interactive System Productivity Facility (ISPF)  
Planning and Customizing  
z/OS Version 1 Release 11.0

Publication No. GC34-4814-08

We appreciate your comments about this publication. Please comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this book. The comments you send should pertain to only the information in this manual or product and the way in which the information is presented.

For technical questions and information about products and prices, please contact your IBM branch office, your IBM business partner, or your 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. IBM or any other organizations will only use the personal information that you supply to contact you about the issues that you state on this form.

Comments:

Thank you for your support.

Send your comments to the address on the reverse side of this form.

If you would like a response from IBM, please fill in the following information:

\_\_\_\_\_

Name

\_\_\_\_\_

Address

\_\_\_\_\_

Company or Organization

\_\_\_\_\_

Phone No.

\_\_\_\_\_

E-mail address



Fold and Tape

Please do not staple

Fold and Tape



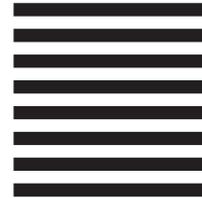
NO POSTAGE  
NECESSARY  
IF MAILED IN THE  
UNITED STATES

# BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Corporation  
Reader Comments  
DTX/E269  
555 Bailey Avenue  
San Jose, CA  
U.S.A. 95141-9989



Fold and Tape

Please do not staple

Fold and Tape





Program Number: 5694-A01

Printed in USA

GC34-4814-08

