Contents

Preface................................................................................................................................. ix
   About this document........................................................................................................ ix
   Who should use this document....................................................................................... ix
   What is in this document?................................................................................................ ix
   How to read the syntax diagrams.................................................................................... x

z/OS information................................................................................................................ xv

How to send your comments to IBM................................................................................ xvii
   If you have a technical problem...................................................................................... xvii

Summary of changes.............................................................................................................. xix
   Summary of changes for z/OS Version 2 Release 3 (V2R3)............................................. xix
   Summary of changes for z/OS Version 2 Release 2 (V2R2).............................................. xx
   Summary of changes for z/OS Version 2 Release 1 (V2R1).............................................. xx

What’s in the z/OS V2R3 ISPF library?............................................................................. xxx

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

   Chapter 1. z/OS V2R3 ISPF planning............................................................................ 3
      Hardware and software requirements............................................................................ 3
      Notes on migrating from previous releases to z/OS V2R3 ISPF............................... 3
      Installation, maintenance, and migration diagnostics.................................................. 3
      Language Environment considerations....................................................................... 4
      Software Configuration and Library Manager (SCLM)............................................... 4
      Maintenance considerations......................................................................................... 5
      Load module search order............................................................................................ 5
      TSO logon procedure..................................................................................................... 5
      Edit backup and recovery considerations...................................................................... 7

   Chapter 2. The ISPF Configuration Table................................................................. 9
      The ISPF Configuration utility...................................................................................... 9
      Create/Modify Settings and Regenerate Keyword File................................................. 11
      Edit Keyword File Configuration Table....................................................................... 29
      Verify Keyword Table Contents.................................................................................. 30
      Build Configuration Table Load Module..................................................................... 31
      Convert Assembler Configuration Table to Keyword File......................................... 32
      Build SMP/E USERMOD.......................................................................................... 33
      Convert Configuration Table LoadMod to Keyword File........................................... 35

   Chapter 3. Customizing ISPF.................................................................................... 37
      Allocating optional image ISPF library...................................................................... 37
      Customizing action bars.............................................................................................. 37
      Invoking the ISPDTLC conversion utility..................................................................... 38
      Customizing for profile sharing.................................................................................. 43
      Customizing the ISPF Configuration utility............................................................... 43
      Temporary data set names........................................................................................... 44
      SYSIKJUA enqueue...................................................................................................... 44
      Limitation of other products....................................................................................... 44
Changing the session language default value.................................................................136

Chapter 6. ISPF installation-wide exits........................................................................137
  How to install the installation-wide exits..............................................................138
  How to use the macros to define ISPXDT............................................................139
  Sample ISPXDT definition...................................................................................140
  Exit parameter list...............................................................................................141
  Error processing.................................................................................................142

Chapter 7. Customizing PDF.....................................................................................143
  Edit mode defaults............................................................................................143
    Site-wide edit profile initialization.................................................................143
    Creating a ZDEFAULT profile........................................................................144
    Action bars and extended color in Edit..........................................................144
    Edit backup and recovery..............................................................................145
  Data set allocation defaults for the Outlist utility..............................................145
  Using the Hardcopy utility with DBCS support....................................................146
  Creating PDF translation tables........................................................................148
  PDF Foreground and Batch customizing............................................................149
    Foreground processing panels and CLISTs......................................................149
    Batch processing panels, CLISTs, and skeletons..........................................161
    ISRSCAN and ISRLEMX programs...............................................................175
  Customizing Browse and Edit............................................................................180
    Providing customized Browse and Edit panels.............................................180
    Enabling Browse panels to display Unicode data..........................................184
  Customizing member list panels.......................................................................184
  Customizing the z/OS UNIX directory list panel..............................................187
  IBM Products option.........................................................................................187

Part 2. Reference.....................................................................................................191

Chapter 8. PDF installation-wide exits....................................................................193
  Data set list filter exit........................................................................................193
  Data set allocation exit......................................................................................194
  Activity monitoring exits..................................................................................198
  Exit 1: BROWSE primary command start.........................................................199
  Exit 2: BROWSE primary command end...........................................................199
  Exit 3: EDIT primary command start...............................................................199
  Exit 4: EDIT primary command end.................................................................200
  Exit 5: EDIT macro start....................................................................................200
  Exit 6: EDIT macro end.....................................................................................200
  Exit 7: EDIT line command start......................................................................200
  Exit 8: EDIT line command end......................................................................200
  Exit 9: Library/Data Set utility sub-function start..........................................200
  Exit 10: Library/Data Set utility sub-function end...........................................201
  Exit 11: Data Set List utility line command start............................................201
  Exit 12: Data Set List utility line command end..............................................202
  Data set list command exit..............................................................................203
  Print utility exit..............................................................................................204
  Print utility exit on ISPF termination and LOG/LIST commands..................206
  Compress request exit......................................................................................209
  Data set name change exit..............................................................................210
  Member list filter exit......................................................................................212
  Analyzing patterns with the member list filter exit.......................................212
  Analyzing member names with the member list filter exit............................213
  Member list built-in line command exit..........................................................214
Chapter 13. Programming interface macros for customers.......................................................... 321

Chapter 14. ISPF data set descriptions....................................................................................... 323

Appendix A. ISPF enqueue processing for data integrity.......................................................... 327
RESERVE processing for partitioned data set extended............................................................. 327
Serializing with non-ISPF TSO and BATCH.............................................................................. 327
ISPF data set integrity enqueue.................................................................................................. 328
Member name enqueue.............................................................................................................. 328
z/OS UNIX file Enqueue............................................................................................................. 328
ISPF GUI TSO line mode message queue enqueue................................................................. 329
Workstation file name enqueue............................................................................................... 329
SCLM VSAM enqueue.................................................................................................................. 329

Appendix B. Accessibility........................................................................................................... 331
Accessibility features.................................................................................................................... 331
Consult assistive technologies..................................................................................................... 331
Keyboard navigation of the user interface.................................................................................... 331
Dotted decimal syntax diagrams.............................................................................................. 331

Notices........................................................................................................................................ 335
Terms and conditions for product documentation................................................................. 336
IBM Online Privacy Statement................................................................................................. 337
Policy for unsupported hardware............................................................................................. 337
Minimum supported hardware................................................................................................. 338
Programming Interface Information....................................................................................... 338
Trademarks............................................................................................................................... 338

Index......................................................................................................................................... 339
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, exits, 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 V2R3 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 37 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 93 describes how to tune ISPF to reduce system load and improve response times for interactive users.
- Chapter 5, “Customizing DM,” on page 105 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 137 describes how you can write exit routines to collect system-related information.
- Chapter 7, “Customizing PDF,” on page 143 describes how to modify the distributed release of PDF to suit the needs of your installation.
- Chapter 8, “PDF installation-wide exits,” on page 193 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 217 lists the keywords in the ISPF Configuration Table and their allowable and default values.
• Chapter 10, “Exits,” on page 263 lists ISPF the installation-wide exits.
• Chapter 11, “PDF translation tables,” on page 281 lists the PDF translation tables.
• Chapter 12, “Dialog development model listings,” on page 289 lists all of the edit models included in the ISPF SKELS library.
• Chapter 13, “Programming interface macros for customers,” on page 321 lists the macros provided by ISPF as programming interfaces.
• Chapter 14, “ISPF data set descriptions,” on page 323 lists the target and distribution data sets used by ISPF, and their contents.
• Appendix A, “ISPF enqueue processing for data integrity,” on page 327 explains how ISPF ensures data integrity.

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 xi 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.
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.

---

*Figure 1. Sample syntax diagram*
Sometimes you only need to type the first few letters of a keyword. The required part of the keyword appears in uppercase letters.

```
DEFAULT_KEYWORD
    KEYword
```

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_variable
```

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

```
COMMAND_NAME — required_variable
```

**Optional items**  
Optional items appear below the main path.

```
OPTIONAL_KEYWORD= variable
```

**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.

```
required_choice1
    required_choice2
    required_choice3
```

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

```
optional_choice1
    optional_choice2
```

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

```
DEFAULT_KEYWORD
    KEYWORD1
    KEYWORD2
```

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

```
repeatable_item1
```
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.
This information explains how z/OS references information in other documents and on the web. When possible, this information uses cross document links that go directly to the topic in reference using shortened versions of the document title. For complete titles and order numbers of the documents for all products that are part of z/OS, see z/OS Information Roadmap.

To find the complete z/OS library, go to IBM Knowledge Center (www.ibm.com/support/knowledgecenter/SSLTBW/welcome).
How to send your comments to IBM

We invite you to submit comments about the z/OS product documentation. Your valuable feedback helps to ensure accurate and high-quality information.

Important: If your comment regards a technical question or problem, see instead “If you have a technical problem” on page xvii.

Submit your feedback by using the appropriate method for your type of comment or question:

Feedback on z/OS function
If your comment or question is about z/OS itself, submit a request through the IBM RFE Community (www.ibm.com/developerworks/rfe/).

Feedback on IBM® Knowledge Center function
If your comment or question is about the IBM Knowledge Center functionality, for example search capabilities or how to arrange the browser view, send a detailed email to IBM Knowledge Center Support at ibmkc@us.ibm.com.

Feedback on the z/OS product documentation and content
If your comment is about the information that is provided in the z/OS product documentation library, send a detailed email to mhvrdfs@us.ibm.com. We welcome any feedback that you have, including comments on the clarity, accuracy, or completeness of the information.

To help us better process your submission, include the following information:
• Your name, company/university/institution name, and email address
• The following deliverable title and order number: z/OS ISPF Planning and Customizing, GC19-3623-30
• The section title of the specific information to which your comment relates
• The text of your comment.

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

IBM or any other organizations use the personal information that you supply to contact you only about the issues that you submit.

If you have a technical problem
If you have a technical problem or question, do not use the feedback methods that are provided for sending documentation comments. Instead, take one or more of the following actions:
• Go to the IBM Support Portal (support.ibm.com).
• Contact your IBM service representative.
• Call IBM technical support.
Summary of changes

This information includes terminology, maintenance, and editorial changes. Technical changes or additions to the text and illustrations for the current edition are indicated by a vertical line to the left of the change.

Summary of changes for z/OS Version 2 Release 3 (V2R3)

The following changes are made for z/OS Version 2 Release 3 (V2R3).

June 2019
Maintenance and terminology changes are made for z/OS Version 2 Release 3 in June 2019.

New information
• “Prohibiting connections to the ISPF Workstation Agent” on page 105

Changed information
• Chapter 5, “Customizing DM,” on page 105
• “Pre-allocation of List/Log data sets” on page 109
• “Customizing the gateway calling environment” on page 82
• “Customizing IBM HTTP server powered by Apache” on page 82
• “Customizing Remote Systems Explorer” on page 89
• “Customizing other environments” on page 89
• “ISPF site-wide profile customizations” on page 230
• “Default CUA color settings” on page 240
• “ISPSPROF general values” on page 249

May 2018
Maintenance and terminology changes are made for z/OS Version 2 Release 3 in May 2018.

September 2017
Changed information
• TSO 8-character ID, see the following topics:
  – “General ISPF Settings Panels” on page 12
  – “Miscellaneous settings” on page 244
• Extended statistics usability, see the following topic:
  – “Edit site-wide profile customizations” on page 226
Summary of changes for z/OS Version 2 Release 2 (V2R2)

The following changes are made for z/OS Version 2 Release 2 (V2R2).

New

- Interactive ISPF Gateway support is provided for interactive communication and to manage the TSO/E sessions using z/OS CEA TSO/E address space services. See “Interactive ISPF Gateway” on page 46 for more information.
- The GLOBAL_LINE_COMMAND_TABLE keyword is added to define a default line command table under “Edit-related settings” on page 222.
- The GLOBAL_DISABLE_PACK keyword is added to disable the PACK operation used by the editor under “Edit-related settings” on page 222.
- The “Member name enqueue” on page 328 section is updated for PDSE member generations in Appendix A.

Changed

- Enhancements are added for the ISPF Gateway with IBM HTTP server powered by Apache in “Customizing IBM HTTP server powered by Apache” on page 82.
- The “Customizing Remote Systems Explorer” on page 89 section is updated under “Legacy ISPF Gateway” on page 69.

Summary of changes for z/OS Version 2 Release 1 (V2R1)

The following changes are made for z/OS Version 2 Release 1 (V2R1).

Extended attributes

- The extended member list format display has been updated to add an I(Info) command to display a panel with all the relevant extended information.

For changes to this document relating to this modification, see:
- “Exit 11: Data Set List utility line command start” on page 201

Updated samples

- The APL samples, ISPAPLTT and ISRAPLTT, have been updated to add APL characters X'70', X'76', and X'77'.

Edit recovery data sets

- ISPF has been enhanced to use TYPE=T on CLOSE for Edit Backup/Recovery data sets. The PARTIAL_RELEASE allocation restriction has been removed.
What's in the z/OS V2R3 ISPF library?

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

<table>
<thead>
<tr>
<th>Title</th>
<th>Order Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>z/OS ISPF Dialog Developer's Guide and Reference</td>
<td>SC19-3619–30</td>
</tr>
<tr>
<td>z/OS ISPF Dialog Tag Language Guide and Reference</td>
<td>SC19-3620–30</td>
</tr>
<tr>
<td>z/OS ISPF Edit and Edit Macros</td>
<td>SC19-3621–30</td>
</tr>
<tr>
<td>z/OS ISPF Messages and Codes</td>
<td>SC19-3622–30</td>
</tr>
<tr>
<td>z/OS ISPF Planning and Customizing</td>
<td>GC19-3623–30</td>
</tr>
<tr>
<td>z/OS ISPF Reference Summary</td>
<td>SC19-3624–30</td>
</tr>
<tr>
<td>z/OS ISPF Software Configuration and Library Manager Guide and Reference</td>
<td>SC19-3625–30</td>
</tr>
<tr>
<td>z/OS ISPF Services Guide</td>
<td>SC19-3626–30</td>
</tr>
</tbody>
</table>
Part 1. Planning and customizing
Chapter 1. z/OS V2R3 ISPF planning

Application dialogs that were created to run under earlier versions of ISPF will run under z/OS V2R3 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 is not supported.

As documented in the Choosing the z/OS base and optional features topic underPreparing the target system in z/OS Planning for Installation, because the z/OS base elements and optional features are integrated into a single package with compatible service levels, you must install, with few exceptions, the entire z/OS product. Using ISPF from one z/OS release with z/OS base elements or optional features from another z/OS release 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 V2R3 ISPF

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

IPV6

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

The z/OS V2R3 ISPF data sets are all named ISP.SISPxxxx and ISP.AISPxxxx.

Using SMP to install z/OS V2R3 ISPF deletes previous releases of ISPF.

If you have customized the ISPF configuration table for your installation, copy the customized configuration load module (ISPCFIGU) to a location that is in the standard MVS search sequence, or allocated to ISPLLIB, so that z/OS V2R3 ISPF will use the customized configuration. After you are running z/OS V2R3 ISPF, you can use the ISPF Configuration utility to set any new configuration options.

If you are running with an ISPLLIB DD statement and the z/OS V2R3 ISPF SISPLOAD and SISPLPA are not in the current LPA or the linklist, you must include the z/OS V2R3 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 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 linklist.

Installation, maintenance, and migration diagnostics

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

Problem: Unpredictable results such as OC4 or OC1 abends upon invocation of ISPF.

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 and an ISPLLIB is allocated, data sets allocated to STEPLIB should also be allocated to ISPLLIB.

Problem: New levels of code residing in STEPLIB do not appear to be executed. This includes changes to customer applications and changes made by PTFs.

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 and an ISPLLIB is allocated, data sets allocated to STEPLIB should also be allocated to 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.

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 the linklist 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 detailed information about SCLM, refer to z/OS ISPF Software Configuration and Library Manager Guide and Reference.
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 a 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 isREQed 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.

When using STEPLIB to test new maintenance 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 4 for more information about SCEERUN and SCEERUN2.

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

Figure 2 on page 6 is a sample TSO logon procedure (logon proc) to allocate the ISPF data sets required to execute ISPF.
//ISPF  PROC
//ISPF  EXEC PGM=IKJEFT01,DYNAMNBR=100
//*
//STEPLIB  DD DSN=ISP.SISPLPA,DISP=SHR
//         DD DSN=ISP.SISPLLOAD,DISP=SHR
//         DD DSN=CEE.SCEERUN,DISP=SHR
//         DD DSN=CEE.SCEERUN2,DISP=SHR
//*
//ISPMLIB  DD DSN=ISP.SISPMxxx,DISP=SHR
//          DD DSN=SYS1.SBPXMENU,DISP=SHR
//*
//*SYSHelp  DD DSN=ISP.SISPHELP,DISP=SHR
//*
//ISPPLIB  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.SISPxxx,DISP=SHR
//          DD DSN=SYS1.SBPXTENU,DISP=SHR
//*
//ISPLIB  DD DSN=ISP.ISPAPP,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
//*
OLD, or MOD, and the data sets must already exist. Or, these data sets must be temporary data sets. For more information, see “Customizing for profile sharing” on page 43.

If you are using TSO line mode support, your logon procedure must invoke the alternate entry point PGM=IKJEFT1I. 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 106 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 I.

If an ISPLLIB is allocated when you use a STEPLIB to test new ISPF maintenance, 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 4 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 6: 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.

### Edit backup and recovery considerations

Edit backup and recovery helps you to recover data that might otherwise be lost. For example, you would use edit recovery to re-establish the edit session at the point of failure after a power outage or system failure.

Edit backup and recovery is enabled when you enter Edit mode and recovery mode is on. You can set recovery mode on or off with the RECOVERY command.

If recovery mode is on, Edit writes data to an edit recovery data set which is used if a recovery is required. Some restrictions apply to the type of data set that can be used as an edit recovery data set.

For more information about backup and recovery, see “Edit backup and recovery” on page 145.
Chapter 2. The ISPF Configuration Table

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.

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 create or modify the settings and regenerate the keyword file. You can then convert the keyword file into a configuration table 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 9.

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.

Related references

Chapter 9, “ISPF Configuration Table keywords and values,” on page 217

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
• convert an existing ISPF configuration table load module 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 three 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.

Use the **Convert Configuration Table Loadmod to 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 3 on page 10.

---

**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
7. Convert Configuration Table Loadmod to Keyword File

**Keyword File Data Set**

Data Set: ______________________________
Member: __________________

**Configuration Table Assembler Source Data Set**

Data Set: ______________________________
Member: __________________

**Output File Content for Keyword File**

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 3. 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 “Create/Modify Settings and Regenerate Keyword File” on page 11).
2. Edit Keyword File Configuration Table (see “Edit Keyword File Configuration Table” on page 29).
3. Verify Keyword Table Contents (see “Verify Keyword Table Contents” on page 30).
4. Build Configuration Table Load Module (see “Build Configuration Table Load Module” on page 31).
5. Convert Assembler Configuration Table to Keyword File (see “Convert Assembler Configuration Table to Keyword File” on page 32).
6. Build SMP/E USERMOD (see “Build SMP/E USERMOD” on page 33).
7. Convert Configuration Table Loadmod to Keyword File (see “Convert Configuration Table Loadmod to Keyword File” on 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 4 on page 11.

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.

---

General ISPF Settings
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
6 Log and List Defaults
7 Terminal and User Defaults
8 Workstation Defaults
9 Workstation Download Defaults

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 12)
  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 23)
  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.
Modify PDF Edit Configuration Settings

Command ==> ____________ More: +

Miscellaneous Edit Settings
- Maximum Number of Edit Profiles . . . . 25
- Maximum Number of Edit Clipboards . . . 11
- Site-wide Initial Macro
- Maximum Initial Storage for Edit . . . . 0 (Number of 1K Blocks)
- Maximum Edit Clipboard Size . . . . 0 (Number of 4K Pages)
- Undo Storage Size . . . . 0 (Number of 1K Blocks)
- Text Flow Terminators . . . . :&<
- Edit CUT Default Action . . . . REPLACE (APPEND or REPLACE)
- Edit PASTE Default Action . . . . KEEP (DELETE or KEEP)
- Global Line Command Table
- Previous Generation Save Default Action NOGEN (NEWGEN or NOGEN)

Enter "/" to select option
/ Allow Edit Highlighting
/ Default Editor to have Highlighting Enabled
/ Highlight Assembler Continuation Errors
/ Default Editor to have Action Bars Present
/ Warn on Trailing Blank Truncation
/ Allow Creation of CREATE/REPLACE Target Data Set
- Force ISRE776 if RCHANGE passed arguments
- Enable Extended Statistics
- Disable PACK globally
F1=Help F2=Split F3=Exit F7=Backward F8=Forward F9=Swap F12=Cancel

The ISPF Configuration Table 13
### Modify PDF Edit Configuration Settings

**Command ==>**

<table>
<thead>
<tr>
<th>Option</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>/ STATS ON</td>
<td>_ STATS</td>
</tr>
<tr>
<td>/ RECOVERY ON</td>
<td>_ RECOVERY</td>
</tr>
<tr>
<td>/ RECOVERY warning message</td>
<td>_ RECOVERY warning</td>
</tr>
<tr>
<td>/ SETUNDO ON</td>
<td>_ SETUNDO</td>
</tr>
<tr>
<td>/ PACK ON</td>
<td>_ PACK</td>
</tr>
<tr>
<td>/ CAPS ON</td>
<td></td>
</tr>
<tr>
<td>/ NOTE ON</td>
<td></td>
</tr>
<tr>
<td>/ NUMBER ON</td>
<td></td>
</tr>
<tr>
<td>/ AUTONUM ON</td>
<td></td>
</tr>
<tr>
<td>/ AUTOLIST ON</td>
<td></td>
</tr>
<tr>
<td>/ PROFILE LOCK</td>
<td></td>
</tr>
<tr>
<td>/ AUTOSAVE ON</td>
<td></td>
</tr>
<tr>
<td>/ AUTOSAVE PROMPT</td>
<td></td>
</tr>
</tbody>
</table>

**Edit Highlighting**

**Language**

<table>
<thead>
<tr>
<th>Option</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Automatic determination</td>
<td>_ HILITE ON</td>
</tr>
</tbody>
</table>

**Margins**

<table>
<thead>
<tr>
<th>Language</th>
<th>Left</th>
<th>Right</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>PL/I</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>PL/X</td>
<td>*</td>
<td>*</td>
</tr>
</tbody>
</table>

---

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

---

**Command ==>**

<table>
<thead>
<tr>
<th>Option</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>2. Assembler</td>
<td></td>
</tr>
<tr>
<td>3. PL/I</td>
<td>Highlight DO/END logic</td>
</tr>
<tr>
<td>4. COBOL</td>
<td>Highlight IF logic</td>
</tr>
<tr>
<td>5. Pascal</td>
<td></td>
</tr>
<tr>
<td>6. C</td>
<td>Match parentheses</td>
</tr>
<tr>
<td>7. BookMaster</td>
<td>_ Highlight FIND strings</td>
</tr>
<tr>
<td>8. REXX</td>
<td>_ Highlight cursor position</td>
</tr>
<tr>
<td>9. ISPF Panel</td>
<td></td>
</tr>
<tr>
<td>10. ISPF Skeleton</td>
<td></td>
</tr>
<tr>
<td>11. JCL</td>
<td></td>
</tr>
<tr>
<td>12. ISPF DTL</td>
<td></td>
</tr>
<tr>
<td>13. Other (CLIST, etc.)</td>
<td></td>
</tr>
<tr>
<td>14. Default (no language)</td>
<td></td>
</tr>
<tr>
<td>15. PL/X</td>
<td></td>
</tr>
<tr>
<td>16. IDL</td>
<td></td>
</tr>
<tr>
<td>17. SuperC Listing</td>
<td></td>
</tr>
<tr>
<td>18. HTML</td>
<td></td>
</tr>
<tr>
<td>19. XML</td>
<td></td>
</tr>
</tbody>
</table>

**Margins**

<table>
<thead>
<tr>
<th>Language</th>
<th>Left</th>
<th>Right</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>PL/I</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>PL/X</td>
<td>*</td>
<td>*</td>
</tr>
</tbody>
</table>

---

*Figure 8. 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).

```
<table>
<thead>
<tr>
<th>Command</th>
<th>Modify Edit/View/Browse VSAM Settings</th>
<th>Row 1 to 2 of 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>VSAM Enablement</td>
<td>Enter &quot;/&quot; to select option</td>
<td></td>
</tr>
<tr>
<td>VSAM Commands</td>
<td>FMNINV DSE /</td>
<td></td>
</tr>
<tr>
<td>VSAM Edit Command</td>
<td>FMNINV DSB /</td>
<td></td>
</tr>
<tr>
<td>VSAM Browse Command</td>
<td>FMNINV DSV /</td>
<td></td>
</tr>
</tbody>
</table>
```

**Figure 9. 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.

```
<table>
<thead>
<tr>
<th>Command</th>
<th>Modify PDF Configuration Settings</th>
<th>More:</th>
</tr>
</thead>
<tbody>
<tr>
<td>PDF Exits</td>
<td>Data Set Allocation Program Exit . . . . . . . . .</td>
<td></td>
</tr>
<tr>
<td>Print Utility Program Exit . . . . . . . . . . . .</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Compress Program Exit . . . . . . . . . . . . . .</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Compress Command Exit . . . . . . . . . . . . . .</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Set List Filter Program Exit . . . . . . . .</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Member List Filter Program Exit . . . . . . . .</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Set Name Change Program Exit . . . . . . . .</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Set List Line Command Program Exit . . . . . . . .</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Activity Monitoring Program Exit . . . . . . . . . .</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Member List Line Command Program Exit . . . . . . . .</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Member List Line Command Command Exit . . . . . . . .</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

PDF Data Set Characteristics

```
| Record Length . . . 133 | SuperC Block Sizes |
| Block Size . . . 13566 | 0 |
```

**Figure 10. Modify PDF Configuration Settings panel (ISPPMOD3)**
Modify PDF Configuration Settings

Command ===>

<table>
<thead>
<tr>
<th>Block Size</th>
<th>13566</th>
<th>Update Data Set</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary Blocks</td>
<td>200</td>
<td>Profile Data Set</td>
<td>0</td>
</tr>
<tr>
<td>Secondary Blocks</td>
<td>100</td>
<td>Statements Data Set</td>
<td>0</td>
</tr>
<tr>
<td>Listing Primary Quantity</td>
<td>50</td>
<td>Listing Secondary Quantity</td>
<td>100</td>
</tr>
<tr>
<td>Listing Secondary Quantity</td>
<td>50</td>
<td>Update Primary Quantity</td>
<td>15</td>
</tr>
<tr>
<td>Update Secondary Quantity</td>
<td>30</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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 11. Modify PDF Configuration Settings panel (ISPPMOD3)

Modify PDF Configuration Settings

Command ===>

<table>
<thead>
<tr>
<th>When to use COPY or COPYMOD</th>
</tr>
</thead>
<tbody>
<tr>
<td>2  1. Use COPY if the target block size is equal to or greater than the source block size, COPYMOD otherwise</td>
</tr>
<tr>
<td>2. Use COPY if the target block size is equal to the source block size, COPYMOD otherwise</td>
</tr>
<tr>
<td>3. Always use COPYMOD</td>
</tr>
</tbody>
</table>

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 12. Modify PDF Configuration Settings panel (ISPPMOD3)
Modify PDF Configuration Settings
Command ===>

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

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

Figure 13. Modify PDF Configuration Settings panel (ISPPMOD3)

Modify PDF Configuration Settings
Command ===>

| 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

When to use 8-character user ID layouts
1. When 8-character user IDs are enabled on the system
2. Always

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.
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 field. 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 . . . 43003

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)

Command ==>

Modify ISPF Sitewide Defaults

More: - +

scroll defaults                      Status area default
Minimum scroll value 1
Maximum scroll value 9999999
Select Option 7.1 Dialog Test Panel
1  1. ISPYPFP
2  2. ISPYPFPA
3  3. ISPYPFPA

Command line placement
1  1. Bottom
2  2. Asis
PRINTDS Option
1  1. DEST
2  2. WRITER

Reset flags
Minimum scroll value 1
Maximum scroll value 9999999
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 ===>**

Maximum Scroll Value: 9999999

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

**ISPF Data Set Characteristics**

<table>
<thead>
<tr>
<th>Log Data Set</th>
<th>List Data Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Record Length . . . : 125</td>
<td>Records per Block . . 0</td>
</tr>
<tr>
<td>Block Size . . . . . : 0</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ISPCTLx Data Set</th>
<th>ISPLSTx Data Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Record Length . . . : 80</td>
<td>Record Length . . . : 121</td>
</tr>
<tr>
<td>Block Size . . . . . : 00</td>
<td>Block Size . . . . : 0</td>
</tr>
<tr>
<td>Primary Quantity . . : 10</td>
<td>Primary Quantity . . : 1</td>
</tr>
<tr>
<td>Secondary Quantity . : 100</td>
<td>Secondary Quantity . : 10</td>
</tr>
</tbody>
</table>

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 ===>**

<table>
<thead>
<tr>
<th>ISPCTLx Data Set</th>
<th>ISPWRKx Data Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Record Length . . : 80</td>
<td>Record Length . . : 256</td>
</tr>
<tr>
<td>Block Size . . . . : 800</td>
<td>Block Size . . . . : 0</td>
</tr>
<tr>
<td>Primary Quantity . : 100</td>
<td>Primary Quantity . : 999</td>
</tr>
<tr>
<td>Secondary Quantity . : 100</td>
<td>Secondary Quantity . : 999</td>
</tr>
</tbody>
</table>

LOG Data Set Process Option
- 1. Process Option Not Set
- 2. Print and Delete
- 3. Delete Without Printing
- 4. Keep
- 5. Keep and Allocate a New LOG

LIST Data Set Process Option
- 1. Process Option Not Set
- 2. Print and Delete
- 3. Delete Without Printing
- 4. Keep
- 5. Keep and Allocate a New LIST

Enter "/" to select option
- / Use Default PDF Unit for ISPF Data Sets
- _ Use Additional Qualifier for PDF Output Data Sets

Temporary Data Sets
- Additional Qualifier . .

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: - + 

ISPF Multi-logon Profile Options

Enter "/" to select option

- Multi-logon Profile Sharing
  1. Prompt for Profile ENQ Lockout
  2. ENQ Lock Wait . . . . . 1000
  3. Reset Shared Profile Settings

- ENQ Lock Retry Count . .1

System Profile conflicts

1  1. Keep
2  2. Discard
3  3. Prompt

Reference List conflicts

1  1. Keep
2  2. Discard
3  3. Prompt

ISPF Profile conflicts

1  1. Keep
2  2. Discard
3  3. Prompt

Edit Profile conflicts

1  1. Keep
2  2. Discard
3  3. Prompt

Application Profile conflicts

1  1. Keep
2  2. Discard
3  3. Prompt

Batch Profile conflicts

1  1. Keep
2  2. Discard
3  3. Prompt

ISPF Profile conflicts

1  1. Keep
2  2. Discard
3  3. Prompt

Application Profile conflicts

1  1. Keep
2  2. Discard
3  3. Prompt

ISPF PDSE Enqueue Processing

Select this option to prevent ISPF from issuing a data set reserve for PDSE

ISPF Init Command

Local PRINTDS Options . . . NONUM

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

Figure 19. Modify ISPF Sitewide Defaults panel 5 (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

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 .
APPLID for Site Command Table 3 .

Miscellaneous DM Settings
Maximum Number of Split Screens . 8

ZDATEF may use the national language convention to replace the characters YY, MM, DD and the national language delimiter.

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

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

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

F1=HELP F2=SPLIT F3=END F4=RETURN F5=RFIND F6=CHANGE
F7=UP F8=DOWM 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.

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

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 - DD.MM.YY

F1=HELP F2=SPLIT F3=END F4=RETURN F5=RFIND F6=CHANGE
F7=UP F8=DOWM 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 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=CHANGE
F7=UP F8=DOWM 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 on page 23 shows the options for the first two panel elements.
Modify ISPDFTS 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 24. Modify ISPDFTS and Other DM Settings panel (ISPPMOD5)

Modify Log/List Configuration Settings

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

Log/List Job Cards

Log/List Job Card1
Log/List Job Card2
Log/List Job Card3
Log/List Job Card4

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

Log/List Job Card4

Unique Job Character .. _

Log Settings

Enter "/" to select option

- DS Unique Char . . . . . . 1
- Lines Per Page . . . . . . 60
- Primary Quantity . . . . . 10
- Secondary Quantity . . . . 10
- Batch SYSOUT Class . . . .
- Local Printer ID or
- Local SYSOUT Class . . . .

List Settings

Enter "/" to select option

- Logical Record Length . . . . 121
- Primary Quantity . . . . . 100
- Secondary Quantity . . . . 200
- Batch SYSOUT Class . . . .
- Local Printer ID or
- Local SYSOUT Class . . . .

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

Modify Log/List Configuration Settings

Command ===>

writer-name . . . . .
Local SYSOUT Class . . .

List Settings

Enter "/" to select option

- Logical Record Length . . . . 121
- Primary Quantity . . . . . 100
- Secondary Quantity . . . . 200
- Local Printer ID or
- Local SYSOUT Class . . . .

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

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 set to 43000. This value does not change with new versions or releases of ISPF.

**Sitewide Defaults Version Level** 43001

<table>
<thead>
<tr>
<th>Settings Panel Defaults</th>
<th>Window Frame Color/Intensity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aspect Ratio . . . . . . . 0</td>
<td>Color</td>
</tr>
<tr>
<td>Input Field Pad Character B</td>
<td>1. Blue</td>
</tr>
<tr>
<td>Delimiter . . . . . . .</td>
<td>2. Red</td>
</tr>
<tr>
<td>Family Printer . . . . . : 2</td>
<td>3. Pink</td>
</tr>
<tr>
<td>Device Name . . . . . . .</td>
<td>4. Green</td>
</tr>
<tr>
<td></td>
<td>5. Turquoise</td>
</tr>
<tr>
<td></td>
<td>6. Yellow</td>
</tr>
<tr>
<td></td>
<td>7. White</td>
</tr>
</tbody>
</table>

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

Enter "/" to select option

Reset Device Name

**Terminal Type**

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. 3278H0
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

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

Command ===>

More: - +

Environ Defaults
 Termtrac DD Name . . ISP$SNAP
  1

Enter "/" to select option
 _ Reset Termtrac DD Name

Environ Enbldump Environ Termtrac
  2  1. ON
  2. OFF

Retrieve Defaults
Retrieve Minimum Length ..

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

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

Number of Keys
FKA Setting
 Enter "/" to select option
 _ Display Prefix

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

F12=Cancel

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

Modify Terminal and User Settings

Command ===>

More: - +

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

Global Colors
 Enter a numeric color code to change a default color.
 For example, to change everything that is currently blue to yellow, enter 6
 in the Blue entry field.
(1) Blue . . . . 1
(2) Red . . . . 2
F1=Help F2=Split F3=Exit F7=Backward F8=Forward F9=Swap

F12=Cancel

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

Command ==> ____________________________ More: -

in the Blue entry field.
(1) Blue . . . . . . . . 1
(2) Red . . . . . . . . 2
(3) Pink . . . . . . . . 3
(4) Green . . . . . . . 4
(5) Turquoise . . . . 5
(6) Yellow . . . . . . . 6
(7) White . . . . . . . . 7

Enter "/" to select option
- GUI Host Colors (/=ON)
- GUI PC Colors (/=ON)
- Message Identifier (/=ON)
- Panel Identifier (/=ON)
- Screen Name Identifier (/=ON)
- System Name Identifier (/=ON)
- User Identifier (/=ON)
- Enable Euro Symbol (/=YES)
- Reset Enable Euro Symbol (/=YES)

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

Figure 32. 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.

Workstation 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 field set to 43000. 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

GUI APPC Addresses
  GUI APPC Address1
  GUI APPC Address2
  GUI APPC Address3
  GUI APPC Address4
  GUI APPC Address5

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 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 34. Workstation Defaults panel (ISPPMOD8)

Workstation Defaults

Command ===>

More: - +

/ GUI Display ENTER Key
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
/
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 field. 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

WSA Download FTP User . . .
WSA Download Data Set . . 'ISP.SISPGUI'
Enter "/" to select option
Reset WSA Download Data Set

WSA Down Level Warning
1  1. Yes            2. No
F1=Help      F2=Split     F3=Exit      F7=Backward  F8=Forward   F9=Swap
F12=Cancel

WSA Download Directory

WSA Download FTP Address

Workstation Download Option                  Workstation Download Platform
1  1. Download Now                     1. Windows
2. Do Not Download                     2. Reserved
3. Cease Warnings                     3. AIX

Workstation Download Method
-  1. Download Using FTP
   2. Download Using ISPF/Client Server
   3. Download Manually
Enter "/" to select option
   /  Browse FTP Errors
   \  Create Directory

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

Figure 36. 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 37 on page 30.

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.

/* ISPF Configuration table definition. Generated by REXX ISPCMOD */
/* Created 10:40:04 on 9 Jan 2004 */
/* by user LSACKV. */
/* All values were included. */
/* */
 /*--------------------------------------------------------------------*/
 /*--------------------------------------------------------------------*/
DATA_SET_ALLOCATION_PROGRAM_EXIT            = NONE
PRINT_UTILITY_PROGRAM_EXIT                  = NONE
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 37. 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 38 on page 30.

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 38. 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 39 on page 31 is displayed. The listing shows the line in error and the reason that ISPF flagged it as incorrect.
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 40 on page 32.

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 any of the VSAM restriction keywords (VSAM_RESTRICTED_EDIT_DATASET, VSAM_RESTRICTED_BROWSE_DATASET, VSAM_RESTRICTED_VIEW_DATASET, or VSAM_RESTRICTED_ALL_DATASET) is used in your keyword file. These VSAM restriction keywords are associated with the fields in the Restricted Data Sets section of the Modify Edit/View/Browse VSAM Settings panel (ISPPMOD2) of the ISPF Configuration Utility.

**Note:** If you convert your keyword file when it contains any of the VSAM restriction keywords, a VSAM configuration load module is created. If you later remove every VSAM restriction keyword from the keyword file and convert it again, the VSAM configuration load module is not created. However, the previously created VSAM configuration load module is not deleted. Because the ISPF Configuration Utility does not know the name that was used for the previously created VSAM configuration load module, it is your responsibility to delete it from all locations where it exists.

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.
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.
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 41 on page 33.

---

**Figure 41. 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 2 Release 3.0 is HIF7R02.
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.

**Note:**

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 for z/OS Reference for more information about the requirement for SMPTLOAD.

**Convert Configuration Table LoadMod to Keyword File**

If you choose Option 7, Convert Configuration Table LoadMod to Keyword File on the ISPF Configuration utility main menu panel, ISPF converts a configuration table load module to a keyword file that can be used as input to Options 1 and 2. The active configuration or a module from a data set can be chosen.

This can be used to reconstruct source that may have been lost.

All keywords are produced.
Chapter 3. Customizing ISPF

Here are the ways in which you can customize ISPF.

Allocating optional image ISPF library

The data set described in Table 1 on page 37 is not a required ISPF library. You must allocate it only if an application in GUI mode uses images.

<table>
<thead>
<tr>
<th>DDNAME</th>
<th>Description</th>
<th>RECFM</th>
<th>LRECL</th>
<th>BLKSIZE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISPILIB</td>
<td>Image Input Library</td>
<td>FB</td>
<td>80</td>
<td>(See note)</td>
</tr>
</tbody>
</table>

**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 on page 37. These embed files are distributed in the ISP.SISPGENU library.

<table>
<thead>
<tr>
<th>File name</th>
<th>DTL File for…</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISPDFIL1</td>
<td>File action bar choice</td>
</tr>
<tr>
<td>ISPFMENU</td>
<td>SCLM menu action bar choice</td>
</tr>
<tr>
<td>ISPDFUNC</td>
<td>Functions action bar choice</td>
</tr>
<tr>
<td>ISPDHUCM</td>
<td>Action bar choice</td>
</tr>
<tr>
<td>ISPDHYXM</td>
<td>Action bar choice</td>
</tr>
<tr>
<td>ISPFJOBFC</td>
<td>Jobcard action bar choice</td>
</tr>
<tr>
<td>ISPDLANG</td>
<td>Compilers action bar choice</td>
</tr>
<tr>
<td>ISPDMENU</td>
<td>Menu action bar choice</td>
</tr>
<tr>
<td>ISPDREFE</td>
<td>Reflist action bar choice</td>
</tr>
</tbody>
</table>
Table 2. Embedded DTL source files (continued)

<table>
<thead>
<tr>
<th>File name</th>
<th>DTL File for...</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISPDREFL</td>
<td>Reflist action bar choice</td>
</tr>
<tr>
<td>ISPDREFM</td>
<td>Refmode action bar choice</td>
</tr>
<tr>
<td>ISPSAVE</td>
<td>Save action bar choice</td>
</tr>
<tr>
<td>ISPFSCLM</td>
<td>SCLM action bar choice</td>
</tr>
<tr>
<td>ISPDTEST</td>
<td>Test action bar choice</td>
</tr>
<tr>
<td>ISPDUTIL</td>
<td>Utilities action bar choice</td>
</tr>
</tbody>
</table>

2. Reconvert the affected panels by specifying the appropriate list of members (Table 3 on page 38) and running the ISPDTLC conversion utility.

Table 3. DTL list of panels

<table>
<thead>
<tr>
<th>File name</th>
<th>DTL List for...</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISPFAB</td>
<td>All action bar GMLs</td>
</tr>
<tr>
<td>ISPFALL</td>
<td>All product GMLs</td>
</tr>
<tr>
<td>ISPFIL1</td>
<td>File action bar choice</td>
</tr>
<tr>
<td>ISPMENU1</td>
<td>SCLM menu action bar choice</td>
</tr>
<tr>
<td>ISPFUNC</td>
<td>Functions action bar choice</td>
</tr>
<tr>
<td>ISPHUCM</td>
<td>Action bar choice</td>
</tr>
<tr>
<td>ISPHYXM</td>
<td>Action bar choice</td>
</tr>
<tr>
<td>ISPJOBIC</td>
<td>Jobcard action bar choice</td>
</tr>
<tr>
<td>ISPLANG</td>
<td>Compilers action bar choice</td>
</tr>
<tr>
<td>ISPMENU</td>
<td>Menu action bar choice</td>
</tr>
<tr>
<td>ISPREFE</td>
<td>Reflist action bar choice</td>
</tr>
<tr>
<td>ISPREFL</td>
<td>Reflist action bar choice</td>
</tr>
<tr>
<td>ISPREFM</td>
<td>Refmode action bar choice</td>
</tr>
<tr>
<td>ISPSAVEB</td>
<td>Save action bar choice</td>
</tr>
<tr>
<td>ISPPSCLM</td>
<td>SCLM action bar choice</td>
</tr>
<tr>
<td>ISPTTEST</td>
<td>Test action bar choice</td>
</tr>
<tr>
<td>ISPUTIL</td>
<td>Utilities action bar choice</td>
</tr>
</tbody>
</table>

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

**Invoking the ISPDTLC conversion utility**

ISPDTLC 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 95.

For more information about ISPDTLC 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:

```
ISPDTLC
```

Figure 42 on page 39 shows the first screen of the ISPDTLC invocation panel with the default data set names and with the required options selected.

![First ISPDTLC screen](image)

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 43 on page 40).
Enter ISR as the Keylist Application ID and select the options as shown. Scroll to the next screen (Figure 44 on page 40).

Scroll to the next screen (Figure 45 on page 41).
Scroll to the last screen (Figure 46 on page 41).

**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 47 on page 42 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.
Figure 47. Sample JCL for Batch conversion

The CONVACTB EXEC (written in REXX) is:

```rexx
'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`.

**Note:**

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.

```rexx
'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.

```rexx
'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:

- **DTLGML**
  
  `your.appl.gml.dataset`

- **DTLGML**
  
  `ISP.SISPGENU`

- **DTLGML**
  
  `ISP.SISPGMLI`

- **DTLGML**
  
  `ISP.SISPSAMP`

- **DTLPAN**
  
  `your.output.panel.dataset`

- **DTLMSG**
  
  `your.output.message.dataset`

- **DTLLOG**
  
  `your.log.dataset`
Your .gml dataset (profile) for creating uppercased English panels should include:

DTLGML
your.appl.gml.dataset

DTLGML
ISP.SISPGENU

DTLGML
ISP.SISPGMLI

DTLGML
ISP.SISPSAMP

DTLPAN
your.output.uppercase.panel.dataset

DTLMSG
your.output.message.dataset

DTLLOG
your.log.dataset

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 ISPDTLC 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 230.
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 ISPFTTRC. 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

The following limitation applies to shared profiles:

**TSO/E Reconnect**

TSO/E users can only reconnect to their sessions from the same system where they were already logged on, not from a remote z/OS system.

**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 a thin gateway service that allows applications to start a TSO address space, send a TSO or ISPF command as input, and receive a response as output. Callers of this service are typically applications that are serving remote clients that want to issue TSO or ISPF commands. This section describes how the gateway works, how to install and customize the gateway, and how client applications can use the gateway.

Starting from z/OS V2R2, the “Interactive ISPF Gateway” provides an improved ISPF gateway service for callers to start a TSO address space. The new service provides all of the same capabilities as the existing “Legacy ISPF Gateway” and many more. While the Legacy ISPF Gateway is still provided, users should consider migrating to the Interactive ISPF Gateway to benefit from the new features. Future enhancements for the TSO/ISPF client gateway will be made to the Interactive ISPF Gateway.

Advantages of using the Interactive ISPF Gateway:

- The TSO/E address spaces are started by using a TSO logon procedure. Sites can choose to use their existing TSO logon procedures for TSO/E address spaces started through the gateway.
- There is support for callers to execute programs that are interactive, using a conversational pattern.
- Attention interrupts can be sent to terminate commands in progress.
- The Interactive ISPF Gateway uses z/OS Common Event Adapter (CEA) TSO/E address space services to start and manage TSO/E address spaces. The z/OS CEA TSO/E address space services provide support for the reuse of TSO/E address spaces, improving performance when a single TSO/E address space is used to issue multiple TSO or ISPF commands.
- The gateway can capture, and return to the client, messages that are issued using the TPUT and WTO macros and messages issued by TSO/E REXX.
- TSO/E address spaces that are started using CEA TSO/E address space services have the same characteristics as TSO/E address spaces for users who have logged in through the TSO/E logon screen.
Therefore, existing processes used to manage TSO/E address spaces also apply to address spaces that are started using CEA TSO/E address space services. For example the operator command:

\[ C \text{ U= } \text{user-id, A=asid} \]

can be used to cancel an address space that is started using CEA TSO/E address space services.

Requirements for using the Interactive ISPF Gateway:

- Callers must define the CGI_CEATSO environment variable and set it to the value TRUE.
- Callers must use Amode 64 because this is required by the underlying CEA TSO/E address space services.
- A system administrator must configure and start the z/OS CEA address space.

For more information on CEA TSO/E address space services, refer to CEA TSO/E address space services in \textit{z/OS MVS Programming: Callable Services for High-Level Languages}. For information on preparing to use the Interactive ISPF Gateway, see “Preparing to use the Interactive ISPF Gateway” on page 47.

Table 4 on page 45 describes the load modules that comprise the TSO/ISPF client gateway. The footnotes indicate the required load modules for using the Interactive ISPF Gateway and the required load modules for using the Legacy ISPF Gateway.

<table>
<thead>
<tr>
<th>Name</th>
<th>Library</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISPZINT</td>
<td>ISP.SISPLPA</td>
<td>Gateway initialization - routes processing to ISPZINO or ISPZINL. (1,2)</td>
</tr>
<tr>
<td>ISPZINO</td>
<td>ISP.SISPLPA</td>
<td>Legacy ISPF Gateway - manages TSO sessions using spawned processes, routes command requests to these sessions. (2)</td>
</tr>
<tr>
<td>ISPZINL</td>
<td>SYS1.SIEALNKE</td>
<td>Interactive ISPF Gateway – manages TSO sessions by using CEA TSO/E address space services and routing command requests to these sessions. (1)</td>
</tr>
<tr>
<td>ISPZTSO</td>
<td>SYS1.LINKLIB</td>
<td>TSO initialization - attaches a TSO session to run a command. ISPZTSO must run APF-authorized. (2)</td>
</tr>
<tr>
<td>ISPZCNT</td>
<td>ISP.SISPLOAD</td>
<td>ISPF initialization - allocates data sets and starts an ISPF session. (2)</td>
</tr>
<tr>
<td>ISPZCMD</td>
<td>ISP.SISPLOAD</td>
<td>ISPF command interface - invokes an ISPF command. (1,2)</td>
</tr>
<tr>
<td>ISPZTMO</td>
<td>ISP.SISPLOAD</td>
<td>Reusable session time-out processor. (2)</td>
</tr>
</tbody>
</table>

1 – Load module is used by the Interactive ISPF Gateway.
2 – Load module is used by the Legacy ISPF Gateway.

Table 5 on page 45 describes the TSO/ISPF client gateway files installed into the z/OS UNIX file system. The footnotes indicate the required files for using the Interactive ISPF Gateway and the required files for using the Legacy ISPF Gateway.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/usr/lpp/ispf/bin/ISPZINL</td>
<td>A stub file with the sticky bit set on to enable invocation of the Interactive ISPF Gateway load module ISPZINL. (1)</td>
</tr>
</tbody>
</table>
### Interactive ISPF Gateway

This section provides information about preparing for and using the Interactive ISPF Gateway. For information about the Interactive ISPF Gateway and the Legacy ISPF Gateway and the benefits of using the Interactive ISPF Gateway, see “TSO/ISPF client gateway” on page 44.

To use the Interactive ISPF Gateway, define the environment variable CGI_CEATSO and set it to the value TRUE before invoking the gateway.

The Interactive ISPF Gateway is invoked and runs under the caller’s address space. It then uses the z/OS CEA TSO/E address space services to start and communicate with the TSO address space on behalf of the caller. The types of requests that can be issued by the caller are:

- Start a TSO/E address space. Available options allow the client to specify whether ISPF is started in the address space and whether a dormant TSO session is used if available.
- Issue a TSO command (or ISPF command, if ISPF is started) in the TSO/E address space.
- Communicate interactively with the command running in the TSO/E address space.
- Ping a started TSO/E address space. If 15 minutes pass without receipt of a ping for the address space, CEA TSO/E address space services ends the address space.
- Send an attention interrupt to the command running in the TSO/E address space.
- End a TSO/E address space. Available options allow the client to specify whether the address space is cancelled, logged off, or allowed to become dormant.

Some types of gateway requests (for example, start a TSO/E address space, issue a command in the address space, and end the address space) can be combined in a single request. Upon completion of a request, the gateway returns the results to the client. Examples of items returned are the return code value, the output from the command run in the address space, and the information needed to reuse the address space for a subsequent command.

---

#### Table 5. Gateway files installed into the z/OS UNIX file system (continued)

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/usr/lpp/ispf/bin/ISPZINO</td>
<td>A stub file with the sticky bit set on to enable invocation of the Legacy IPSF Gateway load module ISPZINO. (2)</td>
</tr>
<tr>
<td>/usr/lpp/ispf/bin/ISPZINT</td>
<td>A stub file with the sticky bit set on to enable invocation of the gateway initialization load module ISPZINT. (1,2)</td>
</tr>
<tr>
<td>/usr/lpp/ispf/bin/ISPZTSO</td>
<td>A stub file with the sticky bit set on to enable invocation of the TSO initialization load module ISPZTSO. (2)</td>
</tr>
<tr>
<td>/usr/lpp/ispf/bin/ISPZXENV</td>
<td>A REXX routine called by the gateway XML API routine, ISPZXXML, to set up environment variables. (1,2)</td>
</tr>
<tr>
<td>/usr/lpp/ispf/bin/ISPZXXML</td>
<td>A REXX routine that is the XML API for the gateway. (1,2)</td>
</tr>
</tbody>
</table>

1 – File is used by the Interactive ISPF Gateway.
2 – File is used by the Legacy ISPF Gateway.

If you are using the Interactive ISPF Gateway, see “Interactive ISPF Gateway” on page 46 for additional information.

If you are using the Legacy ISPF Gateway, see “Legacy ISPF Gateway” on page 69 for additional information.
After a TSO/E address space is created, it can be used for a single TSO/ISPF command and then ended or it can be kept active and used for subsequent commands. The advantage of keeping the address space active is that the overhead of establishing a new address space for every TSO/ISPF command request is avoided.

The gateway does not provide the network communications and data transport between the client and z/OS host. The gateway is a thin service designed to interface with a local z/OS application that is serving remote users that have been authenticated to z/OS. The local application is responsible for any network communications when serving the remote client connections.

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

### Preparing to use the Interactive ISPF Gateway

Consider these points as you prepare your system to use the Interactive ISPF Gateway:

- An OMVS segment must be defined in the security system (RACF or equivalent) for each user of the gateway. It must specify a valid non-zero uid, home directory, and shell command.
- The prerequisites described in *System prerequisites for the CEA TSO/E address space services* in *z/OS MVS Programming: Callable Services for High-Level Languages* must be completed.
- 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,O
  SETOMVS MAXPROCUSER=50.
  ```

  Setting a value that is too low can cause the gateway to fail.
- Table 4 on page 45 describes the load modules that comprise the gateway. The load modules that are used by the Interactive ISPF Gateway are indicated by the footnotes. 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)
  ```

  For information on using this command, refer to your security server's documentation. Failure to define ISPZINT as being controlled might result in messages, such as CSV042I or ICH422I, being issued.
- Table 5 on page 45 describes the gateway files installed into the z/OS UNIX file system. The files that are used by the Interactive ISPF Gateway are indicated by the footnotes.
- The CEA TSO/E address space services programs are in directory /usr/lib. This directory must be included in the LIBPATH environment variable available to the gateway.
- If a user has a TN3270 ISPF session active while using the Interactive ISPF Gateway, the TN3270 session must use ISPF profile sharing when it uses the same ISPF profile data set as is used by the logon procedure specified for the gateway. Profile sharing is enabled by specifying the SHRPROF parameter on the ISPF, PDF, or ISPSTART command used to start ISPF. Sites can also use the ISPF configuration table to define SHRPROF as a default parameter for these commands. An alternative to using ISPF profile sharing is to use different ISPF profile data sets for TN3270 ISPF sessions and the gateway. This can be done by using different logon procedures for TN3270 and the gateway.
- For additional steps that must be completed depending on the environment from which you call the gateway, see “Customizing the gateway calling environment” on page 82.

### Using the Interactive ISPF Gateway

This section describes how to provide requests to the Interactive ISPF Gateway and the format of the output returned by the gateway.
Using the Interactive ISPF Gateway XML API

Requests can be passed to the Interactive ISPF Gateway in XML format. The client passes the request XML, by means of STDIN, to the XML API module ISPZXML. After processing the input XML, ISPZXML calls module ISPZXENV to define the necessary environment variables. After defining the environment variables, ISPZXENV calls module ISPZINT to initialize the gateway. If environment variable CGI_CEATSO is defined and is set to the value TRUE, ISPZINT calls module ISPZINL to invoke the Interactive ISPF Gateway to process the request. The output from the request is passed by means of STDOUT back to the client.

The XML schema shown here, which is supplied in member ISPZXCEI in the ISPF samples data set ISP.SISPSAMP, describes the request XML format and can be used to validate the XML for a request:

```xml
<?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>
            <!-- REQUEST: Request type. Required on all gateway calls -->
            <xs:element name="request" minOccurs="1">
              <xs:simpleType>
                <xs:restriction base="xs:string">
                  <!-- NEWTSO: Start a new TSO session without ISPF active. -->
                  <xs:enumeration value="NEWTSO"/>
                  <!-- NEWTSOISPF: Start a new TSO session with ISPF active. -->
                  <xs:enumeration value="NEWTSOISPF"/>
                  <!-- RECONNTSO: If a dormant TSO session is available, reconnect to it without ISPF active. Otherwise, this is equivalent to NEWTSO. -->
                  <xs:enumeration value="RECONNTSO"/>
                  <!-- RECONNTSOISPF: If a dormant TSO session is available, reconnect to it with ISPF active. Otherwise, this is equivalent to NEWTSOISPF. -->
                  <xs:enumeration value="RECONNTSOISPF"/>
                  <!-- REUSE: Send command to a started session. -->
                  <xs:enumeration value="REUSE"/>
                  <!-- RESPOND: Respond to a TSO prompt. -->
                  <xs:enumeration value="RESPOND"/>
                  <!-- PING: Ping a started session. -->
                  <xs:enumeration value="PING"/>
                  <!-- ATTN: ATTN a started session. -->
                  <xs:enumeration value="ATTN"/>
                  <!-- DORMANT: If dormant sessions are enabled and max dormant sessions is not reached, put the TSO session in a dormant state. Otherwise, this is equivalent to LOGOFF. -->
                  <xs:enumeration value="DORMANT"/>
                  <!-- LOGOFF: LOGOFF the started TSO session. -->
                  <xs:enumeration value="LOGOFF"/>
                  <!-- CANCEL: CANCEL the started TSO session. -->
                  <xs:enumeration value="CANCEL"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:element>
            <!-- PROCNAME: Procedure name. Required with request of NEWTSO | NEWTSOISPF | RECONNTSO | RECONNTSOISPF -->
            <xs:element name="procname" type="xs:string" minOccurs="0"/>
            <!-- ACCTNUM: Account number. Required with request of NEWTSO | NEWTSOISPF | RECONNTSO | RECONNTSOISPF -->
            <xs:element name="acctnum" type="xs:string" minOccurs="0"/>
            <!-- GROUPID: Group identifier. Required with request of NEWTSO | NEWTSOISPF | RECONNTSO | RECONNTSOISPF -->
            <xs:element name="groupid" type="xs:string" minOccurs="0"/>
            <!-- REGIONSZ: Region size. Required with request of NEWTSO | NEWTSOISPF | RECONNTSO | RECONNTSOISPF -->
            <xs:element name="regionsz" type="xs:integer" minOccurs="0"/>
            <!-- SESSID: Session identifier. Required with request of -->
            <xs:element name="sessid" type="xs:string" minOccurs="0"/>
          </xs:all>
        </xs:complexType>
      </xs:element>
    </xs:all>
  </xs:complexType>
</xs:element>
</xs:schema>
```
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 ISPZXCEO 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
<?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:all>
    </xs:complexType>
  </xs:element>
  <xs:element name="OPERATIONS-LOG"/>
</xs:schema>
```

**Using the Interactive ISPF Gateway native API**

Requests can be passed to the Interactive ISPF Gateway in native format. Native API requests consist of a parameter string passed by means of STDIN to the gateway module ISPZXENV. ISPZXENV defines the necessary environment variables and then calls module ISPZINT to initialize the gateway. If environment variable CGI_CEATSO is defined and is set to the value TRUE, ISPZINT calls module ISPZINL to invoke the Interactive ISPF Gateway to process the request. The output from a request is passed by means of STDOUT back to the client.

**Note:** If the client defines environment variables using a method other than by calling gateway module ISPZXENV, the native API request can be passed by means of STDIN directly to gateway module ISPZINT.
Native API requests are of the format:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cmdresp</td>
<td>A TSO or ISPF command to be issued in a TSO/E address space. The command format is free form. A response must be in the format: RESPONSE &quot;response text&quot;. When used, this parameter must be specified as the first parameter in the request.</td>
</tr>
<tr>
<td>loglevel</td>
<td>The level of logging performed by the gateway during processing of the request. Logging information is included in the information returned to the client by means of STDOUT. Specify this parameter using the format: &amp;LOGLEVEL=log_level. The logging levels are: 0 No logging. This is the default level. 1 Log error information. Provides additional information when errors occur during request processing. 2 Log debug information. Provides information about the request processing that can be useful when debugging problems. 4 Log communication information. Provides information about the messages that are sent by the gateway to TSO or ISPF and received by the gateway from TSO or ISPF. 8 Log time information. Provides information about the elapsed time during various phases of gateway processing. 16 Log information to the system console. Causes the logging information generated by the gateway to be written to the system console. 32 Include time stamps in the log information. Causes a time stamp value to be included in some lines of log information.</td>
</tr>
</tbody>
</table>
Notes:

• The logging level values can be added together to request multiple types of logging information. For example, specify &LOGLEVEL=7 to request the logging of error, debug, and communication information.

• This parameter should be specified immediately following the cmdresp parameter, or first when the cmdresp parameter is not specified, in order to provide the most comprehensive logging information.

request
This parameter specifies an action to be taken by the gateway.
Specify this parameter using the format: &REQUEST=action
The possible values for action are:

NEWTSO
Start a new TSO/E address space. Do not reconnect to a dormant address space. Do not start ISPF in the address space. This address space can only be used for issuing TSO commands. The procname, acctnum, groupid, and regionsz parameters must be specified with this request value.

NEWTSOISPF
Start a new TSO/E address space. Do not reconnect to a dormant address space. Start ISPF in the address space. This address space can be used for issuing TSO and ISPF commands. The procname, acctnum, groupid, and regionsz parameters must be specified with this request value.

RECONNTSO
Reconnect to a dormant TSO/E address space if one is available. Otherwise, start a new address space. Do not start ISPF in the address space. This address space can only be used for issuing TSO commands. The procname, acctnum, groupid, and regionsz parameters must be specified with this request value.

RECONNTSOISPF
Reconnect to a dormant TSO/E address space if one is available. Otherwise, start a new address space. Start ISPF in the address space. This address space can be used for issuing TSO and ISPF commands. The procname, acctnum, groupid, and regionsz parameters must be specified with this request value.

Note: ISPF cannot be active in a dormant address space. Therefore, you cannot reconnect to a dormant address space in which ISPF is already active.

REUSE
Reuse a TSO/E address space for a new command. The address space to reuse is identified by the sessid parameter. The sessid parameter must be specified with this request value.

RESPOND
Respond to a prompt from a TSO/E address space. The target address space for the response is identified by the sessid parameter. The sessid and cmdresp parameters must be specified with this request value.

PING
Ping the TSO/E address space that is started or reconnected by the same client request or that is identified by the sessid parameter. The sessid parameter must be specified with this request value unless the address space is started or reconnected by the same client request. If 15 minutes pass without receipt of a ping for the address space, CEA TSO/E address space services ends the address space.

ATTN
Send an attention interrupt to the TSO/E address space that is identified by the sessid parameter. The sessid parameter must be specified with this request value.

DORMANT
Put the TSO/E address space in a dormant state. This request is satisfied only if dormant sessions are enabled and the maximum number of dormant sessions is not reached. Otherwise, this is equivalent to LOGOFF. The TSO/E address space is the address space that was started or reconnected by the same client request or that is identified by the sessid parameter. The sessid
parameter must be specified with this request value unless the address space is started or reconnected by the same client request.

LOGOFF
Log off the TSO/E address space that is started or reconnected by the same client request or that is identified by the sessid parameter. The sessid parameter must be specified with this request value unless the address space is started or reconnected by the same client request.

CANCEL
Cancel the TSO/E address space that is started or reconnected by the same client request or that is identified by the sessid parameter. The sessid parameter must be specified with this request value unless the address space is started or reconnected by the same client request.

**procname**
The name of the TSO/E logon procedure to be used for a TSO/E session. This parameter is required on a request for the start of a new TSO/E session (NEWTSO or NEWTSOISPF) or a reconnect to a dormant TSO/E session (RECONNTSO or RECONNTSOISPF). Callers can obtain this information by using z/OS Security Manager services, such as RACROUTE REQUEST=EXTRACT. Valid values are 1 to 8 characters long.

Specify this parameter using the format: &PROCNAME=procedure_name

**acctnum**
The TSO/E account number to be used for a TSO/E session. This parameter is required on a request for the start of a new TSO/E session (NEWTSO or NEWTSOISPF) or a reconnect to a dormant TSO/E session (RECONNTSO or RECONNTSOISPF). Callers can obtain this information by using z/OS Security Manager services, such as RACROUTE REQUEST=EXTRACT. Valid values are 1 to 40 characters long.

Specify this parameter using the format: &ACCTNUM=account_number

**groupid**
The TSO/E group name to be used for a TSO/E session. This parameter is required on a request for the start of a new TSO/E session (NEWTSO or NEWTSOISPF) or a reconnect to a dormant TSO/E session (RECONNTSO or RECONNTSOISPF). Callers can obtain this information by using z/OS Security Manager services, such as RACROUTE REQUEST=EXTRACT. Valid values are 1 to 8 characters long.

Specify this parameter using the format: &GROUPID=group_id

**regionsz**
The region size to be used for a TSO/E session. This parameter is required on a request for the start of a new TSO/E session (NEWTSO or NEWTSOISPF) or a reconnect to a dormant TSO/E session (RECONNTSO or RECONNTSOISPF). Callers can obtain this information by using z/OS Security Manager services, such as RACROUTE REQUEST=EXTRACT. Valid values are 0 to 2,096,128.

Specify this parameter using the format: &REGIONSZ=region_size

**sessid**
The information that uniquely identifies a started TSO/E session. This information is returned to the client by the gateway when a TSO/E session is started or reconnected and consists of the following:

- Address space information version
- Address space ID
- Address space token
- Address space index
- Message queue ID
- Address space type

Specify this parameter using the format:

&VER=ver&ASID=asid&STOKEN=stoken&INDEX=index&MSGQID=msgqid&TYPE=type

**Note:** This is the format used by the gateway to return the session identifying information to the client when the session is started or reconnected. The information is returned by the gateway between the <SESS> and </SESS> tags. The client can provide the information on subsequent requests exactly as it was returned by the gateway.
With the exception described in the following note, the sessid parameter is required with any request parameter value where the gateway must identify the TSO/E session to be used. This includes all of the following types of requests:

&REQUEST=REUSE
   Reuse the session for a new command.

&REQUEST=RESPOND
   Respond to a TSO prompt from the session.

&REQUEST=PING
   Ping the session.

&REQUEST=ATTN
   Send an attention interrupt to the session.

&REQUEST=DORMANT
   Put the session in a dormant state.

&REQUEST=LOGOFF
   Log off of the session.

&REQUEST=CANCEL
   Cancel the session.

Note: If a single input from the client requests both the start or reconnect of a TSO/E session and one of the request types above, the session identifying information is not required on the request. For information on combining request types, see “Combining requests using the Interactive ISPF Gateway” on page 63.

Starting a TSO/E address space using the Interactive ISPF Gateway

To start a TSO/E address space, send a request to the gateway with the request parameter set to one of these values:

NEWTSO
   Start a new TSO/E address space. Do not reconnect to a dormant address space. Do not start ISPF in the address space. This address space can only be used for issuing TSO commands.

NEWTSOISPF
   Start a new TSO/E address space. Do not reconnect to a dormant address space. Start ISPF in the address space. This address space can be used for issuing TSO and ISPF commands.

RECONNTSO
   Reconnect to a dormant TSO/E address space if one is available. Otherwise, start a new address space. Do not start ISPF in the address space. This address space can only be used for issuing TSO commands.

RECONNTSOISPF
   Reconnect to a dormant TSO/E address space if one is available. Otherwise, start a new address space. Start ISPF in the address space. This address space can be used for issuing TSO and ISPF commands.

Note: Dormant TSO/E address spaces and the ability to reconnect to them are functions provided by CEA TSO/E address space services. To enable this feature, which is disabled by default, specify non-zero values for the RECONSESSIONS and RECONTIME statements in the TSOASMGR parmib statement in the CEAPRMxx parmib member. For additional information about dormant sessions and enabling their use, refer to Reconnecting to CEA TSO/E address spaces in z/OS MVS Programming: Callable Services for High-Level Languages.

Four items must be provided with a request to start (or reconnect to) a TSO/E address space. These are the same items that are entered when a user logs on to a TSO/E session:

• Procedure name – specified with the procname parameter.
• Account number – specified with the acctnum parameter.
• Group name – specified with the groupid parameter.
• Region size – specified with the regionsz parameter.

Callers can obtain this information by using z/OS Security Manager services, such as RACROUTE REQUEST=EXTRACT.

**Using the XML API to start a TSO/E address space**

The following example shows the input and resulting output using the XML API to start a TSO/E address space. In this example, a new TSO/E address space is requested. ISPF is not started in the address space. Therefore, the address space can only be used for issuing TSO commands:

**Input:**

```xml
<?xml version="1.0"?>
<ISPF-INPUT>
xmns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="ispf.xsd">
  <SERVICE-REQUEST>
    <request>NEWTSO</request>
    <procname>ISPFPROC</procname>
    <acctnum>SAMPACCT</acctnum>
    <groupid>DEFAULT</groupid>
    <regionsz>2000000</regionsz>
  </SERVICE-REQUEST>
</ISPF-INPUT>
```

**Output:**

```xml
<?xml version="1.0"?>
<ISPF-OUTPUT>
  <SERVICE-REQUEST>
    <request>NEWTSO</request>
    <procname>ISPFPROC</procname>
    <acctnum>SAMPACCT</acctnum>
    <groupid>DEFAULT</groupid>
    <regionsz>2000000</regionsz>
  </SERVICE-REQUEST>
  <SERVICE-RESPONSE>
    <RETURN-CODE>0</RETURN-CODE>
    <SESSION-INFO>
      <SESS>&VER=1&ASID=88&STOKEN=000001600000006C&INDEX=1&MSGQID=3342342&TYPE=TSO</SESS>
      <TSOPROMPT>NO</TSOPROMPT>
    </SESSION-INFO>
  </SERVICE-RESPONSE>
  <OPERATIONS-LOG>
    <![CDATA[
      Content-type: text/plain
      ISPZINL started - z/OS V2R3 01JAN17 Base
      Data read from STDIN is
      &REQUEST=NEWTSO&PROCNAME=ISPFPROC&ACCTNUM=SAMPACCT&GROUPID=DEFAULT&REGIONSZ=2000000
      *** XML-NOTE *** Reference tagged SERVICE-RESPONSE
    ]]>}
  </OPERATIONS-LOG>
</ISPF-OUTPUT>
```

**Note:** The value returned between the `<SESS>` and `</SESS>` tags is the sessid value that must be returned on subsequent requests to use the session.

**Using the native API to start a TSO/E address space**

The following example shows the input and resulting output using the native API to start a TSO/E address space. In this example, a new TSO/E address space is requested. ISPF is started in the address space. Therefore, it can be used for issuing TSO and ISPF commands:

**Input:**

```text
&REQUEST=NEWTSOISPF &PROCNAME=ISPFPROC &ACCTNUM=SAMPACCT &GROUPID=DEFAULT &REGIONSZ=2000000
```

**Output:**

```text
ISPF started - z/OS V2R3 01JAN17 Base
Data read from STDIN is
&REQUEST=NEWTSO&PROCNAME=ISPFPROC&ACCTNUM=SAMPACCT&GROUPID=DEFAULT&REGIONSZ=2000000
*** XML-NOTE *** Reference tagged SERVICE-RESPONSE
```
Note: The value returned between the <SESS> and </SESS> tags is the sessid value that must be returned on subsequent requests to use the session.

Issuing a TSO/IPSF command using the Interactive ISPF Gateway

To issue a TSO or ISPF command in a TSO/E address space, send a request to the gateway with the cmdresp parameter set to the command string.

You must also include the request parameter to indicate whether you want the command to be issued in a new TSO/E address space, in a reconnected TSO/E address space, or in an active TSO/E address space. The request parameter must be set to one of the following values:

NEWTSO
The command is issued in a new TSO/E address space in which ISPF is not started. Only use this value if your command does not require that ISPF be started in the address space. For this value, the procname, acctnum, groupid, and regionsz parameters must also be specified.

NEWTSOISPF
The command is issued in a new TSO/E address space in which ISPF is started. Use this value if your command requires that ISPF be started in the address space. For this value, the procname, acctnum, groupid, and regionsz parameters must also be specified.

RECONNTSO
If a dormant TSO/E address space is available, the address space is reconnected and the command is issued in that address space. Otherwise, the command is issued in a new TSO/E address space. ISPF is not started in the address space. Only use this value if your command does not require that ISPF be started in the address space. For this value, the procname, acctnum, groupid, and regionsz parameters must also be specified.

RECONNTSOISPF
If a dormant TSO/E address space is available, the address space is reconnected and the command is issued in that address space. Otherwise, the command is issued in a new TSO/E address space. ISPF is started in the address space. Use this value if your command requires that ISPF be started in the address space. For this value, the procname, acctnum, groupid, and regionsz parameters must also be specified.

REUSE
The command is issued in an active TSO/E address space. You must be aware of whether ISPF is started in the address space, based upon the request that was previously issued to start or reconnect to the address space. For this value, the sessid parameter must also be specified.

Using the XML API to issue a TSO/IPSF command in a TSO/E address space

The following example shows the input and resulting output using the XML API to issue a TSO/IPSF command in a TSO/E address space. In this example, the TIME command is issued in a new TSO/E address space in which ISPF is not started.

Input:

<?xml version="1.0"?>
Using the native API to issue a TSO/IPSF command in a TSO/E address space

The following example shows the input and resulting output using the native API to issue a TSO/ISPF command in a TSO/E address space. In this example, the TSO TIME command is issued in an already active TSO/E address space in which ISPF is started:

Input:

TSO TIME &REQUEST=REUSE&VER=1&ASID=106&STOKEN=000001A80000004E&INDEX=2&MSGQID=9764869&TYPE=ISPF

Output:

Content-type: text/plain
ISPZINL started - z/OS V2R3 01JAN17 Base
Data read from STDIN is
TSO TIME &REQUEST=REUSE&VER=1&ASID=106&STOKEN=000001A80000004E&INDEX=2&MSGQID=9764869&TYPE=ISPF
*** OUTPUT FROM TSO SESSION ***
*** XML-NOTE *** Reference tagged SERVICE-RESPONSE

56  z/OS: ISPF Planning and Customizing
**Interactive communication using the Interactive ISPF Gateway**

When the client requests that a command be issued in a TSO/E address space and that command prompts the client for input, the gateway processing of the request that issued the command ends. Any data that is written by the command before it issued the prompt is returned to the client in the gateway output, along with an indication that the command is waiting for input. The indication that the command is waiting for input is a TSOPROMPT line with the value YES:

```xml
<TSOPROMPT>YES</TSOPROMPT>
```

When included in the output, the TSOPROMPT line can be found in the information that is surrounded by the `<SESSION-INFO>` and `</SESSION-INFO>` tags. If the TSOPROMPT line is not included in the output or the value on the line is NO, the command has ended and is not waiting for input.

To respond to a prompt from the command running in a TSO/E address space, send a request to the gateway with the request parameter set to RESPOND. The target TSO/E address space for the response must be identified using the sessid parameter. Use the cmdresp parameter to provide the response text. The cmdresp parameter must be specified using the format:

```
RESPONSE "response text"
```

**Using the XML API for interactive communication**

The following example shows the input and resulting output using the XML API for interactive communication.

In this example, the client issues the command EX 'TEST.EXEC(GWSVMULT)' to run a REXX exec. The exec writes the line “Hello client, what is your name?” and then prompts for input. This is the output returned to the client when the exec prompts for input:

```xml
<?xml version="1.0"?>
<ISPF-OUTPUT>
  <SERVICE-REQUEST>
    <request>REUSE</request>
    <sessid>&VER=1&ASID=88&STOKEN=0000016000000076&INDEX=1&MSGQID=10944520&TYPE=TSO</sessid>
    <cmdresp>EX 'TEST.EXEC(GWSVMULT)'</cmdresp>
  </SERVICE-REQUEST>
  <SERVICE-RESPONSE>
    <TSO>
      <![CDATA[Hello client, what is your name?]]>
    </TSO>
    <RETURN-CODE>0</RETURN-CODE>
    <SESSION-INFO>
      <TSOPROMPT>YES</TSOPROMPT>
    </SESSION-INFO>
    <OPERATIONS-LOG>
      <![CDATA[Content-type: text/plain
      ISPZINL started - z/OS V2R3 01JAN17 Base
      Data read from STDIN is
      EX 'TEST.EXEC(GWSVMULT)' REQUEST=REUSE
      &VER=1&ASID=88&STOKEN=0000016000000076&INDEX=1&MSGQID=10944520&TYPE=TSO
      *** OUTPUT FROM TSO SESSION ***
      *** XML-NOTE *** Reference tagged SERVICE-RESPONSE
      ]]>]]>
  </OPERATIONS-LOG>
</ISPF-OUTPUT>
```
The following XML input from the client provides the response:

```xml
<?xml version="1.0"?>
<ISPF-INPUT>
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="ispf.xsd">
    <SERVICE-REQUEST>
        <request>RESPOND</request>
        <cmdresp>RESPONSE "JOHN"</cmdresp>
        <sessid>&VER=1&ASID=88&STOKEN=0000016000000076&INDEX=1&MSGQID=10944526&type=TSO</sessid>
    </SERVICE-REQUEST>
</ISPF-INPUT>
```

Output:

```xml
<?xml version="1.0"?>
<ISPF-OUTPUT>
    <SERVICE-REQUEST>
        <request>RESPOND</request>
        <cmdresp>RESPONSE "JOHN"</cmdresp>
        <sessid>&VER=1&ASID=88&STOKEN=0000016000000076&INDEX=1&MSGQID=10944526&type=TSO</sessid>
    </SERVICE-REQUEST>
    <SERVICE-RESPONSE>
        <TSO>
            <![CDATA[
                Hello JOHN
            ]]>
        </TSO>
        <RETURN-CODE>0</RETURN-CODE>
        <SESSION-INFO>
            <TSOPROMPT>NO</TSOPROMPT>
        </SESSION-INFO>
    </SERVICE-RESPONSE>
    <OPERATIONS-LOG>
        <![CDATA[
            Content-type: text/plain
            ISPZINL started - z/OS V2R3 01JAN17 Base
            Data read from STDIN is
            RESPONSE "JOHN"&REQUEST=RESPOND
            &VER=1&ASID=88&STOKEN=0000016000000076&INDEX=1&MSGQID=10944526&type=TSO
            *** XML-NOTE *** Reference tagged SERVICE-RESPONSE
        ]]>"
    </OPERATIONS-LOG>
</ISPF-OUTPUT>
```

As indicated in the output by the TSOPROMPT line with the value NO, the command did not prompt the client a second time. However, if the client is prompted a second time, the TSOPROMPT line would have the value YES and the example above could be repeated to provide the new response.

**Using the native API for interactive communication**

The following example shows the input and resulting output using the native API for interactive communication.

In this example, the client issues the command TSO PING to run the PING program. The program writes the line “Enter host name or address” and prompts for input. The following output is returned to the client when the program prompts for input:

```xml
<?xml version="1.0"?>
<ISPF-OUTPUT>
    <SERVICE-REQUEST>
        <request>RESPOND</request>
        <cmdresp>RESPONSE "JOHN"</cmdresp>
        <sessid>&VER=1&ASID=104&STOKEN=000001A00000019B&INDEX=1&MSGQID=10616837&type=ISPF
            &REQUEST=REUSE
        </sessid>
    </SERVICE-REQUEST>
    <SERVICE-RESPONSE>
        <ISPF>
            <![CDATA[
                Enter host name or address
            ]]>"
        </ISPF>
    </SERVICE-RESPONSE>
    <OPERATIONS-LOG>
        <![CDATA[
            Content-type: text/plain
            ISPZINL started - z/OS V2R3 01JAN17 Base
            Data read from STDIN is
            TSO PING &REQUEST=REUSE
            &VER=1&ASID=104&STOKEN=000001A00000019B&INDEX=1&MSGQID=10616837&type=ISPF
            *** OUTPUT FROM ISPF SESSION ***
            *** XML-NOTE *** Reference tagged SERVICE-RESPONSE
            Command passed to SELECT -
            CMD(PING)
            <![CDATA[
                Enter host name or address
            ]]>"
        ]]>"
    </OPERATIONS-LOG>
</ISPF-OUTPUT>
```
The following native input from the client provides the response:

```
RESPONSE "TEST-GWY.US.XYZ.COM" &REQUEST=RESPOND &VER=1&ASID=104&STOKEN=000001A00000019B&INDEX=1&MSGQID=10616837&TYPE=ISPF
```

Output:

```
Content-type: text/plain
ISPZINL started - z/OS V2R3 01JAN17 Base
Data read from STDIN is RESPONSE "TEST-GWY.US.XYZ.COM" &REQUEST=RESPOND &VER=1&ASID=104&STOKEN=000001A00000019B&INDEX=1&MSGQID=10616837&TYPE=ISPF
</ISPINFO>
CS V2R2: Pinging host TEST-GWY.US.XYZ.COM (192.0.2.61)
Ping #1 response took 0.001 seconds.
</ISPF>
RC=0
</ISPINFO>
</SESSION-INFO>
</TSOPROMPT>NO</SESSION-INFO>
</SESSION-INFO>

As indicated in the output by the TSOPROMPT line with the value NO, the program did not prompt the client a second time. If the client is prompted a second time, the TSOPROMPT line would have the value YES and the example above could be repeated to provide the new response.

**Pinging a TSO/E address space using the Interactive ISPF Gateway**

To ping a TSO/E address space, send a request to the gateway with the request parameter set to PING. The session to be pinged must be identified using the sessid parameter.

**Note:** If 15 minutes pass without receipt of a ping for the address space, CEA TSO/E address space services ends the address space.

**Using the XML API to ping a TSO/E address space**

The following example shows the input and resulting output using the XML API to ping a TSO/E address space:

**Input:**

```
<?xml version="1.0"?>
<ISPF-INPUT>
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:noNamespaceSchemaLocation="ispf.xsd">
 <SERVICE-REQUEST>
  <request>PING</request>
  <sessid>&VER=1&ASID=88&STOKEN=000001600000006C&INDEX=1&MSGQID=3342342&TYPE=TSO</sessid>
 </SERVICE-REQUEST>
</ISPF-INPUT>
```

**Output:**

```
<?xml version="1.0"?>
<ISPF-OUTPUT>
 <SERVICE-REQUEST>
  <request>PING</request>
  <sessid>&VER=1&ASID=88&STOKEN=000001600000006C&INDEX=1&MSGQID=3342342&TYPE=TSO</sessid>
 </SERVICE-REQUEST>
 <SERVICE-RESPONSE>
```

Customizing ISPF 59
Using the native API to ping a TSO/E address space

The following example shows the input and resulting output using the native API to ping a TSO/E address space:

Input:

\&REQUEST=PING \&VER=1\&ASID=106\&STOKEN=000001A80000004E\&INDEX=2\&MSGQID=9764869\&TYPE=ISPF

Output:

Content-type: text/plain

ISPZINL started - z/OS V2R3 01JAN17 Base
Data read from STDIN is \&REQUEST=PING \&VER=1\&ASID=106\&STOKEN=000001A80000004E\&INDEX=2\&MSGQID=9764869\&TYPE=ISPF

**ISRINFO**
RC=0

**SESSION-INFO**
**TSOPROMPT**NO

Sending an attention interrupt using the Interactive ISPF Gateway

To send an attention interrupt to a TSO/E address space, send a request to the gateway with the request parameter set to ATTN. The session must be identified using the sessid parameter.

Using the XML API to send an attention interrupt to a TSO/E address space

The following example shows the input and resulting output using the XML API to send an attention interrupt to a TSO/E address space:

Input:

```xml
<?xml version="1.0"?>
<ISPF-INPUT>
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="ispf.xsd">
<SERVICE-REQUEST>
<request>ATTN</request>
<sessid>&REQUEST=PING &VER=1&ASID=88&STOKEN=000001600000006C&INDEX=1&MSGQID=3342342&TYPE=TSO</sessid>
</SERVICE-REQUEST>
</ISPF-INPUT>
```

Output:

```xml
<?xml version="1.0"?>
<ISPF-OUTPUT>
<SERVICE-REQUEST>
60
z/OS: ISPF Planning and Customizing
```
Using the native API to send an attention interrupt to a TSO/E address space

The following example shows the input and resulting output using the native API to send an attention interrupt to a TSO/E address space:

**Input:**

```
&REQUEST=ATTN &VER=1&ASID=106&STOKEN=000001A80000004E&INDEX=2&MSGQID=9764869&TYPE=ISPF
```

**Output:**

```
Content-type: text/plain
ISPZINL started - z/OS V2R3 01JAN17 Base
Data read from STDIN is
&REQUEST=ATTN &VER=1&ASID=106&STOKEN=000001A80000004E&INDEX=2&MSGQID=9764869&TYPE=ISPF
*** XML-NOTE *** Reference tagged SERVICE-RESPONSE
EZZ3112I Host name or address not entered
RC=100
```

Ending a TSO/E address space using the Interactive ISPF Gateway

To end a TSO/E address space, send a request to the gateway with the request parameter set to one of these values:

**DORMANT**

If dormant sessions are enabled and the maximum number of dormant sessions is not reached, put the TSO/E session in a dormant state. Otherwise, this is equivalent to LOGOFF. The TSO/E address space is the address space that is started or reconnected by this request or that is identified by the **sessid** parameter.

**LOGOFF**

Log off the TSO/E address space that is started or reconnected by this request or that is identified by the **sessid** parameter.

**CANCEL**

Cancel the TSO/E address space that is started or reconnected by this request or that is identified by the **sessid** parameter.
**Note:** Dormant TSO/E address spaces and the ability to reconnect to them are functions provided by CEA TSO/E address space services. To enable this feature, which is disabled by default, specify non-zero values for the RECONSESSIONS and RECONTIME statements in the TSOASMGR parmlib statement of the CEAPRMxx parmlib member. For additional information about dormant sessions and enabling their use, refer to [Reconnecting to CEA TSO/E address spaces](z/OS MVS Programming: Callable Services for High-Level Languages).

**Using the XML API to end a TSO/E address space**

The following example shows the input and resulting output using the XML API to end a TSO/E address space. In this example, the TSO/E address space, which is started by a previous gateway request, is put in a dormant state if dormant sessions are enabled and the maximum number of dormant sessions is not reached. Otherwise, the session is logged off:

**Input:**

```xml
<?xml version="1.0"?>
<ISPF-INPUT
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="ispf.xsd">  
<SERVICE-REQUEST>
<request>DORMANT</request>
</SERVICE-REQUEST>
</ISPF-INPUT>
```

**Output:**

```xml
<?xml version="1.0"?>
<ISPF-OUTPUT>
<SERVICE-REQUEST>
<request>DORMANT</request>
</SERVICE-REQUEST>
</ISPF-OUTPUT>
```

**Using the native API to end a TSO/E address space**

The following example shows the input and resulting output using the native API to end a TSO/E address space. In this example, the TSO/E address space, which is started by a previous gateway request, is logged off:

**Input:**

```plaintext
&REQUEST=LOGOFF &VER=1&ASID=106&STOKEN=000001A80000004E&INDEX=2&MSGQID=9764869&TYPE=ISPF
```

**Output:**

```plaintext
Content-type: text/plain
ISPZINL started - z/OS V2R3 01JAN17 Base
Data read from STDIN is
&REQUEST=DORMANT&VER=1&ASID=88&STOKEN=0000016000000006C&INDEX=1&MSGQID=3342342&TYPE=TSO
*** XML-NOTE *** Reference tagged SERVICE-RESPONSE
```
Combining requests using the Interactive ISPF Gateway

Some types of requests can be combined into a single gateway input. For example, the client can send in
to the gateway a single input that requests all of the following:

• Start a TSO/E address space
• Issue a command in the address space
• End the address space.

Combining request types in a single input has the following restrictions:

• Only one of NEWTSO, NEWTSOISPF, RECONNTSO, RECONNTSOISPF, and REUSE can be
  specified in a
• Only one of CANCEL, LOGOFF, and DORMANT can be
  specified in a single input
• RESPOND cannot be combined with any other request type
• ATTN cannot be combined with any other request type.

Using the XML API to combine requests

The following example shows the input and resulting output using the XML API to combine requests. This
example combines two requests:

• Issue the TIME command by reusing the active TSO/E address space identified by the sessid parameter
• Ping the TSO/E address space to keep it active.

Input:

```xml
<?xml version="1.0"?>
<ISPF-INPUT>
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="ispf.xsd">
  <SERVICE-REQUEST>
    <request>REUSE</request>
    <sessid>&VER=1&ASID=88&STOKEN=0000016000000077&INDEX=2&MSGQID=10682373&TYPE=TSO</sessid>
    <request>PING</request>
    <cmdresp>TIME</cmdresp>
  </SERVICE-REQUEST>
</ISPF-INPUT>
```

Output:

```xml
<?xml version="1.0"?>
<ISPF-OUTPUT>
  <SERVICE-REQUEST>
    <request>REUSE</request>
    <sessid>&VER=1&ASID=88&STOKEN=0000016000000077&INDEX=2&MSGQID=10682373&TYPE=TSO</sessid>
    <request>PING</request>
    <cmdresp>TIME</cmdresp>
  </SERVICE-REQUEST>
  <SERVICE-RESPONSE>
    <TSO><![CDATA[
      IKJ56650I TIME-10:01:24 PM. CPU-00:00:00 SERVICE-326 SESSION-00:00:54 JULY 3, 2014
    ]]]>
  </SERVICE-RESPONSE>
  <RETURN-CODE>0</RETURN-CODE>
</ISPF-OUTPUT>
```

```text
Content-type: text/plain
```
Using the native API to combine requests

The following example shows the input and resulting output using the native API to combine requests.

This example combines two requests:

• Issue the TSO LISTCAT command in a new TSO/E address space in which ISPF is started
• After completing the command, logoff the address space.

Input:

```
TSO LISTCAT('SYS1.LINKLIB') &REQUEST=NEWTSO&PROCNAME=ISPFPROC &ACCTNUM=SAMPACCT &GROUPID=DEFAULT &REGIONSZ=2000000 &REQUEST=LOGOFF
```

Output:

```
Content-type: text/plain

ISPZINL started - z/OS V2R3 01JAN17 Base
Data read from STDIN is TSO LISTCAT('SYS1.LINKLIB') &REQUEST=NEWTSO&PROCNAME=ISPFPROC &ACCTNUM=SAMPACCT &GROUPID=DEFAULT &REGIONSZ=2000000 &REQUEST=LOGOFF
*** OUTPUT FROM ISPF SESSION ***
<ISPINFO>
Command passed to SELECT -
CMD(LISTCAT('SYS1.LINKLIB'))
<ISP>
NONVSAM ------- SYS1.LINKLIB
   IN-CAT --- CATALOG.MASTER.SYSPLEXA
</ISP>
RC=0
</ISPINFO>
```

Controlling the Interactive ISPF Gateway logging level

To control the level of logging performed by the gateway during the processing of a request, include the loglevel parameter on the request. For a description of the logging levels provided by the gateway, see “Using the Interactive ISPF Gateway native API” on page 49.

Using the XML API to control the gateway logging level

The following example shows the input and resulting output using the XML API to control the gateway logging level. In this example, debug log information is requested:

Input:

```
```
Using the native API to control the gateway logging level

The following example shows the input and resulting output using the native API to control the gateway logging level. In this example, debug and communication log information is requested:

Input:

```plaintext
TSO TIME &LOGLEVEL=6 &REQUEST=REUSE &VER=1&ASID=97&STOKEN=00000184000000046&INDEX=1&MSGQID=3801094&TYPE=ISPF
```

Output:

```xml
<?xml version="1.0"?>
<ISPF-OUTPUT>
  <SERVICE-REQUEST>
    <loglevel>2</loglevel>
    <request>NEWTSO</request>
    <procname>ISPFPROC</procname>
    <acctnum>SAMPACCT</acctnum>
    <groupid>DEFAULT</groupid>
    <regionsz>2000000</regionsz>
  </SERVICE-REQUEST>
  <SERVICE-RESPONSE>
    <RETURN-CODE>0</RETURN-CODE>
    <SESSION-INFO>
      <SESS>&VER=1&ASID=106&STOKEN=000001A8000000052&INDEX=1&MSGQID=10223621&TYPE=TSO</SESS>
      <TSOPROMPT>No</TSOPROMPT>
    </SESSION-INFO>
  </SERVICE-RESPONSE>
  <OPERATIONS-LOG>
    <![CDATA[
      Content-type: text/plain
      ISPZINL started - z/OS V2R3 01JAN17 Base
      Data read from STDIN is
      &LOGLEVEL=2&REQUEST=NEWTSO&PROCNAME=ISPFPROC&ACCTNUM=SAMPACCT&GROUPID=DEFAULT&REGIONSZ=2000000
    ISPZTrc2 Received LogLevel  = 2
    ISPZTrc2 - Client requested log level: DEBUG
    ISPZTrc2 Received REQUEST = NEWTSO
    ISPZTrc2 - Client requested start of new TSO session without ISPF active
    ISPZTrc2 Received PROCNAME = ISPFPROC
    ISPZTrc2 Received ACCTNUM = SAMPACCT
    ISPZTrc2 Received GROUPID = DEFAULT
    ISPZTrc2 Received REGIONSZ = 2000000
    ISPZTrc2 No command received
    ISPZTrc2 *****************************************
    ISPZTrc2 Calling CEATsoRequest() with CeaTsoStart
    ISPZTrc2 *****************************************
    ISPZTrc2 Connected to a new TSO/E session
    ISPZTrc2 CeaTsorequest Asid       = 106
    ISPZTrc2 CeaTsorequest Stoken     = 000001A8000000052
    ISPZTrc2 CeaTsorequest Index      = 1
    ISPZTrc2 CeaTsorequest Msgqueueid = 10223621
    ISPZTrc2 ***** START OF TSO LOGON MESSAGES *****
    ISPZTrc2 IKJ56455I USER1 LOGON IN PROGRESS AT 00:25:34 ON JULY 17, 2015
    ISPZTrc2 IKJ56951I NO BROADCAST MESSAGES
    ISPZTrc2 **** END OF TSO LOGON MESSAGES *****
    *** XML-NOTE *** Reference tagged SERVICE-RESPONSE
    ]]>}
  </OPERATIONS-LOG>
</ISPF-OUTPUT>
```
IPSPZIn started - z/OS V2R3 01JAN17 Base
Data read from STDOIN is
TSO TIME &LOGLEVEL=6 &REQUEST=REUSE
&VER=1&ASID=97&STOKEN=0000018400000046&INDEX=26&MSGQID=3801094&TYP=ISPF
ISPZTrc2 Received LogLevel = 6
Request = REUSE
Client requested log level: DEBUG COMM
ISPZTrc2 - Client requested
re-use of existing session
ISPZTrc2 Received VER = 1
ISPZTrc2 Received ASID = 97
ISPZTrc2 Received STOKEN = "0000018400000046"
ISPZTrc2 Received INDEX = 2
ISPZTrc2 Received MSGQID = 3801094
ISPZTrc2 - ISPF session
specified
ISPZTrc2 Received command = TSO
ISPZTrc2 Sending message (type=32771): TSO
ISPZTrc4 Sent Message Type: TIME
ISPZTrc4 Sent Message Length: 9
ISPZTrc4 Sent Message: TSO

*** OUTPUT FROM ISPF SESSION

ISPZTrc4 Received Message Type: TIME
ISPZTrc4 Received Message Length: 122
ISPZTrc4 Received Message:

{"VERSION":"0100","DATA":"Entering service call processing 01:35:00.27 "} Entering service call processing 01:35:00.27

ISPZTrc4 Received Message Type: TIME
ISPZTrc4 Received Message Length: 85
ISPZTrc4 Received Message:

{"VERSION":"0100","DATA":"Entering ISPZCMD: TSO/ISPF session initiated 2015071-BASE z22 15/07/17 01:35 "} Entering ISPZCMD: TSO/ISPF session initiated 2015071-BASE z22 15/07/17 01:35
ISPZTrc4 Received Message Type: 2
ISPZTrc4 Received Message Length: 89
ISPZTrc4 Received Message Text:

{"VERSION":"0100","DATA":"Entering service call processing 01:35:00.27 "} Entering service call processing 01:35:00.27

ISPZTrc4 Received Message Type: TIME
ISPZTrc4 Received Message Length: 9
ISPZTrc4 Sent Message: TSO

Session-00:00:41 July 17, 2015
TIME-01:35:41 AM. CPU-00:00:00 SERVICE-3671

ISPZTrc4 Received Message Type: ®REQUEST  = REUSE
ISPZTrc4 Received LogLevel = 6
ISPZTrc4 - Client requested log level: DEBUG COMM
ISPZTrc4 Received Message Text:

{"VERSION":"0100","DATA":"</ISPF>"} </ISPF>

ISPZTrc4 Received Message Type: TIME
ISPZTrc4 Received Message Length: 50
ISPZTrc4 Received Message:

{"VERSION":"0100","DATA":"<ISPINFO>"} ISPZTrc4 Received Message
ISPZTrc4 Received Message Type: ISPINFO
ISPZTrc4 Received Message Type: TIME
ISPZTrc4 Received Message Length: 53
ISPZTrc4 Received Message:

{"VERSION":"0100","DATA":"<ISPINFO>"} ISPZTrc4 Received Message
ISPZTrc4 Received Message Type: ISPINFO
ISPZTrc4 Received Message Type: CMD(TIME)
ISPZTrc4 Received Message Length: 127
ISPZTrc4 Received Message:

{"VERSION":"0100","DATA":"<ISPINFO>"} ISPZTrc4 Received Message
ISPZTrc4 Received Message Type: ISPINFO
ISPZTrc4 Received Message Type: CMD(TIME)
ISPZTrc4 Received Message Length: 50
ISPZTrc4 Received Message:

{"VERSION":"0100","DATA":"<ISPINFO>"} ISPZTrc4 Received Message
ISPZTrc4 Received Message Type: ISPINFO
ISPZTrc4 Received Message Type: CMD(TIME)
ISPZTrc4 Received Message Length: 71
ISPZTrc4 Received Message:

{"VERSION":"0100","DATA":"<ISPINFO>"} ISPZTrc4 Received Message
ISPZTrc4 Received Message Type: ISPINFO
ISPZTrc4 Received Message Type: CMD(TIME)
ISPZTrc4 Received Message Length: 53
ISPZTrc4 Received Message:

{"VERSION":"0100","DATA":"<ISPINFO>"} ISPZTrc4 Received Message
ISPZTrc4 Received Message Type: ISPINFO
ISPZTrc4 Received Message Type: CMD(TIME)
ISPZTrc4 Received Message Length: 49
ISPZTrc4 Received Message:

{"VERSION":"0100","DATA":"<ISPINFO>"} ISPZTrc4 Received Message
ISPZTrc4 Received Message Type: ISPINFO
ISPZTrc4 Received Message Type: CMD(TIME)
Note: Each line of log information that is included in the output is prepended with the string ISPZTrcn, where n is the decimal log level that caused that line to be included in the output. For example, a line that is the output for error logging is prepended with the string ISPZTrc1.

Interactive ISPF Gateway return codes

When an error is detected in the input received by the gateway or an error occurs during the processing of a request, a return code is included in the output returned to the client. In most cases, repeating the command with a loglevel parameter value that includes error logging information provides additional information about the error.

Some errors that occur during the processing of a request are detected by the CeaTsoRequest() function of CEA TSO/E address space services. Information about the return codes, reason codes and diagnostic codes returned for these problems can be found in Return, reason, and diagnostic codes in z/OS MVS Programming: Callable Services for High-Level Languages.

Some errors that occur during the processing of a request are detected by C functions that are called by the gateway. When this occurs, the return code returned by the gateway indicates the failing function. The additional information provided in the error logging information includes the return code and reason code returned by the function. Information about these return codes and reason codes can be found in the section documenting the function in z/OS XL C/C++ Runtime Library Reference.

The following return codes are returned by the ISPF gateway itself:

- 87  
    A TSO logon parameter (for example, procedure name or account number) is not valid.

- 88  
    Conflicting parameters are specified on the gateway request.

- 89  
    A message was received on the message queue with a message type of ISPF. This is not expected.

- 90  
    An error occurred on a call to the iconv_close() function.

- 91  
    An error occurred on a call to the iconv_open() function.

- 92  
    An error occurred on a call to the iconv() function.

- 93  
    An error occurred on a call to the msgsnd() function.

- 94  
    An error occurred on a call to the __msgrcv_timed() function.

- 95  
    An error occurred on a call to the setenv() function.

- 96  
    The response contains no text.
No data was read from STDIN.

A parameter value on the gateway request is bad.

A required parameter is missing on the gateway request.

Receiving return code information using the XML API

The following example shows the return code information received using the XML API. In this example, the procname parameter is not provided on a request to start a new TSO/E address space:

```xml
<?xml version="1.0"?>
<ISPF-OUTPUT>
<SERVICE-REQUEST>
<request>NEWTSO</request>
<acctnum>SAMPACCT</acctnum>
<groupid>DEFAULT</groupid>
<regionsz>2000000</regionsz>
</SERVICE-REQUEST>
<SERVICE-RESPONSE>
<RETURN-CODE>-99</RETURN-CODE>
</SERVICE-RESPONSE>
<OPERATIONS-LOG>
<![CDATA[
Content-type: text/plain
ISPZINL started - z/OS V2R3 01JAN17 Base
Data read from STDIN is &REQUEST=NEWTSO&ACCTNUM=SAMPACCT&GROUPID=DEFAULT
*** XML-NOTE *** Reference tagged SERVICE-RESPONSE
02 *** 'ISPZINL'
     +++ RC(157) +++
]]>
</OPERATIONS-LOG>
</ISPF-OUTPUT>
```

Repeating the command with a loglevel value that includes error information causes the following line to be included in the OPERATIONS_LOG section of the output:

***ERROR: PROCNAME, ACCTNUM, GROUPID, and REGIONSZ must be provided to start a session

Receiving return code information using the native API

The following example shows the return code information received using the native API. In this example, the session identification information provided on a call to CeaTsoRequest() to end an address space does not identify a known address space:

```plaintext
Content-type: text/plain
ISPZINL started - z/OS V2R3 01JAN17 Base
Data read from STDIN is &REQUEST=PING &VER=1&ASID=97&STOKEN=0000018400000046&INDEX=2&MSGQID=3801094&TYPE=ISPF
<ISPINFO>
RC=-1
RSN=0483100A
</ISPINFO>
82 *** 'ISPZINL'
     +++ RC(255) +++
```

Repeating the command with a loglevel value that includes error information causes the following lines to be included in the output:

*** CEATsoPing failed ***
CEATsoRequest Asid     = 97
CEATsoRequest Stoken   = 0000018400000046
Legacy ISPF Gateway

This section provides information about installing, customizing, and using the Legacy ISPF Gateway. For information about the Interactive ISPF Gateway and the Legacy ISPF Gateway and the benefits of using the Interactive ISPF Gateway, see “TSO/ISPF client gateway” on page 44.

To use the Legacy ISPF Gateway, ensure that the environment variable CGI_CEATSO is not defined or that it is set to a value other than TRUE before invoking the gateway.

Legacy ISPF 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 gateway does not provide the data transport between the client and z/OS host. The gateway is designed to interface with an authenticated connection between the client and z/OS host such as 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 Legacy ISPF 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,O
  SETOMVS MAXPROCUSER=50.
  ```

  Setting a value that is too low can cause the interface to fail.
- Table 4 on page 45 describes the load modules that comprise the gateway. 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 on page 45 describes the gateway files installed on the z/OS UNIX file system.
• For additional steps that must be completed depending on the environment from which the gateway is called, see “Customizing the gateway calling environment” on page 82.

Installing the Legacy ISPF 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 Legacy ISPF 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. The maximum length of the string defining the data sets for a DD is 255 characters. If the data sets you want to define for a DD exceed this length, use the customer defined allocation exec to allocate the DD. 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
```
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
```

Note: If you have the PDSMAN product installed, it must be disabled for use with the gateway by including the record in the ISPF.conf file:

```none
ezyoff=nullfile
```

Using the Legacy ISPF 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 Legacy ISPF 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
<?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"/>
                    <!-- Reusable ISPF session stays active between calls -->
                    <xs:enumeration value="REUSE"/>
                  </xs:restriction>
                </xs:simpleType>
              </xs:element>
              <xs:element name="ispprof" type="xs:string" minOccurs="0"/>
              <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
<?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
<?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
/* REXX */
parse upper arg dsn .
address "ISPEXEC" "dsinfo dataset('"dsn"')"
if rc = 0 then do
  say 'Volume' = ' zdsvol
  say 'Primary allocation' = ' strip(zds1ex) zds1spc
  say 'Secondary allocation' = ' strip(zds2ex) zds2spc
end
else do
  say 'DSINFO rc = ' rc
  say 'ZERRMSG = ' zerrmsg
end
exit rc
```

**Note:** System symbols &SYSUID and &SYSPREF can be used as the high-level qualifier for the ISPF profile data set name specified with the <ispprof> tag.

**Receiving output from the Legacy ISPF 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
<?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:all>
  </xs:complexType>
 </xs:element>
</xs:schema>
```
Here is the XML returned for the TSO LISTCAT command above:

```xml
<?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/java142/31.4/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 PATH_INFO=
  12 CONTENT_LENGTH=274
  13 _CEE_RUNOPTS=ENVAR("_CEE_ENVFILE=//DD:ENV")
  14 _BPX_SPAWN_SCRIPT=YES
  15 _CEEREFERER=.
  16 DOCUMENT_ROOT=usr/lpp/ispf/bin
  Customizing ISPF  73
  ]]>
 </OPERATIONS-LOG>
</ISPF-OUTPUT>
```
Shown here is the XML returned from running the DINFO REXX program in a "reusable" ISPF session (see above):

```xml
<?xml version="1.0"?>
<ISPF-OUTPUT>
<SERVICE-REQUEST>
<service>ISPF</service>
<session>REUSE</session>
<command>TSO DINFO sys1.linklib</command>
</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
Secondary 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/products/java142/31.4/bin/
3 AUTH_TYPE=Basic
4 DOCUMENT_URI=/ISPZXML
]]>
</OPERATIONS-LOG>
<ISPF-OUTPUT>
```
Customizing ISPF 75
Return and reason codes from the Legacy ISPF 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="ispcpISPZ0007">ISPF OR SCLM SERVICE HAS ENDED ABNORMALLY</REASON-CODE>
```

Table 6 on page 77 lists the possible reason codes when there is a return code of 8 from the gateway:
Table 6. Possible reason codes for a return code of 8 from the gateway

<table>
<thead>
<tr>
<th>Reason Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISPZ0001</td>
<td>Error in ALLOCATION/WRITE to dsname.</td>
</tr>
<tr>
<td>ISPZ0002</td>
<td>Processing terminates.</td>
</tr>
<tr>
<td>ISPZ0003</td>
<td>Following function has failed: MSG: function</td>
</tr>
<tr>
<td>ISPZ0009</td>
<td>Error reading the ISPF.conf configuration file.</td>
</tr>
<tr>
<td>ISPZ0010</td>
<td>Ensure file exists in directory specified by the environment variable CGI_ISPCONF in the server configuration.</td>
</tr>
<tr>
<td>ISPZ0012</td>
<td>Error in ISPF data set allocation: See error message below</td>
</tr>
</tbody>
</table>
| ISPZ0013    | Error in allocating the following DD and data set names:  
MSG: DD=ddname  
MSG: dsname1 dsname2... dsnamen |
| 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 7 on page 77 lists the possible reason codes when there is a return code of 12 from the gateway:

Table 7. Possible reason codes for a return code of 12 from the gateway

<table>
<thead>
<tr>
<th>Reason Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISPZ0007</td>
<td>ISPF or SCLM service has ended abnormally.</td>
</tr>
<tr>
<td>ISPZ0008</td>
<td>Review log for error details.</td>
</tr>
</tbody>
</table>

Using the native API of the Legacy ISPF Gateway

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
```

ISPPROF may contain the strings &SYSUID., &SYSPREF. and &SYSNAME. in any order. These strings can also be used in the CGI_ISPPREF environment variable. Note that the trailing dot (.) must exist for the variable to be recognized correctly.

**&SYSUID.**
Replaced by the userid of the active session.

**&SYSPREF.**
Replaced by the TSO PREFIX, if that can be determined; otherwise, it has the same value as &SYSUID..

**&SYSNAME.**
Replaced by the SYSNAME IPL parameter.

For example, for this command (where PREFIX is mypref and SYSUID is myuser):

```
ISPF tso mycmd&SESSION=SPAWN&ISPPROF=HLQ..&SYSPREF..&SYSUID..LLQ
```

where &SYSPREF is "mypref" and &SYSUID is "myuse" the string becomes:

```
ISPF tso mycmd&SESSION=SPAWN&ISPPROF=HLQ.mypref.myuser.LLQ
```

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

```
ISPFFUNC=SHUTDOWN
```

or

```
ISPFFUNC=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>
ISPFFUNC=SHUTDOWN
</ISPINFO>
```

**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
```
<table>
<thead>
<tr>
<th>Environment Variable</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>_CEE_ENVFILE</td>
<td>//DD:ENV</td>
</tr>
<tr>
<td>PATH</td>
<td>/bin:.:/usr/sbin:/usr/lpp/internet/bin:/usr/lpp/internet/sbin:/usr/lpp/products/java531-UK14829/35.0/bin</td>
</tr>
<tr>
<td>JAVA_HOME</td>
<td>/usr/lpp/products/java142/J1.4/bin</td>
</tr>
<tr>
<td>CLASSPATH</td>
<td>/usr/lpp/internet/server_root/CAServlet</td>
</tr>
<tr>
<td>STEPLIB</td>
<td>CURRENT</td>
</tr>
<tr>
<td>_BPX_BATCH_SPAWN</td>
<td>SPAWN</td>
</tr>
<tr>
<td>_BPX_SHAREAS</td>
<td>YES</td>
</tr>
<tr>
<td>LIBPATH</td>
<td>/bin:/usr/lpp/internet/bin:/usr/lpp/internet/sbin</td>
</tr>
<tr>
<td>JAVA_HOME</td>
<td>/usr/lpp/products/java142/J1.4/</td>
</tr>
<tr>
<td>CLASSPATH</td>
<td>/usr/lpp/internet/server_root/CAServlet</td>
</tr>
<tr>
<td>COUNTERDIR</td>
<td>NULL</td>
</tr>
<tr>
<td>REPORTBITS</td>
<td>77</td>
</tr>
<tr>
<td>SERVER_SOFTWARE</td>
<td>IBM HTTP Server/V5R3M0</td>
</tr>
<tr>
<td>SERVER_NAME</td>
<td>PTHISD1.AU.IBM.COM</td>
</tr>
<tr>
<td>SERVER_PORT</td>
<td>1726</td>
</tr>
<tr>
<td>_CEERUNOPTS</td>
<td>ENVAR(&quot;_CEE_ENVFILE=//DD:ENV&quot;)</td>
</tr>
</tbody>
</table>

Number of environment variables is 59
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-UK14829/35.0/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=lib/nls/msg/%L/%N:/usr/lpp/internet/%L/%N:
     /lib/nls/msg/En_US.UTF-8
  8 LIBPATH=/bin:/usr/lpp/internet/bin:/usr/lpp/internet/sbin
  9 # JAVA_HOME=/usr/lpp/products/java531-UK14829/35.0/bin
 10 JAVA_HOME=/usr/lpp/products/java142/J1.4/
 11 CLASSPATH=/usr/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.LTRANS.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 HTTPPragma=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 ISPZTSO
Customizing the gateway calling environment

If you plan to use IBM HTTP server powered by Apache to invoke the gateway, see “Customizing IBM HTTP server powered by Apache” on page 82 for information about customizing that environment for invoking the gateway.

If you plan to use IBM z/OS Explorer's Remote Systems Explorer (RSE) to invoke the gateway, see “Customizing Remote Systems Explorer” on page 89 for information about customizing that environment for invoking the gateway.

If you plan to invoke the gateway by means other than through IBM HTTP server powered by Apache or RSE (for example, from the OMVS shell), see “Customizing other environments” on page 89 for information about customizing the environment for invoking the gateway.

Customizing IBM HTTP server powered by Apache

If you plan to use IBM HTTP server powered by Apache to invoke the gateway, changes must be made to the HTTP configuration and environment files.

Make these changes to the HTTP configuration file, httpd.conf:

• Include Alias and ScriptAlias directives to map the gateway URLs to their file system locations. The path specified in these directives must be the path where the gateway was installed. For example:

```plaintext
Alias /ISPZIVP.html /usr/lpp/ispf/bin/ISPZIVP.html
ScriptAlias /ISPZIVP.cgi /usr/lpp/ispf/bin/ISPZIVP.cgi
ScriptAlias /ISPZXML /usr/lpp/ispf/bin/ISPZXML
```

• If the gateway modules (see Table 4 on page 45) are not in the LINKLIST, include a STEPLIB directive to indicate the load library data sets that contain these modules. For example:

```
setenv STEPLIB DEV.USER.LOAD:DEV.STG.LOAD:DEV.BASE.LOAD
```

• Include LoadModule directives and a Directory directive to:
  - cause IBM HTTP server powered by Apache to prompt users to enter their user ID and password
  - invoke the gateway under the user's user ID
  - allow the gateway directory to serve content only to users who are authenticated using the System Authorization Facility (SAF) security product

The path specified in the Directory directive must be the path where the gateway was installed. For example:

```
LoadModule auth_basic_module modules/mod_auth_basic.so
LoadModule authnz_saf_module modules/mod_authnz_saf.so
<Directory /usr/lpp/ispf/bin>
  AuthName "SAF auth ISPF Gateway"
  AuthType Basic
  AuthBasicProvider saf
```

82 z/OS: ISPF Planning and Customizing
• To use the Interactive ISPF Gateway, include the following directives:
  – Set the CGI_CEATSO directive to the value TRUE:

    ```bash
    setenv CGI_CEATSO TRUE
    ```

    When the CGI_CEATSO directive is not included or is set to a value other than TRUE, the Legacy ISPF Gateway is invoked.
  – Set the LIBPATH directive to include the directory where the CEA TSO/E address space services programs are located (/usr/lib). For example:

    ```bash
    setenv LIBPATH /usr/lib
    ```

• To use the Legacy ISPF Gateway, include the following directives:
  – Set the CGI_ISPWORK directive to the path for the WORKAREA directory used by the gateway. For example, this directive specifies the default path for the WORKAREA directory:

    ```bash
    setenv CGI_ISPWORK /var/ispf
    ```

  – Set the CGI_ISPCONF directive to the path for the CONFIG directory where the ISPF configuration file ISPF.conf is stored. For example, this directive specifies the default path for the CONFIG directory:

    ```bash
    setenv CGI_ISPCONF /etc/ispf
    ```

  – Set the CGI_ISPLOGLEVEL directive to the level of logging requested from the gateway. See “Customizing other environments” on page 89 for information on the available log levels. For example, this directive requests default logging:

    ```bash
    setenv CGI_ISPLOGLEVEL 0
    ```

    Make this change to the HTTP environment file, envvars:
  • Update the path statement to include the path where the gateway was installed. If dot(,), indicating the current directory, is already specified then no update is required. In this example, the gateway path /usr/lpp/ispf/bin is added to the existing path statement:

    ```bash
    export PATH=$PATH:/etc/ihs9_rw/bin:/usr/lpp/ispf/bin
    ```

    For additional information about configuring IBM HTTP server powered by Apache, review the manuals at IBM HTTP Server in IBM Knowledge Center (www.ibm.com/support/knowledgecenter/SSEQT3).

**Verifying the IBM HTTP server powered by Apache customization**

This Installation Verification Process (IVP) applies if you have configured IBM HTTP server powered by Apache to invoke the gateway. IBM HTTP server powered by Apache must be running and the IVP Alias or ScriptAlias directives must be configured in the httpd.conf file for successful verification processing.

From a browser, type the location URL address:

```bash
http://hostname:portnumber/ISPZIVP.html
```

where:

**hostname**

  Is the name of the TCP/IP host on which IBM HTTP server powered by Apache is running.

**port number**

  Is the port used in the httpd.conf file (the default port is 80).
If IBM HTTP server powered by Apache is running, you are prompted for a valid TSO user ID and password for the system on which the Web server is running:

After you enter your TSO user ID and password, the browser displays the HTTP welcome screen:

**TSO/ISPF Client Gateway**

**Welcome to the installation and customization IVP process**

**You have successfully connected to the z/OS Host HTTP server.**

Press submit to continue the installation and customization verification process.

**Select gateway:**
- : Interactive ISPF Gateway
- : Legacy ISPF Gateway

**Provide the following for Interactive ISPF Gateway:**
- Procedure
- Acct Nmbr
- Group Ident
- Size

**Command to run from ISPF:**
- Run in a 'reusable' ISPF session
- Show operations log
- Run installation verification

*Figure 48. HTTP welcome screen*

If your connection fails, check to ensure that:
• IBM HTTP server powered by Apache has successfully initialized.
• The z/OS UNIX System Services file system mount point containing the TSO/ISPF Client Gateway installation is mounted.
• The hostname:portnumber value used in the URL is correct. If it appears to be correct, try pinging the hostname).
• There are no firewall restrictions.
• The Alias directive in the httpd.conf file is set correctly:

```
Alias  /ISPZIVP.html  /usr/lpp/ispf/bin/ISPZIVP.html
```

After you receive the welcome screen, continue with the IVP, which checks and validates your installation and customization process, by taking these steps:

To test using the Interactive ISPF Gateway:

1. Select the radio button for **Interactive ISPF Gateway**.
2. Enter the procedure name, account number, group ID, and region size to be used for the created TSO/E session.
3. Enter a TSO command (for example, TSO TIME or TSO LISTA) in the **Command to run from ISPF** field.
4. Select the **Show operations log** checkbox if you want the operations log included in the response. If the test does not complete successfully, the operations log might contain information to help you determine the reason.
5. Select the **Run installation verification** checkbox.
6. Click on the **Submit** push-button.

To test using the Legacy ISPF Gateway:

1. Select the radio button for **Legacy ISPF Gateway**.
2. Enter a TSO command (for example, TSO TIME or TSO LISTA) in the **Command to run from ISPF** field.
3. Select the **Show operations log** checkbox if you want the operations log included in the response. If the test does not complete successfully, the operations log might contain information to help you determine the reason.
4. Select the **Run installation verification** checkbox.
5. Click on the **Submit** push-button.

Figure 49 on page 86 shows an example of the expected validation responses from a test using the Interactive ISPF Gateway. In this example, TSO TIME was entered in the **Command to run from ISPF** field, the **Show operations log** checkbox was not checked, and the **Run installation verification** checkbox was checked:
TSO/ISPF Client Gateway install verification for HTTP

Review IVP log messages from HOST below:

Using Interactive ISPF Gateway
PROCNAME = ISPFPROC
ACCTNUM  = SAMPACCT
GROUPID  = DEFAULT
REGIONSZ = 2000000

CHECK1 : ENVIRONMENT VARIABLES - key variables displayed below:

- Server Name             = mvs236.rtp.raleigh.ibm.com
- Server Port             = 80
- Server PATH             = /usr/sbin:/bin:/usr/local/bin:/etc/ihs9_rw/bin:/usr/lpp/ispf/bin
- STEPLIB
- LIBPATH                 = /usr/lib
- CGI_ISPCONF             =
- CGI_ISPWORK             =
- CGI_CEATSO              = TRUE
- Current login name      = USER1
- Home directory          = /u/user1
- Current working directory (HOMEDIR) = /usr/lpp/ispf/bin

CHECK2 : HFS MODULES

Checking for Interactive ISPF Gateway modules in current directory
ISPZINT
ISPZINL
ISPZXML
ISPZXENV

CHECK3 : TSO/ISPF INITIALIZATION

( TSO/ISPF session will be initialized and run command : TSO TIME )
Command output -

TIME-02:18:46 AM. CPU-00:00:00 SERVICE-55768 SESSION-00:00:02 JULY 25,2014

Host installation verification completed successfully

Figure 49. Example of expected validation response using the Interactive ISPF Gateway

Figure 50 on page 87 shows the expected validation responses from a test using the Legacy ISPF Gateway. In this example, TSO TIME was entered in the Command to run from ISPF field, the Show operations log checkbox was not checked, and the Run installation verification checkbox was checked:
TSO/ISPF Client Gateway install verification for HTTP

Review IVP log messages from HOST below:

Using Legacy ISPF Gateway

---------------------------------------
CHECK1 : ENVIRONMENT VARIABLES - key variables displayed below:

Server Name             = mvs236.rtp.raleigh.ibm.com
Server Port             = 80
Server PATH             = /usr/sbin:/bin:/usr/local/bin:/.:/etc/ihs9_rw/bin:/usr/lpp/ispf/bin
STEPLIB                 =
LIBPATH                 =
CGI_ISPCONF             = /etc/ispf
CGI_ISPWORK             = /var/ispf
CGI_CEATSO              =
Current login name      = USER1
Home directory          = /u/user1
Current working directory (HOMEDIR) = /usr/lpp/ispf/bin

---------------------------------------
CHECK2 : HFS MODULES

Checking for Legacy ISPF Gateway modules in current directory
ISPZINT
ISPZINO
ISPZTSO
ISPZXML
ISPZXENV

---------------------------------------
CHECK3 : TSO/ISPF INITIALIZATION

( TSO/ISPF session will be initialized and run command : TSO TIME )

Command output -
  IKJ56650I TIME-02:43:49 AM. CPU-00:00:00 SERVICE-68102 SESSION-00:00:01 JULY 25,2014

Host installation verification completed successfully

---------------------------------------

Figure 50. Example of expected validation response using the Legacy ISPF Gateway

Java sample to invoke the gateway through a HTTP server

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 gateway through a HTTP server and passing XML to request running the TSO PROFILE command.

Before using this sample, you must take the actions described on the lines indicated with //**.

```java
package com.ibm.ispfcall;
import java.io.*;
import java.net.*;
import java.util.*;

//*******************************************************************/
// Before using this sample, you must take the actions described on  //
// the lines indicated with //**.                                    //
//*******************************************************************/
public class XmlIspf {
    public static void main(String[] args) {
        try {
            //** Create file C:\logon.txt, containing your user ID and password
            //** in the format USERID:PASSWORD.
            BufferedReader in = new BufferedReader(new FileReader("C:\\logon.txt"));
            String logon = in.readLine();
            in.close();
            String encodedlogon =
```
Base64.getEncoder().encodeToString(logon.getBytes("utf-8"));

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

// URL details for CGI POST
/** Replace SITE.COM with name of TCP/IP host where HTTP server
/** is running.
/** Replace 1234 with port number on which HTTP server is listening.
URL url = new URL("http", "SITE.COM", 1234, "/ISPXML");
HttpURLConnection con = (HttpURLConnection) url.openConnection();

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

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

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

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

}

}

public static void doPut(URL url, HttpURLConnection con, String ISPFFUNC, String fileIn) throws IOException
{
if (ISPFFUNC.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>");
    //** Uncomment these lines if you want to invoke the
    //** Interactive ISPF Gateway. Otherwise, see below.
    //** Replace YOURPROC, YOURACCT, YOURGRP, AND YOURSIZE
    //** with your TSO logon values.
    //** out.println("<loglevel>3</loglevel>");
    //** out.println("<request>NEWTSO</request>");
    //** out.println("<procname>YOURPROC</procname>");
    //** out.println("<acctnum>YOURACCT</acctnum>");
    //** out.println("<group>YOURGRP</group>");
    //** out.println("<regionsz>YOURSIZE</regionsz>");
    //** out.println("<cmdresp>TSO PROFILE</cmdresp>");
    //** out.println("<request>LOGOFF</request>");
    //** Uncomment these lines to invoke the Legacy ISPF Gateway.
    //** Otherwise, see above.
    //** 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");
    }
Customizing Remote Systems Explorer

If you plan to invoke the gateway using the Remote Systems Explorer (RSE) function from IBM z/OS Explorer, customize the RSE environment. For information about installing and configuring RSE, see Basic customization in IBM Developer for z Systems (www.ibm.com/support/knowledgecenter/SSQ2R2).

Customizing other environments

If you plan to invoke the gateway by means other than through the HTTP server or 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 8 on page 89. The footnotes indicate which environment variables are required for the Interactive ISPF Gateway and which environment variables are required for the Legacy ISPF Gateway.

<table>
<thead>
<tr>
<th>Environment variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>STEPLIB</td>
<td>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 45) if these modules are not in the LINKLIST. (1,2)</td>
</tr>
<tr>
<td>CGI_ISPCONF</td>
<td>Specifies the CONFIG directory path where the configuration files are stored. (2)</td>
</tr>
<tr>
<td>CGI_ISPWORK</td>
<td>Specifies the WORKAREA directory path that is used for temporary files. (2)</td>
</tr>
<tr>
<td>CGI_ISPPREF</td>
<td>Specifies the temporary data set prefix. (2)</td>
</tr>
<tr>
<td>CGI_CEATSO</td>
<td>Indicates whether the Legacy ISPF Gateway or the Interactive ISPF Gateway is invoked. (1)</td>
</tr>
</tbody>
</table>
| CGI_ISPDEBUG_MIN     | Controls whether the list of defined environment variables is written to STDOUT by the Legacy ISPF Gateway.  
**Note:** CGI_ISPLOGLEVEL is recommended as a replacement for this variable. |
| CGI_ISPLOGLEVEL      | Specifies the level of logging performed by the Legacy ISPF Gateway. |
| CGI_ISPDEBUG         | Specifies whether debug information is included in the log information produced by the Legacy ISPF Gateway.  
**Note:** CGI_ISPLOGLEVEL is recommended as a replacement for this variable. |
1 – Variable is required for the Interactive ISPF Gateway.
2 – Variable is required for the Legacy ISPF Gateway.

A sample of the REXX routine ISPZXENV is provided in member ISPZXENV in the ISPF installation directory (/usr/lpp/ispf/bin). These lines in ISPZXENV may require modification:

```
STEPLIB = 'ISP.SISPLPA:ISP.SISPLOAD'
The STEPLIB variable should specify the data sets containing the load modules for the TSO/ISPF client gateway (see Table 4 on page 45). 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 Legacy ISPF Gateway” on page 70. This variable is not used by the Interactive ISPF Gateway.

CGI_ISPWORK = '/var/ispf'
The CGI_ISPWORK variable specifies the pathname of the directory where work files for the gateway are stored. The value must be no longer than 80 characters. For more information on the work directory, see “Installing the Legacy ISPF Gateway” on page 70. This variable is not used by the Interactive ISPF Gateway.

CGI_ISPPREF
The CGI_ISPPREF variable specifies the prefix to be used to allocate cataloged z/OS work files. It supports the use of system variables &SYSUID, &SYSPREF and &SYSNAME. These system variables are described in “Using the native API of the Legacy ISPF Gateway” on page 77. If not specified, &SYSPREF..ISPF.VCMISPF is used. The prefix must resolve to a value of 35 characters or less. This variable is not used by the Interactive ISPF Gateway.

CGI_CEATSO
The CGI_CEATSO variable specifies which version of the gateway is used. If the variable is defined and is set to a value of TRUE, the Interactive ISPF Gateway is used. If the variable is not defined or is set to a value other than TRUE, the Legacy ISPF Gateway is used.

CGI_ISPDEBUG_MIN
The CGI_ISPDEBUG_MIN variable controls whether the list of defined environment variables is written to STDOUT by the Legacy ISPF Gateway. If the variable is defined and is set to a value of TRUE, the list of defined environment variables is not written to STDOUT. This can result in improved performance for a client that has many environment variables defined.

Note: CGI_ISPLOGLEVEL is recommended as a replacement for this variable. This variable is not used by the Interactive ISPF Gateway.

CGI_ISPLOGLEVEL
The CGI_ISPLOGLEVEL variable specifies the level of logging performed by the Legacy ISPF Gateway. Specify a non-zero value to activate more than the default logging. Add logging level values together to request multiple types of logging information.

1
Log debug information. Provides information about the request processing that can be useful when debugging problems. Equivalent to specifying CGI_ISPDEBUG='TRUE'.

2
Controls whether the list of defined environment variables is written to STDOUT. Include this logging level if you do not want the list of environment variables written to STDOUT. This can result in improved performance for a client that has many environment variables defined. Equivalent to specifying CGI_ISPDEBUG_MIN='TRUE'.

4
Include time stamps. Causes a time stamp value to be included in some lines of log information.

8
Include additional log information when the gateway is invoked via HTTP. This log level should be included only when debugging gateway problems in the HTTP environment.

This variable is not used by the Interactive ISPF Gateway.
**CGI_ISPDEBUG**

The CGI_ISPDEBUG variable specifies whether debug information is included in the log information produced by the Legacy ISPF Gateway. Provides information about the request processing that can be useful when debugging problems. Specify TRUE to include debug information in the log information.

**Note:** CGI_ISPLOGLEVEL is recommended as a replacement for this variable. This variable is not used by the Interactive ISPF Gateway.

Place the amended version of ISPZXENV in a directory referenced before /usr/lpp/ispf/bin in your PATH variable, otherwise you may lose your changes when system maintenance is applied.
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 “Use virtual I/O” on page 93).
- Remove or tailor edit functions (see “Remove or tailor Edit functions” on page 93).
- Remove action bars from ISPF panels (see “Remove action bars from ISPF panels” on page 94).
- Remove scrollable areas from ISPF panels (see “Remove scrollable areas from ISPF panels” on page 95).
- Preprocess ISPF panels (see “Preprocess all ISPF panels” on page 95).
- Add load modules to LPALST (see “Add load modules to the pageable link-pack area (LPA)” on page 96).
- Allocate execution data sets (see “Allocate execution data sets” on page 99).
- Disable generic searches (see “Disable generic high-level qualifiers” on page 99).
- Preallocate ISPF temporary data sets to VIO (see “Preallocate ISPF temporary data sets to VIO” on page 101).

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 NO. 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 ISPDTLC with the NOACTBAR option by specifying:

   ```
   ISPDTLC (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:

   ```
   'ISPDTLC 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 38.
**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 ISPALL (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 ISPALL (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 38.

**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.
- An INEXT section.

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 97 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.

Note:

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: FLMQOBLD, FLMQQCNT, FLMQOINT, FLMQPRM, and FLMQSTQ. 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 on page 97 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>
<thead>
<tr>
<th>Load Module</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FLM$CPI</td>
<td>SCLM</td>
</tr>
<tr>
<td>FLMB</td>
<td>SCLM</td>
</tr>
</tbody>
</table>
Table 9. Minimum recommended load modules for the LPA (continued)

<table>
<thead>
<tr>
<th>Load Module</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FLMCPCS</td>
<td>SCLM</td>
</tr>
<tr>
<td>FLMIO24</td>
<td>SCLM</td>
</tr>
<tr>
<td>FLMP</td>
<td>SCLM</td>
</tr>
<tr>
<td>FLMRTL</td>
<td>SCLM</td>
</tr>
<tr>
<td>FLMS$LNK</td>
<td>SCLM</td>
</tr>
<tr>
<td>FLMRTLIB</td>
<td>SCLM</td>
</tr>
<tr>
<td>ISPICP</td>
<td>ISPSTART</td>
</tr>
<tr>
<td>ISPISM</td>
<td>Settings processor</td>
</tr>
<tr>
<td>ISPKEY</td>
<td>KEYLIST command processor</td>
</tr>
<tr>
<td>ISPKLU</td>
<td>KEYLIST Utility</td>
</tr>
<tr>
<td>ISPLLP</td>
<td>Log/List processor</td>
</tr>
<tr>
<td>ISPMAIN</td>
<td>ISPF controller</td>
</tr>
<tr>
<td>ISPNLxxx</td>
<td>where xxx designates the languages you are using</td>
</tr>
<tr>
<td>ISPOPI</td>
<td>Panel ID processor</td>
</tr>
<tr>
<td>ISPOPT</td>
<td>Options processor</td>
</tr>
<tr>
<td>ISPOPF</td>
<td>PFSHOW processor</td>
</tr>
<tr>
<td>ISPOPF</td>
<td>PFSHOW processor</td>
</tr>
<tr>
<td>ISPPOMI</td>
<td>MSGID processor</td>
</tr>
<tr>
<td>ISRBRO</td>
<td>Main Browse module</td>
</tr>
<tr>
<td>ISREDIT</td>
<td>Main Edit module</td>
</tr>
<tr>
<td>ISRNLxxx</td>
<td>where xxx designates the languages you are using</td>
</tr>
<tr>
<td>ISPQGUI</td>
<td>ISPF C/S processor</td>
</tr>
<tr>
<td>ISPSUBS</td>
<td>Common subroutines (alias name: ISPCIP)</td>
</tr>
<tr>
<td>ISPSUBX</td>
<td>Common subroutines</td>
</tr>
<tr>
<td>ISPTASK</td>
<td>Dialog driver</td>
</tr>
<tr>
<td>ISPTCM</td>
<td>TSO command table</td>
</tr>
<tr>
<td>ISPTUTOR</td>
<td>Tutorial processor (option T)</td>
</tr>
<tr>
<td>ISPYLxxx</td>
<td>where xxx designates the languages you are using</td>
</tr>
<tr>
<td>ISP32nnn</td>
<td>where nnn designates the terminal types in use</td>
</tr>
<tr>
<td>ISRBRO</td>
<td>Main Browse module</td>
</tr>
<tr>
<td>ISREDIT</td>
<td>Main Edit module</td>
</tr>
<tr>
<td>ISRNLxxx</td>
<td>where xxx designates the languages you are using</td>
</tr>
<tr>
<td>ISRPLEX</td>
<td>Edit extended highlight module</td>
</tr>
<tr>
<td>ISRPX</td>
<td>Edit extended highlight customization module</td>
</tr>
<tr>
<td>ISRRCL</td>
<td>Action bar router utility</td>
</tr>
<tr>
<td>ISRSEPRM</td>
<td>More SuperC</td>
</tr>
<tr>
<td>ISRSUBS</td>
<td>Main PDF subroutine module</td>
</tr>
<tr>
<td>ISRSUBX</td>
<td>Main below 16M PDF subroutine module</td>
</tr>
<tr>
<td>ISRSEPRM</td>
<td>More SuperC</td>
</tr>
<tr>
<td>ISRSUPC</td>
<td>Superc</td>
</tr>
</tbody>
</table>
Table 9. Minimum recommended load modules for the LPA (continued)

<table>
<thead>
<tr>
<th>Load Module</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISRUDA</td>
<td>PDF Utilities</td>
</tr>
<tr>
<td>ISRUMC</td>
<td>PDF Move Copy</td>
</tr>
<tr>
<td>ISR32nnn</td>
<td>where nnn designates the terminal types in use</td>
</tr>
</tbody>
</table>

ISRSUPC is shipped RMODE(24) and is large. Installations that do not use Edit Compare or Superc functions frequently could consider moving this module from LPA to free up below the line storage.

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 193.
Here 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.

```assembly
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    @01,005
XR    @02,002

* Translate and test for *, %, blank, or period
TRT   DSNAME(44,@05),TABLE
SR    @01,005
AR    @05,001

* 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   @06,006
ST    @06,EXITRC
EXITRTN  DS    0H
L     @15,EXITRC
L     @14,12(,@13)
LM    @00,012,20(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'\%'

* 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
```
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 51 on page 102 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.
In this section of JCL, there is one DD for each screen defined, based on the value of keyword MAXIMUM_NUMBER_OF_SPLIT_SCREENS 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.

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

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

**Note:**

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 ISPFFILE 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 FTOPEN 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:

- “Prohibiting connections to the ISPF Workstation Agent” on page 105
- “Changing the default name for the ISPF Client/Server component download data set” on page 106
- “Set up ISPF GUI for TSO Line Mode support” on page 106
- “SMF command accounting” on page 106
- “Pre-allocation of List/Log data sets” on page 109
- “Specifying the maximum number of split screens” on page 110
- “Setting ISPF site-wide defaults” on page 110
- “Customizing command tables” on page 111
- “Creating ISPF terminal translation tables” on page 112
- “Creating ISPF code page translation tables” on page 120
- “Displaying square brackets used in C programs” on page 129
- “ISPEXEC processing” on page 129
- “ISPF-to-APL2 terminal type mappings” on page 130
- “Load APL2 workspace” on page 131
- “Tailoring ISPF defaults” on page 131
- “Customizing the ISPF TSO command table (ISPTCM)” on page 131
- “Alternate option 7.1 panels” on page 135
- “ISPF multicultural support” on page 136

See also Chapter 6, “ISPF installation-wide exits,” on page 137.

Prohibiting connections to the ISPF Workstation Agent

The communication between ISPF and the ISPF Workstation Agent is not secure. Therefore, avoid using the ISPF Workstation Agent. z/OS V2R4 is planned to be the last release to support the ISPF Workstation Agent.

System Authorization Facility (SAF) resource ISPF.WSA, in the FACILITY class, can be used to prohibit users from initiating a connection from ISPF to the ISPF Workstation Agent.

Use the following commands to define the resource ISPF.WSA in the FACILITY class with a universal access (UACC) of NONE. This will prohibit all users from initiating a connection to the ISPF Workstation Agent.

```
RDEFINE FACILITY ISPF.WSA UACC(NONE)
SETROPTS RACLIST(FACILITY) REFRESH
```

A sample job to issue the commands is supplied in member ISPRACF in the ISPF samples data set ISP.SISPSAMP.
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: ISPDTSSI, ISPDTWIN, ISPDTTSSK, ISPDTINI, IKJDTINI, ISPDTTER, IKJDTTER, ISPDTPUT, IKJDTPUT, ISPDTGET, IKJDTGET, ISPDTPC, and ISPOGUI.

• The ISPF subsystem initialization routine, ISPDTSSI, 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:

    ISPDX001 ISPF Subsystem Initialization: Completed successfully
    ISPDX002 ISPF Subsystem Initialization: Error creating PC number - is ISPDTPC in LPA
    ISPDX003 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 131 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 ISPFSWAP 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 on page 107 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

<table>
<thead>
<tr>
<th>Option</th>
<th>Module Names</th>
<th>TSO Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>ISPISM</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>ISRBRO</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>ISREDIT</td>
<td>SUBMIT</td>
</tr>
<tr>
<td>3.1 and 3.2</td>
<td>ISRUDA</td>
<td></td>
</tr>
<tr>
<td>3.3</td>
<td>ISRUMC</td>
<td></td>
</tr>
<tr>
<td>3.4</td>
<td>ISRUDL</td>
<td></td>
</tr>
<tr>
<td>3.5</td>
<td>ISRURS</td>
<td></td>
</tr>
<tr>
<td>3.6</td>
<td>ISRUHC</td>
<td>SUBMIT, ICQCPC00</td>
</tr>
<tr>
<td>3.7</td>
<td>ISPWSD</td>
<td></td>
</tr>
<tr>
<td>3.8</td>
<td>ISRUOLP</td>
<td>STATUS, OUTPUT, SUBMIT, ICQCPC00</td>
</tr>
<tr>
<td>3.9</td>
<td>ISPUCM</td>
<td></td>
</tr>
<tr>
<td>3.11</td>
<td>ISRFMT</td>
<td></td>
</tr>
<tr>
<td>3.12</td>
<td>ISRSSM</td>
<td></td>
</tr>
<tr>
<td>3.13 and 3.15</td>
<td>ISRSEPRM</td>
<td></td>
</tr>
<tr>
<td>3.14</td>
<td>ISRSFM</td>
<td></td>
</tr>
<tr>
<td>3.16</td>
<td>ISRUTABL</td>
<td></td>
</tr>
<tr>
<td>3.17</td>
<td>ISRUUDL</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>ISRFPR</td>
<td>ASM, FORT, PLI, PLIC, LINK, TESTCOB, TESTFORT, CALL, SCRIPT, ALLOC, FREE, ICQCPC00</td>
</tr>
<tr>
<td>5</td>
<td>ISRJB1</td>
<td>SUBMIT</td>
</tr>
<tr>
<td>6</td>
<td>ISRPTC</td>
<td>All commands except those prohibited by ISPF in ISPTCM</td>
</tr>
<tr>
<td>7</td>
<td>ISPYXDR</td>
<td></td>
</tr>
<tr>
<td>7.1</td>
<td>ISPYFI</td>
<td></td>
</tr>
</tbody>
</table>
### Table 10. ISPF Options and related module names (continued)

<table>
<thead>
<tr>
<th>Option</th>
<th>Module Names</th>
<th>TSO Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.2</td>
<td>ISPYPI</td>
<td></td>
</tr>
<tr>
<td>7.3</td>
<td>ISPYVI</td>
<td></td>
</tr>
<tr>
<td>7.4</td>
<td>ISPYTI</td>
<td></td>
</tr>
<tr>
<td>7.5</td>
<td>ISPYLI</td>
<td></td>
</tr>
<tr>
<td>7.6</td>
<td>ISPSI</td>
<td></td>
</tr>
<tr>
<td>7.7.1</td>
<td>ISPYRFI</td>
<td></td>
</tr>
<tr>
<td>7.7.2</td>
<td>ISPYRVI</td>
<td></td>
</tr>
<tr>
<td>7.8</td>
<td>ISPYBI</td>
<td></td>
</tr>
<tr>
<td>7.9</td>
<td>ISPTUTOR</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>ISRALTDI</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>FLMDDL</td>
<td></td>
</tr>
<tr>
<td>10.1</td>
<td>FLMEB$</td>
<td></td>
</tr>
<tr>
<td>10.2</td>
<td>FLMED$</td>
<td></td>
</tr>
<tr>
<td>10.3</td>
<td>FLMUDU$</td>
<td></td>
</tr>
<tr>
<td>10.4</td>
<td>FLMBD$</td>
<td></td>
</tr>
<tr>
<td>10.5</td>
<td>FLMPD$</td>
<td></td>
</tr>
<tr>
<td>10.6</td>
<td>FLMPTC</td>
<td></td>
</tr>
<tr>
<td>10.A</td>
<td>FLMA</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>ISRUDA</td>
<td></td>
</tr>
<tr>
<td>X</td>
<td>ISPLL</td>
<td>SUBMIT</td>
</tr>
</tbody>
</table>

Table 11 on page 108 lists the ISPF commands with the related modules:

### Table 11. ISPF commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Module Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>COLOR</td>
<td>ISPOPT</td>
</tr>
<tr>
<td>CUAATTR</td>
<td>ISPOPT</td>
</tr>
<tr>
<td>ENVIRON</td>
<td>ISPENV</td>
</tr>
<tr>
<td>EXHELP</td>
<td>ISPTUTOR</td>
</tr>
<tr>
<td>FKA</td>
<td>ISPOPF</td>
</tr>
<tr>
<td>HELP</td>
<td>ISPTUTOR</td>
</tr>
<tr>
<td>ISPFVAR</td>
<td>ISPISM</td>
</tr>
<tr>
<td>ISPLIBD</td>
<td>ISPLLS</td>
</tr>
<tr>
<td>ISPPREP</td>
<td>ISPPREP</td>
</tr>
<tr>
<td>ISRLIST</td>
<td>ISRDSLST</td>
</tr>
</tbody>
</table>
Table 11. ISPF commands (continued)

<table>
<thead>
<tr>
<th>Command</th>
<th>Module Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISRROUTE</td>
<td>ISRRCL</td>
</tr>
<tr>
<td>KEYLIST</td>
<td>ISPKLU</td>
</tr>
<tr>
<td>KEYS</td>
<td>ISPOPT</td>
</tr>
<tr>
<td>KEYSHELP</td>
<td>ISPTUTOR</td>
</tr>
<tr>
<td>LIST</td>
<td>ISPLLPL</td>
</tr>
<tr>
<td>LOG</td>
<td>ISPLLPL</td>
</tr>
<tr>
<td>MSGID</td>
<td>ISPOMI</td>
</tr>
<tr>
<td>PANELID</td>
<td>ISPOPI</td>
</tr>
<tr>
<td>PFSHOW</td>
<td>ISPOPF</td>
</tr>
<tr>
<td>PSCOLOR</td>
<td>ISPOPT</td>
</tr>
<tr>
<td>REFACTD</td>
<td>ISRDSLST</td>
</tr>
<tr>
<td>REFACTL</td>
<td>ISRDSLST</td>
</tr>
<tr>
<td>REFADDD</td>
<td>ISRRSLST</td>
</tr>
<tr>
<td>REFADDL</td>
<td>ISRRSLST</td>
</tr>
<tr>
<td>REFLISTD</td>
<td>ISRDSLST</td>
</tr>
<tr>
<td>REFLISTL</td>
<td>ISRDSLST</td>
</tr>
<tr>
<td>REFOPEND</td>
<td>ISRDSLST</td>
</tr>
<tr>
<td>REFOPENL</td>
<td>ISRDSLST</td>
</tr>
<tr>
<td>SAREA</td>
<td>ISPSAM</td>
</tr>
<tr>
<td>SETTINGS</td>
<td>ISPIISM</td>
</tr>
<tr>
<td>START</td>
<td>ISPSTRT</td>
</tr>
<tr>
<td>TSOCMD</td>
<td>ISRPTC</td>
</tr>
<tr>
<td>TUTOR</td>
<td>ISPTUTOR</td>
</tr>
<tr>
<td>ZKEYS</td>
<td>ISPOPT</td>
</tr>
</tbody>
</table>

**Pre-allocation of List/Log data sets**

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 pre-allocate the list or log data set to a sequential data set. If you do this, the data sets are automatically saved when the user logs off TSO. If both data sets are pre-allocated, the termination menu is bypassed when the user exits from ISPF. If the user reenters ISPF before logging off, any new output is added to the end of the sequential data sets.
You can pre-allocate the list and log data sets directly to SYSOUT by including 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 pre-allocate these data sets to SYSOUT, they are automatically printed when the user logs off TSO. If both data sets are pre-allocated, the termination menu is bypassed when the user exits from ISPF. If the user reenters ISPF before logging off, any new output is added to the end of the SYSOUT data sets.

**Note:**

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. 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.

See *Terminating a dialog in z/OS ISPF Dialog Developer's Guide and Reference* for more information.

---

**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_SCREENS 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.
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.
3. Add another 256-character table labeled TTUPP in which all characters translate to themselves except X'08', X'1C', X'1D', and X'1E', which translate to X'40'.

### Uppercase character translation table

The uppercase character translation table (TTUPP in the example shown in Table 12 on page 113) translates screen input data as follows:

- Lowercase alphabetic characters translate to uppercase.
- X'08', X'1C', X'1D', and X'1E' translate to blank (X'40').
- All other hexadecimal values translate to themselves.

**Table 12. Uppercase character translation table example**

<table>
<thead>
<tr>
<th>Table</th>
<th>Hexadecimal Code</th>
<th>Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>TTUPP</td>
<td>DC X'9001020304050607'</td>
<td>(X'00' to X'07')</td>
</tr>
<tr>
<td></td>
<td>DC X'40090A0B0C0D0E0F'</td>
<td>(X'08' to X'0F')</td>
</tr>
<tr>
<td></td>
<td>DC X'1011121314151617'</td>
<td>(X'10' to X'17')</td>
</tr>
<tr>
<td></td>
<td>DC X'7B797A7B7C7D7F7F'</td>
<td>(X'78' to X'7F')</td>
</tr>
<tr>
<td></td>
<td>DC X'B0C1C2C3C4C5C6C7'</td>
<td>(X'80' to X'87')</td>
</tr>
<tr>
<td></td>
<td>DC X'E8E9EAEBCDEEEF'</td>
<td>(X'E8' to X'E.F')</td>
</tr>
<tr>
<td></td>
<td>DC X'F0F1F2F3F4F5F6F7'</td>
<td>(X'F0' to X'F7')</td>
</tr>
<tr>
<td></td>
<td>DC X'F8F9FAFBFCDFEFEFF'</td>
<td>(X'F8' to X'FF')</td>
</tr>
</tbody>
</table>

Table 12 on page 113 shows how the uppercase character translation table might be represented in the assembler module. For example, this is true for Table 12 on page 113:

- The hexadecimal position for a lowercase ‘a’ (X'81'), contains the hexadecimal value for an uppercase ‘A’ (X'C1').
- The hexadecimal position for an uppercase ‘A’ (X'C1') contains the hexadecimal value for an uppercase ‘A’ (X'C1').

Enter the values you want in the 256 hexadecimal positions of the uppercase translation table. When you finish with the table, you are ready to move on to the second translation table.

### Lowercase character translation table

The lowercase character translation table (TTLOW) must be left as it is. Its function is internal to ISPF.
Valid terminal output translation table

The valid terminal output translation table (TTVAL in the example shown in Table 13 on page 114) represents display characters, in hexadecimal, as follows:

- Valid display characters are represented with X'00'.
- Invalid display characters are represented with X'FF'.

<table>
<thead>
<tr>
<th>Table</th>
<th>Hexadecimal Code</th>
<th>Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>TTVAL</td>
<td>DC X'FFFFFFFFFFFFFFFF'</td>
<td>(X'00' to X'07')</td>
</tr>
<tr>
<td></td>
<td>DC X'FFFFFFFFFFFFFFFF'</td>
<td>(X'08' to X'0F')</td>
</tr>
<tr>
<td></td>
<td>DC X'FFFFFFFFFFFFFFFF'</td>
<td>(X'10' to X'17')</td>
</tr>
<tr>
<td></td>
<td>...</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DC X'FF00000000000000'</td>
<td>(X'78' to X'7F')</td>
</tr>
<tr>
<td></td>
<td>DC X'FF00000000000000'</td>
<td>(X'80' to X'87')</td>
</tr>
<tr>
<td></td>
<td>...</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DC X'0000000000000000'</td>
<td>(X'E8' to X'EF')</td>
</tr>
<tr>
<td></td>
<td>DC X'0000000000000000'</td>
<td>(X'F0' to X'F7')</td>
</tr>
<tr>
<td></td>
<td>DC X'0000000000000000'</td>
<td>(X'F8' to X'FF')</td>
</tr>
</tbody>
</table>

Generic string master translation table

The positions in the generic string master translation table (TTGSM in the example shown in Table 14 on page 115) 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>
<thead>
<tr>
<th>Table</th>
<th>Hexadecimal Code</th>
<th>Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>TTGSM</td>
<td>DC X'0101010101010101'</td>
<td>(X'00' to X'07')</td>
</tr>
<tr>
<td></td>
<td>DC X'0101010101010101'</td>
<td>(X'08' to X'0F')</td>
</tr>
<tr>
<td></td>
<td>DC X'0101010101010101'</td>
<td>(X'10' to X'17')</td>
</tr>
<tr>
<td></td>
<td>DC X'0102020202020202'</td>
<td>(X'78' to X'7F')</td>
</tr>
<tr>
<td></td>
<td>DC X'0110101010101010'</td>
<td>(X'80' to X'87')</td>
</tr>
<tr>
<td></td>
<td>DC X'2020010101010101'</td>
<td>(X'E8' to X'EF')</td>
</tr>
<tr>
<td></td>
<td>DC X'4040404040404040'</td>
<td>(X'F0' to X'F7')</td>
</tr>
<tr>
<td></td>
<td>DC X'4040010101010101'</td>
<td>(X'F8' to X'FF')</td>
</tr>
</tbody>
</table>

### 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 284 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 ($), and the at sign (@) (Table 15 on page 115).
- **TTALB**, which does not include the number sign (#), the dollar sign ($), and the at sign (@) (Table 16 on page 116).

**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>
<thead>
<tr>
<th>Table</th>
<th>Hexadecimal Code</th>
<th>Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>TTALP</td>
<td>DC X'FFFFFFFFFFFFFFF'</td>
<td>(X'00' to X'07')</td>
</tr>
<tr>
<td></td>
<td>DC X'FFFFFFFFFFFFFFF'</td>
<td>(X'08' to X'0F')</td>
</tr>
<tr>
<td></td>
<td>DC X'FFFFFFFFFFFFFFF'</td>
<td>(X'10' to X'17')</td>
</tr>
<tr>
<td></td>
<td>DC X'FFFFFFFFFFFFFFF'</td>
<td>(X'78' to X'7F')</td>
</tr>
<tr>
<td></td>
<td>DC X'FFFFFFFFFFFFFFF'</td>
<td>(X'80' to X'87')</td>
</tr>
<tr>
<td></td>
<td>DC X'FFFFFFFFFFFFFFF'</td>
<td>(X'E8' to X'EF')</td>
</tr>
<tr>
<td></td>
<td>DC X'FFFFFFFFFFFFFFF'</td>
<td>(X'F0' to X'F7')</td>
</tr>
<tr>
<td></td>
<td>DC X'FFFFFFFFFFFFFFF'</td>
<td>(X'F8' to X'FF')</td>
</tr>
</tbody>
</table>
Table 16. Sample alphabetic character translation table excluding #, $, and @

<table>
<thead>
<tr>
<th>Table</th>
<th>Hexadecimal Code</th>
<th>Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>TTALB</td>
<td>DC X'FFFFFFFFFFFFFFFFF'</td>
<td>(X'00' to X'07')</td>
</tr>
<tr>
<td></td>
<td>DC X'FFFFFFFFFFFFFFFFF'</td>
<td>(X'08' to X'0F')</td>
</tr>
<tr>
<td></td>
<td>DC X'FFFFFFFFFFFFFFFFF'</td>
<td>(X'10' to X'17')</td>
</tr>
<tr>
<td></td>
<td>DC X'FFFFFFFFFFFFFFFFF'</td>
<td>(X'50' to X'57')</td>
</tr>
<tr>
<td></td>
<td>DC X'FFFFFFFFFFFFFFFFF'</td>
<td>(X'58' to X'5F')</td>
</tr>
<tr>
<td></td>
<td>DC X'FFFFFFFFFFFFFFFFF'</td>
<td>(X'78' to X'7F')</td>
</tr>
<tr>
<td></td>
<td>DC X'FFFFFFFFFFFFFFFFF'</td>
<td>(X'80' to X'87')</td>
</tr>
<tr>
<td></td>
<td>DC X'FFFFFFFFFFFFFFFFF'</td>
<td>(X'88' to X'8F')</td>
</tr>
<tr>
<td></td>
<td>DC X'FFFFFFFFFFFFFFFFF'</td>
<td>(X'E8' to X'EF')</td>
</tr>
<tr>
<td></td>
<td>DC X'FFFFFFFFFFFFFFFFF'</td>
<td>(X'F0' to X'F7')</td>
</tr>
<tr>
<td></td>
<td>DC X'FFFFFFFFFFFFFFFFF'</td>
<td>(X'F8' to X'FF')</td>
</tr>
</tbody>
</table>

Collating sequence translation table

The collating sequence translation table (TTCOL in the example shown in Table 17 on page 116) 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 116, 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 on page 116 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>
<thead>
<tr>
<th>Table</th>
<th>Hexadecimal Code</th>
<th>Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>TTCOL</td>
<td>DC X'0001020304050607'</td>
<td>(X'00' to X'07')</td>
</tr>
<tr>
<td></td>
<td>DC X'08090A0B0C0D0E0F'</td>
<td>(X'08' to X'0F')</td>
</tr>
<tr>
<td></td>
<td>DC X'1011121314151617'</td>
<td>(X'10' to X'17')</td>
</tr>
<tr>
<td></td>
<td>DC X'78797A7B7C7D7E7F'</td>
<td>(X'78' to X'7F')</td>
</tr>
<tr>
<td></td>
<td>DC X'8081828384858687'</td>
<td>(X'80' to X'87')</td>
</tr>
<tr>
<td></td>
<td>DC X'E8E9EAEDEEFEFF'</td>
<td>(X'E8' to X'EF')</td>
</tr>
<tr>
<td></td>
<td>DC X'F0F1F2F3F4F5F6F7'</td>
<td>(X'F0' to X'F7')</td>
</tr>
<tr>
<td></td>
<td>DC X'F8F9FAFBCFDFEFF'</td>
<td>(X'F8' to X'FF')</td>
</tr>
</tbody>
</table>

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

116 z/OS: ISPF Planning and Customizing
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 ISPOWNTT)
• 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 117
- “Changing the DTL source for ISPOPTxx panels” on page 119
- “Invoking ISPTTDEF” on page 120

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 ISPZMMMO. 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 ISPZMMMO are complete.

The last <SELFLD tag in DTL source member ISPISMMN (see Figure 52 on page 117) 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 52. 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 53 on page 117) and creates panel logic to determine which selection number will match a certain terminal type.

<table>
<thead>
<tr>
<th>Terminal Type</th>
<th>3</th>
<th>1. 3277</th>
<th>2. 3277A</th>
<th>3. 3278</th>
<th>4. 3278A</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5</td>
<td>3298A</td>
<td>6. 3278T</td>
<td>7. 3278CF</td>
<td>8. 3278A</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>3278KN</td>
<td>10. 3278AR</td>
<td>11. 3278CY</td>
<td>12. 3278HN</td>
</tr>
<tr>
<td></td>
<td>13 3278HO</td>
<td>14. 3278IS</td>
<td>15. 3278L2</td>
<td>16. BE163</td>
<td></td>
</tr>
<tr>
<td></td>
<td>17</td>
<td>BE190</td>
<td>18. 3278TH</td>
<td>19. 3278CU</td>
<td>20. DEU78</td>
</tr>
<tr>
<td></td>
<td>21</td>
<td>DEU78A</td>
<td>22. DEU78T</td>
<td>23. DEU98A</td>
<td>24. SW116</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>SW131</td>
<td>26. SW500</td>
<td>27. 3278GR</td>
<td>28. 3278L1</td>
</tr>
<tr>
<td></td>
<td>29</td>
<td>OTHER</td>
<td>30. XXXX</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 53. Example: adding a terminal type to panel ISPISMMN

Modified DTL source for the English-language section of the member ISPZMMCH follows:
Figure 54. 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 52 on page 117 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 ISPOWNTT. Modified DTL source for the English-language section of the member ISPZMMSO follows:

```xml
<condexec lang=english>
  <choice selchar=1 checkvar=ztermp match=3277>3277</choice>
  <choice selchar=5 checkvar=ztermp match=3290A>3290A</choice>
  <choice selchar=9 checkvar=ztermp match=327KN>327KN</choice>
  <choice selchar=13 checkvar=ztermp match=3278NO>3278NO</choice>
  <choice selchar=17 checkvar=ztermp match=BE190>BE190</choice>
  <choice selchar=21 checkvar=ztermp match=DEU78A>DEU78A</choice>
  <choice selchar=25 checkvar=ztermp match=Sw131>Sw131</choice>
  <choice selchar=29 checkvar=ztermp match=DEU78T>DEU78T</choice>
  <choice selchar=33 checkvar=ztermp match=Sw500>Sw500</choice>
  <choice selchar=37 checkvar=ztermp match=3278CF>3278CF</choice>
  <choice selchar=41 checkvar=ztermp match=3278CY>3278CY</choice>
  <choice selchar=45 checkvar=ztermp match=3278L2>3278L2</choice>
  <choice selchar=49 checkvar=ztermp match=3278CU>3278CU</choice>
  <choice selchar=53 checkvar=ztermp match=DEU90A>DEU90A</choice>
  <choice selchar=57 checkvar=ztermp match=3278GR>3278GR</choice>
  <choice selchar=61 checkvar=ztermp match=3278A>3278A</choice>
  <choice selchar=65 checkvar=ztermp match=3277KN>3277KN</choice>
  <choice selchar=69 checkvar=ztermp match=3278HN>3278HN</choice>
  <choice selchar=73 checkvar=ztermp match=BE163>BE163</choice>
  <choice selchar=77 checkvar=ztermp match=DEU78>DEU78</choice>
  <choice selchar=81 checkvar=ztermp match=Sw116>Sw116</choice>
  <choice selchar=85 checkvar=ztermp match=3278BL>3278BL</choice>
  <choice selchar=89 checkvar=ztermp match=3278L1>3278L1</choice>
  <choice selchar=93 checkvar=ztermp match=3278L1>3278L1</choice>
  <choice selchar=97 checkvar=ztermp match=3278>3278</choice>
  <choice selchar=101 checkvar=ztermp match=3278CF>3278CF</choice>
  <choice selchar=105 checkvar=ztermp match=3278GR>3278GR</choice>
  <choice selchar=109 checkvar=ztermp match=3278>3278</choice>
  <choice selchar=113 checkvar=ztermp match=3278CF>3278CF</choice>
  <choice selchar=117 checkvar=ztermp match=3278GR>3278GR</choice>
</condexec>
```
Figure 55. DTL source for valid terminal types and associated load module names

In Figure 55 on page 119, 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 148 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 ISPDTLC 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:
Figure 56. 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>
<thead>
<tr>
<th>CCSID</th>
<th>Character Set</th>
<th>Code Page</th>
<th>Country/Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>00037</td>
<td>697</td>
<td>37</td>
<td>U.S.A. Canada Netherlands Portugal Brazil Australia New Zealand</td>
</tr>
<tr>
<td>00273</td>
<td>697</td>
<td>273</td>
<td>Austria Germany</td>
</tr>
<tr>
<td>00277</td>
<td>697</td>
<td>277</td>
<td>Denmark Norway</td>
</tr>
<tr>
<td>00278</td>
<td>697</td>
<td>278</td>
<td>Finland Sweden</td>
</tr>
<tr>
<td>00280</td>
<td>697</td>
<td>280</td>
<td>Italy</td>
</tr>
<tr>
<td>00284</td>
<td>697</td>
<td>284</td>
<td>Spain L.A. Spanish</td>
</tr>
<tr>
<td>00285</td>
<td>697</td>
<td>285</td>
<td>United Kingdom</td>
</tr>
<tr>
<td>00297</td>
<td>697</td>
<td>297</td>
<td>France</td>
</tr>
<tr>
<td>00420</td>
<td>235</td>
<td>420</td>
<td>Arabic</td>
</tr>
<tr>
<td>00424</td>
<td>941</td>
<td>424</td>
<td>Hebrew</td>
</tr>
<tr>
<td>00500</td>
<td>697</td>
<td>500</td>
<td>Switzerland Belgium</td>
</tr>
<tr>
<td>00838</td>
<td>1176</td>
<td>838</td>
<td>Thailand</td>
</tr>
<tr>
<td>00870</td>
<td>959</td>
<td>870</td>
<td>Latin-2</td>
</tr>
<tr>
<td>00871</td>
<td>697</td>
<td>871</td>
<td>Iceland</td>
</tr>
<tr>
<td>00875</td>
<td>923</td>
<td>875</td>
<td>Greece</td>
</tr>
<tr>
<td>00880</td>
<td>960</td>
<td>880</td>
<td>Cyrillic</td>
</tr>
<tr>
<td>01025</td>
<td>1150</td>
<td>1025</td>
<td>Cyrillic</td>
</tr>
<tr>
<td>01026</td>
<td>1126</td>
<td>1026</td>
<td>Turkey</td>
</tr>
<tr>
<td>01047</td>
<td>697</td>
<td>1047</td>
<td>Latin1</td>
</tr>
<tr>
<td>01123</td>
<td>1326</td>
<td>1123</td>
<td>Ukraine</td>
</tr>
<tr>
<td>CCSID</td>
<td>Character Set</td>
<td>Code Page</td>
<td>Country/Language</td>
</tr>
<tr>
<td>---------</td>
<td>---------------</td>
<td>-----------</td>
<td>------------------</td>
</tr>
<tr>
<td>00924</td>
<td>1353</td>
<td>0924</td>
<td>Latin9</td>
</tr>
<tr>
<td>01140</td>
<td>695</td>
<td>1140</td>
<td>U.S.A. Canada</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Netherlands</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Portugal</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Brazil</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Australia</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>New Zealand</td>
</tr>
<tr>
<td>01141</td>
<td>695</td>
<td>1141</td>
<td>Austria Germany</td>
</tr>
<tr>
<td>01142</td>
<td>695</td>
<td>1142</td>
<td>Denmark Norway</td>
</tr>
<tr>
<td>01143</td>
<td>695</td>
<td>1143</td>
<td>Finland Sweden</td>
</tr>
<tr>
<td>01144</td>
<td>695</td>
<td>1144</td>
<td>Italy</td>
</tr>
<tr>
<td>01145</td>
<td>695</td>
<td>1145</td>
<td>Spain L.A. Spanish</td>
</tr>
<tr>
<td>01146</td>
<td>695</td>
<td>1146</td>
<td>United Kingdom</td>
</tr>
<tr>
<td>01147</td>
<td>695</td>
<td>1147</td>
<td>France</td>
</tr>
<tr>
<td>01148</td>
<td>695</td>
<td>1148</td>
<td>Switzerland Belgium</td>
</tr>
<tr>
<td>01149</td>
<td>695</td>
<td>1149</td>
<td>Iceland</td>
</tr>
<tr>
<td>01153</td>
<td>1375</td>
<td>1153</td>
<td>Latin2</td>
</tr>
<tr>
<td>01154</td>
<td>1381</td>
<td>1154</td>
<td>Cyrillic</td>
</tr>
<tr>
<td>01155</td>
<td>1378</td>
<td>1155</td>
<td>Turkey</td>
</tr>
<tr>
<td>01158</td>
<td>1388</td>
<td>1158</td>
<td>Ukraine</td>
</tr>
<tr>
<td>01160</td>
<td>1395</td>
<td>1160</td>
<td>Thailand</td>
</tr>
<tr>
<td>04899</td>
<td>1356</td>
<td>0803</td>
<td>Hebrew</td>
</tr>
<tr>
<td>04971</td>
<td>1371</td>
<td>0875</td>
<td>Greece</td>
</tr>
<tr>
<td>12712</td>
<td>1357</td>
<td>0424</td>
<td>Hebrew</td>
</tr>
<tr>
<td>16804</td>
<td>1461</td>
<td>0420</td>
<td>Arabic</td>
</tr>
</tbody>
</table>

These Extended CCSIDs (shown in Table 20 on page 123) 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

<table>
<thead>
<tr>
<th>CCSID</th>
<th>Character Set</th>
<th>Code Page</th>
<th>Country</th>
</tr>
</thead>
<tbody>
<tr>
<td>00930</td>
<td>1172</td>
<td>290</td>
<td>Japanese (Katakana)</td>
</tr>
<tr>
<td>00939</td>
<td>1172</td>
<td>1027</td>
<td>Japanese (Latin)</td>
</tr>
<tr>
<td>00933</td>
<td>1173</td>
<td>833</td>
<td>Korean</td>
</tr>
<tr>
<td>00935</td>
<td>1174</td>
<td>836</td>
<td>Simplified Chinese</td>
</tr>
<tr>
<td>00937</td>
<td>1175</td>
<td>037</td>
<td>Traditional Chinese</td>
</tr>
<tr>
<td>01159</td>
<td>65535</td>
<td>1159</td>
<td>Traditional Chinese</td>
</tr>
<tr>
<td>01364</td>
<td>65535</td>
<td>0834</td>
<td>Korean</td>
</tr>
<tr>
<td>01371</td>
<td>65535</td>
<td>0835</td>
<td>Traditional Chinese</td>
</tr>
<tr>
<td>01388</td>
<td>65535</td>
<td>0837</td>
<td>Simplified Chinese</td>
</tr>
<tr>
<td>01390</td>
<td>65535</td>
<td>0300</td>
<td>Japanese</td>
</tr>
<tr>
<td>01399</td>
<td>65535</td>
<td>0300</td>
<td>Japanese</td>
</tr>
<tr>
<td>05123</td>
<td>65535</td>
<td>1027</td>
<td>Japanese</td>
</tr>
<tr>
<td>08482</td>
<td>65535</td>
<td>0290</td>
<td>Japanese</td>
</tr>
</tbody>
</table>

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 on page 123.

Table 21. Base CCSIDs Supported

<table>
<thead>
<tr>
<th>CCSID</th>
<th>Character Set</th>
<th>Code Page</th>
<th>Country/Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>00803</td>
<td>1147</td>
<td>424</td>
<td>Hebrew (Old)</td>
</tr>
<tr>
<td>00931</td>
<td>101</td>
<td>037</td>
<td>Japan (English)</td>
</tr>
<tr>
<td>04369</td>
<td>265</td>
<td>273</td>
<td>Germany and Austria</td>
</tr>
<tr>
<td>04371</td>
<td>273</td>
<td>275</td>
<td>Brazil</td>
</tr>
<tr>
<td>04373</td>
<td>281</td>
<td>277</td>
<td>Denmark and Norway</td>
</tr>
<tr>
<td>04374</td>
<td>285</td>
<td>278</td>
<td>Finland and Sweden</td>
</tr>
<tr>
<td>04376</td>
<td>293</td>
<td>280</td>
<td>Italy</td>
</tr>
<tr>
<td>04380</td>
<td>309</td>
<td>284</td>
<td>L.A. (Spanish Speaking)</td>
</tr>
<tr>
<td>04381</td>
<td>313</td>
<td>285</td>
<td>U.K. English</td>
</tr>
<tr>
<td>04393</td>
<td>1129</td>
<td>297</td>
<td>France</td>
</tr>
<tr>
<td>04934</td>
<td>938</td>
<td>838</td>
<td>Thailand</td>
</tr>
</tbody>
</table>
Table 21. Base CCSIDs Supported (continued)

<table>
<thead>
<tr>
<th>CCSID</th>
<th>Character Set</th>
<th>Code Page</th>
<th>Country/Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>04966</td>
<td>959</td>
<td>870</td>
<td>Latin-2</td>
</tr>
<tr>
<td>04976</td>
<td>960</td>
<td>880</td>
<td>Cyrillic</td>
</tr>
<tr>
<td>05029</td>
<td>933</td>
<td>833</td>
<td>Korean</td>
</tr>
<tr>
<td>05031</td>
<td>936</td>
<td>836</td>
<td>Simplified Chinese</td>
</tr>
<tr>
<td>05033</td>
<td>101</td>
<td>037</td>
<td>Traditional Chinese</td>
</tr>
<tr>
<td>08229</td>
<td>101</td>
<td>037</td>
<td>U.S. English and Netherlands</td>
</tr>
<tr>
<td>08476</td>
<td>650</td>
<td>284</td>
<td>Spain</td>
</tr>
<tr>
<td>09122</td>
<td>332</td>
<td>290</td>
<td>Japan (Katakana)</td>
</tr>
<tr>
<td>41460</td>
<td>904</td>
<td>500</td>
<td>Switzerland</td>
</tr>
<tr>
<td>45556</td>
<td>908</td>
<td>500</td>
<td>Switzerland</td>
</tr>
</tbody>
</table>

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 125) 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 ISPnnnnn, where nnnnn 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 125 and Table 23 on page 125 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>
<thead>
<tr>
<th>Table</th>
<th>Hexadecimal Code</th>
<th>Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>TO_500</td>
<td>DC X'4041244344454647'</td>
<td>(X'40' to X'47')</td>
</tr>
<tr>
<td></td>
<td>DC X'484948494D4D4E4B'</td>
<td>(X'48' to X'4F')</td>
</tr>
<tr>
<td></td>
<td>DC X'5851525354555657'</td>
<td>(X'50' to X'57')</td>
</tr>
<tr>
<td></td>
<td>DC X'58594F5B5C5D5EBA'</td>
<td>(X'58' to X'5F')</td>
</tr>
<tr>
<td></td>
<td>. . .</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DC X'78797A7B7C7D7E7F'</td>
<td>(X'78' to X'7F')</td>
</tr>
<tr>
<td></td>
<td>DC X'808128828384858687'</td>
<td>(X'80' to X'87')</td>
</tr>
<tr>
<td></td>
<td>DC X'E8E9EAEBECEDEEEF'</td>
<td>(X'E8' to X'EF')</td>
</tr>
<tr>
<td></td>
<td>DC X'F0F1F2F3F4F5F6F7'</td>
<td>(X'F0' to X'F7')</td>
</tr>
<tr>
<td></td>
<td>DC X'F8F9F9F9F9F9F9F9F'</td>
<td>(X'F8' to X'FE')</td>
</tr>
</tbody>
</table>

Table 23. Table for translating from CCSID 00500 to CCSID 00037.

<table>
<thead>
<tr>
<th>Table</th>
<th>Hexadecimal Code</th>
<th>Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>FROM_500</td>
<td>DC X'4041244344454647'</td>
<td>(X'40' to X'47')</td>
</tr>
<tr>
<td></td>
<td>DC X'484948494D4D4E4B'</td>
<td>(X'48' to X'4F')</td>
</tr>
<tr>
<td></td>
<td>DC X'5851525354555657'</td>
<td>(X'50' to X'57')</td>
</tr>
<tr>
<td></td>
<td>DC X'58594F5B5C5D5EBA'</td>
<td>(X'58' to X'5F')</td>
</tr>
<tr>
<td></td>
<td>. . .</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DC X'78797A7B7C7D7E7F'</td>
<td>(X'78' to X'7F')</td>
</tr>
<tr>
<td></td>
<td>DC X'808128828384858687'</td>
<td>(X'80' to X'87')</td>
</tr>
<tr>
<td></td>
<td>DC X'E8E9EAEBECEDEEEF'</td>
<td>(X'E8' to X'EF')</td>
</tr>
<tr>
<td></td>
<td>DC X'F0F1F2F3F4F5F6F7'</td>
<td>(X'F0' to X'F7')</td>
</tr>
<tr>
<td></td>
<td>DC X'F8F9F9F9F9F9F9F'</td>
<td>(X'F8' to X'FE')</td>
</tr>
</tbody>
</table>

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 on page 125, which you can update. They are distributed in ISP.SISPSAMP.

Table 24. Translation tables provided with ISPF

<table>
<thead>
<tr>
<th>Table name</th>
<th>CCSID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISPSTC1</td>
<td>00037</td>
<td>U.S.A, Canada, Netherlands, Portugal, Brazil, Australia, and New Zealand</td>
</tr>
<tr>
<td>ISPSTC2</td>
<td>00273</td>
<td>Austria and Germany</td>
</tr>
<tr>
<td>ISPSTC3</td>
<td>00277</td>
<td>Denmark and Norway</td>
</tr>
<tr>
<td>ISPSTC4</td>
<td>00278</td>
<td>Finland and Sweden</td>
</tr>
<tr>
<td>ISPSTC5</td>
<td>00280</td>
<td>Italy</td>
</tr>
<tr>
<td>ISPSTC6</td>
<td>00284</td>
<td>Spain and L.A. (Spanish-speaking)</td>
</tr>
<tr>
<td>ISPSTC7</td>
<td>00285</td>
<td>United Kingdom</td>
</tr>
<tr>
<td>ISPSTC8</td>
<td>00297</td>
<td>France</td>
</tr>
<tr>
<td>ISPSTC9</td>
<td>00500</td>
<td>Switzerland and Belgium</td>
</tr>
<tr>
<td>ISPSTC10</td>
<td>00939</td>
<td>Japan (Latin)</td>
</tr>
<tr>
<td>ISPSTC11</td>
<td>00930</td>
<td>Japan (Katakana)</td>
</tr>
</tbody>
</table>
### Table 24. Translation tables provided with ISPF (continued)

<table>
<thead>
<tr>
<th>Table name</th>
<th>CCSID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISPSTC12</td>
<td>00933</td>
<td>Korea</td>
</tr>
<tr>
<td>ISPSTC13</td>
<td>00935</td>
<td>Simplified Chinese</td>
</tr>
<tr>
<td>ISPSTC14</td>
<td>00937</td>
<td>Traditional Chinese</td>
</tr>
<tr>
<td>ISPSTC15</td>
<td>00870</td>
<td>Latin 2</td>
</tr>
<tr>
<td>ISPSTC16</td>
<td>00880</td>
<td>Cyrillic</td>
</tr>
<tr>
<td>ISPSTC17</td>
<td>01025</td>
<td>Cyrillic</td>
</tr>
</tbody>
</table>

#### 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 xxxx 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" CCSID 00500 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=TRTO500,FROM=TRFR500

* TRTO500
  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=TRTO500,FROM=TRFR500
ISPCCSID CCSID=00333,TO=TRT00333,FROM=TRF00333
ISPCCSID CCSID=00444,TO=TRT00444,FROM=TRF00444

* TRTO500
  DC XL191'... 00222 TO 00500

* TRFR500
  DC XL191'... 00222 FROM 00500 (00500 TO 00222)

* TRT00333
  DC XL191'... 00222 TO 00333

* TRT00444
  DC XL191'... 00222 TO 00444
```

**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 ISPTTCnn, where nn is the last 1-digit or 2-digit number of the ISPSTCnn 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=TTC1T6H,FROM=TTC1F6H
* ISPCCSID CCSID=04371,TO=TTC1T7H,FROM=TTC1F7H

* TTC1T5H - CCSID 00037 TO CCSID 00500 Table
TTC1T5H
  DS 0XL191
  DC X'4041424344454647' (X'40' TO X'47')
  DC X'4849B0B1B2B3B4B5B6B7' (X'B0' TO X'B7')
  DC X'5FB1B2B3B4B5B6B7' (X'B8' TO X'B9')
  DC X'6061626364656667' (X'60' TO X'67')
  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'A8A9AAABBCCDDEEFF' (X'A8' TO X'AE')
  DC X'B0B1B2B3B4B5B6B7' (X'B8' TO X'B9')
  DC X'C0C1C2C3C4C5C6C7' (X'C8' TO X'CF')
  DC X'C8C9CACCDBCEDEE' (X'C8' TO X'CF')
  DC X'D0D1D2D3D4D5D6D7' (X'D0' TO X'D7')
  DC X'D8D9DAD0D1D2D3D4D5D6D7' (X'D8' TO X'DF')
  DC X'E0E1E2E3E4E5E6E7' (X'E0' TO X'E7')
  DC X'F0F1F2F3F4F5F6F7' (X'F0' TO X'F7')
  DC X'F8F9F0F1F2F3F4F5F6F7' (X'F8' TO X'FE')

* TTC1F5H - CCSID 00037 FROM CCSID 00500 Table
TTC1F5H
  DS 0XL191
  DC X'4041424344454647' (X'40' TO X'47')
```

Customizing DM 127
* TTC1TB1 - CCSID 00037 TO CCSID 08229 Table *

TTC1TB1  DS   0XL191

DC X'404B4B4B4B4B4B4B'          (X'40' TO X'47')
DC X'4B4B4B4B4C4D4E4F'          (X'48' TO X'4F')
DC X'5051525354555657'          (X'50' TO X'57')
DC X'58595B5B5C5D5EB0'          (X'5B' 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'4B4B4B4B4B4B4B4B'          (X'B0' TO X'B7')
DC X'C0C1C2C3C4C5C6C7'          (X'C0' TO X'C7')
DC X'C8C9CABCACCDECDF'          (X'C8' TO X'CF')
DC X'D0D1D2D3D4D5D6D7'          (X'D0' TO X'D7')
DC X'D8D9D0D1D2D3D4D5D6D7'     (X'D8' TO X'DF')
DC X'E0E1E2E3E4E5E6E7'          (X'E0' TO X'E7')
DC X'E8E9EAEABECAEDEEF'          (X'E8' TO X'EF')
DC X'F0F1F2F3F4F5F6F7'          (X'F0' TO X'F7')
DC X'F8F9FABFBCFDFDE'          (X'F8' TO X'FE')

* TTC1FB1 - CCSID 00037 FROM CCSID 08229 Table *

TTC1FB1  DS   0XL191

DC X'404A4B4B4B4B4B4B'          (X'40' TO X'47')
DC X'4B4A4B4B4A4D4E4F'          (X'48' TO X'4F')
DC X'5051525354555657'          (X'50' TO X'57')
DC X'58595B5B5C5D5EB0'          (X'5B' 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'B0B1B2B3B4B5B6B7'          (X'B0' TO X'B7')
DC X'C0C1C2C3C4C5C6C7'          (X'C0' TO X'C7')
DC X'C8C9CABCACCDECDF'          (X'C8' TO X'CF')
DC X'D0D1D2D3D4D5D6D7'          (X'D0' TO X'D7')
DC X'D8D9D0D1D2D3D4D5D6D7'     (X'D8' TO X'DF')
DC X'E0E1E2E3E4E5E6E7'          (X'E0' TO X'E7')
DC X'E8E9EAEABECAEDEEF'          (X'E8' TO X'EF')
DC X'F0F1F2F3F4F5F6F7'          (X'F0' TO X'F7')
DC X'F8F9FABFBCFDFDE'          (X'F8' TO X'FE')

* TTC1TB2 - CCSID 00037 TO CCSID 04371 Table *

* TTC1FB1 - CCSID 00037 FROM CCSID 08229 Table *

TTC1FB1  DS   0XL191

DC X'404A4B4B4B4B4B4B'          (X'40' TO X'47')
DC X'4B4A4B4B4A4D4E4F'          (X'48' TO X'4F')
DC X'5051525354555657'          (X'50' TO X'57')
DC X'58595B5B5C5D5EB0'          (X'5B' 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'B0B1B2B3B4B5B6B7'          (X'B0' TO X'B7')
DC X'C0C1C2C3C4C5C6C7'          (X'C0' TO X'C7')
DC X'C8C9CABCACCDECDF'          (X'C8' TO X'CF')
DC X'D0D1D2D3D4D5D6D7'          (X'D0' TO X'D7')
DC X'D8D9D0D1D2D3D4D5D6D7'     (X'D8' TO X'DF')
DC X'E0E1E2E3E4E5E6E7'          (X'E0' TO X'E7')
DC X'E8E9EAEABECAEDEEF'          (X'E8' TO X'EF')
DC X'F0F1F2F3F4F5F6F7'          (X'F0' TO X'F7')
DC X'F8F9FABFBCFDFDE'          (X'F8' TO X'FE')

* TTC1TB2 - CCSID 00037 TO CCSID 04371 Table *

* TTC1FB1 - CCSID 00037 FROM CCSID 08229 Table *

* TTC1TB2 - CCSID 00037 TO CCSID 04371 Table *

128 z/OS: ISPF Planning and Customizing
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.

ISPEXC3 processing

ISPEXC is an external entry point in module ISPLINK. This is how ISPF supports the Call ISPEXC 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:

<table>
<thead>
<tr>
<th>NAME</th>
<th>ISPLINK</th>
<th>ISPLINK</th>
<th>TPUT SVC</th>
</tr>
</thead>
<tbody>
<tr>
<td>VER</td>
<td>00FC</td>
<td>0A5D</td>
<td>TPUT SVC</td>
</tr>
<tr>
<td>REP</td>
<td>00FC</td>
<td>0700</td>
<td>NOP INSTRUCTION</td>
</tr>
</tbody>
</table>

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 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 ISRCONFG 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:
Operand Description

HEADER
Parameter values are as follows:

NOUSRFLG
The number of user flag bytes defined. The value can be a number between 0 and 4, inclusive. The default is 0.

DFUSRFLG
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.

DFFLAG
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.

DFCLRLNS
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.

ENTRY
Parameter values are as follows:

ENTNAME
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.

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.
Command requires authorization check. Set this bit on for a command processor that must be invoked as an authorized command.

Command is not to be logged. Set this bit on if the TSO command buffer should not be written to the ISPLOG data set.

Command is not supported by ISPF. Set this bit on for commands that cannot be invoked under ISPF.

Command is command procedure (CLIST). Set this bit on if this is the name of a CLIST member.

Command is a command processor. Set this bit on if this is the name of a command processor program module.

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.

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.

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.

Note:

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.

Must be last macro call.

An example of how to use the ISPMTCM macro to build ISPTCM is shown in Figure 57 on page 134.
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 137 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 58 on page 135.
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 in the z/OS Internet library (www.ibm.com/servers/resourcelink/svc00100.nsf/pages/zosInternetLibrary) 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” on page 136 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 “DEFAULT_SESSION_LANGUAGE” on page 247). For information about changing Configuration Table settings, see Chapter 2, “The ISPF Configuration Table,” on page 9.
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 on page 137 summarizes the ISPF installation-wide exits. The exit ID is a unique number that identifies that installation-wide exit to ISPF.

<table>
<thead>
<tr>
<th>Exit ID</th>
<th>Installation Exit Name</th>
<th>Possible Uses for the Routine</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ISPF initialization</td>
<td>Provides accounting and monitoring capabilities before ISPF initialization.</td>
</tr>
<tr>
<td>2</td>
<td>ISPF termination</td>
<td>Provides accounting and monitoring capabilities before ISPF termination.</td>
</tr>
<tr>
<td>3</td>
<td>SELECT service start</td>
<td>Provides monitoring information and lets you restrict access to applications selected through ISPF.</td>
</tr>
<tr>
<td>4</td>
<td>SELECT service end</td>
<td>Marks the end of a program, command, or menu invoked through any of the SELECT services.</td>
</tr>
<tr>
<td>5</td>
<td>TSO command start</td>
<td>Provides for monitoring and restricting commands invoked through ISPF; allows commands newly added to the system to be invoked without updating ISPTCM.</td>
</tr>
<tr>
<td>6</td>
<td>TSO command end</td>
<td>Provides for monitoring of TSO commands invoked through ISPF.</td>
</tr>
<tr>
<td>7</td>
<td>LIBDEF service</td>
<td>Provides for restrictions on the use of the LIBDEF service.</td>
</tr>
<tr>
<td>8</td>
<td>RESERVE</td>
<td>Allows use of your own method of serializing resources in addition to the RESERVE done by ISPF.</td>
</tr>
<tr>
<td>9</td>
<td>RELEASE</td>
<td>Provides for the release of any resources acquired at the RESERVE user exit.</td>
</tr>
<tr>
<td>10</td>
<td>Logical screen start</td>
<td>Allows for installation-wide exits to gather accounting and monitoring information for each logical screen.</td>
</tr>
<tr>
<td>11</td>
<td>Logical screen end</td>
<td>Gathers accounting and monitoring information for each logical screen.</td>
</tr>
<tr>
<td>12</td>
<td>ISPF/PDF service start</td>
<td>Monitors ISPF and PDF dialog services invoked through the ISPLINK or ISPEXEC interfaces.</td>
</tr>
<tr>
<td>13</td>
<td>ISPF/PDF service end</td>
<td>Marks the termination of ISPF or PDF dialog services invoked through the ISPLINK or ISPEXEC interfaces.</td>
</tr>
<tr>
<td>14</td>
<td>SWAP logical screens</td>
<td>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.</td>
</tr>
<tr>
<td>15</td>
<td>DISPLAY service start</td>
<td>Provides for tailoring of panels to be displayed.</td>
</tr>
</tbody>
</table>
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 263.

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.

Related references

Chapter 10, “Exits,” on page 263

[End Programming Interface Information]
2. Use the exit macros to assemble the exit definition table (ISPXDT). This process is discussed in detail in “Exit macros” on page 263. These macros reside in ISP.SISPMAC.

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 59 on page 139 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.LOCSLIB,DISP=SHR
//SYSLIN DD *
ORDER ISPXDT
ENTRY ISPXDT
   INCLUDE SYSLIB(ISPXDT)
NAME ISPEXITS(R)
/*

Figure 59. 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 263

### 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. For 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.

   ```
   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.

```
ISPMXED START
ISPMLST (3,7,14)
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. For example:

```
ISPMXED START
ISPMLST (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. For example:

```
ISPMDAD DAREA015,100
```

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.)

```
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 ISPMDAD START and ISPMDAD END.

7. Use the ISPMXDD macro to indicate the end of the exit data area definition section. For example:

```
ISPMDAD DAREA015,100
```

Sample ISPXDT definition

Figure 60 on page 141 shows a sample ISPXDT definition that defines exit routines for installation-wide exits 2 and 12.
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:

**exitid**
A fullword binary number that identifies the numeric code for the user exit.

**userid**
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'.

**screenid**
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.
ZENVIR
32 characters of environmental information that is provided by the ZENVIR ISPF system variable. This variable is described in z/OS ISPF Reference Summary.

datalen
A fullword binary number that identifies the length of the exit data area, in bytes.

dataptr
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.
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 Chapter 2, “The ISPF Configuration Table,” on page 9. You make some of these modifications by editing the panel descriptions in the ISPF panel library (ISPPLIB data set).

All panels in the ISPPLIB data set are in ISPF panel format. All messages in the ISPMLIB 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” on page 143
- “Data set allocation defaults for the Outlist utility” on page 145
- “Using the Hardcopy utility with DBCS support” on page 146
- “SCLM Batch considerations” on page 147
- “Creating PDF translation tables” on page 148
- “PDF Foreground and Batch customizing” on page 149
- “Customizing Browse and Edit” on page 180
- “Customizing member list panels” on page 184
- “IBM Products option” on page 187

See also Chapter 8, “PDF installation-wide exits,” on page 193.

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.

© Copyright IBM Corp. 1984, 2019
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 94.

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 222 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.AAAAxxxx.BACKUP, (where AAAA is the application ID and xxxx is a 4-digit number between 0000 and 9999) if ZPREFIX and ZUSER are different, and ZUSER.AAAAxxxx.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.

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:

   `<pdc unavail=zut7 acc1=alt acc2=8>Outlist`

   to

   `<pdc unavail=zut7 acc1=alt acc2=8>Outlist`

   `<pdc unavail=zut7 acc1=alt acc2=8>`
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:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>LRECL</td>
<td>(133)</td>
</tr>
<tr>
<td>BLKSIZE</td>
<td>(3059)</td>
</tr>
<tr>
<td>Primary</td>
<td>(200)</td>
</tr>
<tr>
<td>Secondary</td>
<td>(100)</td>
</tr>
</tbody>
</table>

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 61 on page 148, the 'ISP' data set high-level qualifier represents your ISPF data sets. xxx corresponds to a national language as follows:

<table>
<thead>
<tr>
<th>Language</th>
<th>xxx</th>
</tr>
</thead>
<tbody>
<tr>
<td>US English</td>
<td>ENU (the default)</td>
</tr>
<tr>
<td>Uppercase English</td>
<td>ENP</td>
</tr>
<tr>
<td>Swiss German</td>
<td>DES</td>
</tr>
<tr>
<td>Japanese</td>
<td>JPN</td>
</tr>
<tr>
<td>German</td>
<td>DEU</td>
</tr>
</tbody>
</table>
/*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

ISPMLIB DD DSN=ISP.SISPMXXX,DISP=SHR ISPF MSGS
    DD DSN=ISP.SISPSXXX,DISP=SHR ISPF SKELS
    DD DSN=ISP.SISPSLIB,DISP=SHR ISPF SKELS

ISPPLIB DD DSN=ISP.SISPPXXX,DISP=SHR ISPF PANELS

ISPTLIB DD UNIT=&VIOUNIT;,DISP=(NEW,PASS),SPACE=(CYL,(1,1,5)),
    DCB=(LRECL=80,BKSIZE=19040,DSORG=PO,RECFM=FB),
    DSN=&TABLESP TEMPORARY TABLE LIBRARY
    DD DSN=ISP.ISPPTXXX,DISP=SHR ISPF TABLES

ISPTABL DD UNIT=&VIOUNIT;,DISP=(NEW,PASS),SPACE=(CYL,(1,1,5)),
    DCB=(LRECL=80,BKSIZE=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,BKSIZE=19040,DSORG=PO,RECFM=FB),
    DSN=&TABLESP TEMPORARY TABLE LIBRARY

ISPLOG DD SYSOUT=*,
    DCB=(LRECL=120,BKSIZE=2400,DSORG=PS,RECFM=FB)

ISPCTL1 DD DISP=NEW,UNIT=VIO,SPACE=(CYL,(1,1)),
    DCB=(LRECL=80,BKSIZE=800,RECFM=FB) TEMPORARY FILE
    DD DSN=&TABLESP TAILORING DATASET

SYSTEM DD SYSOUT=* TAILORING DATASET

SYSPROC DD DSN=&&&&CLIST&STEP,DISP=(OLD,DELETE)
    DD DSN=ISP.SISPCLIB,DISP=SHR CLIST LIBRARY OW01230

*/

Figure 61. Sample FLMLIBS skeleton

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 ISP.SISPSAMP 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
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 281

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. The following code block 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: 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.

```plaintext
)PANEL KEYSLIST(ISRSAB,ISR)  /* ISRFPA - ENGLISH - 5.6 */
)ATTR DEFAULT(...) FORMAT(MIX)
  0B TYPE(AB)
  0D TYPE(PS)
  04 TYPE(ABS) GE(ON)
  05 TYPE(FT)
  09 TYPE(FF)
  0A TYPE(N) SKIP(ON) GE(ON)
  11 TYPE(SAC)
  22 TYPE(WASL) SKIP(ON) GE(ON)
```
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(ISRROUTE) PARM('SET')
PDC DESC('View') UNAVAIL(ZPM2) MNEM(1) ACC(CTRL+V)
ACTION RUN(ISRROUTE) PARM('BRI')
PDC DESC('Edit') UNAVAIL(ZPM3) MNEM(1) ACC(CTRL+E)
ACTION RUN(ISRROUTE) PARM('EDIT')
PDC DESC('Other IBM Products...') UNAVAIL(ZPM4) MNEM(6) ACC(CTRL+L)
ACTION RUN(ISRROUTE) PARM('OIB')
PDC DESC('SCLM') UNAVAIL(ZPM5) MNEM(3) ACC(CTRL+L)
ACTION RUN(ISRROUTE) PARM('SCL')
PDC DESC('ISPF Command Shell') UNAVAIL(ZPM6) MNEM(6) ACC(CTRL+W)
ACTION RUN(ISRROUTE) PARM('WRK')
PDC DESC('ISPF Workplace') UNAVAIL(ZPM7) MNEM(6) 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('Utilities') MNEM(1)
PDC DESC('Library') UNAVAIL(ZUT1) MNEM(1) ACC(ALT+1)
ACTION RUN(ISRROUTE) PARM('U1')
PDC DESC('Move/Copy') UNAVAIL(ZUT2) MNEM(1) ACC(ALT+2)
ACTION RUN(ISRROUTE) PARM('U2')
PDC DESC('Data Set List') UNAVAIL(ZUT4) MNEM(2) ACC(ALT+4)
ACTION RUN(ISRROUTE) PARM('U4')
PDC DESC('Reset Statistics') UNAVAIL(ZUT5) MNEM(5) ACC(ALT+5)
ACTION RUN(ISRROUTE) PARM('U5')
PDC DESC('Hardcopy') UNAVAIL(ZUT6) MNEM(1) ACC(ALT+6)
ACTION RUN(ISRROUTE) PARM('U6')
PDC DESC('Download...') UNAVAIL(ZUT7) MNEM(2) ACC(ALT+7)
ACTION RUN(ISRROUTE) PARM('U7')
PDC DESC('Outlist') UNAVAIL(ZUT8) MNEM(1) ACC(ALT+9)
ACTION RUN(ISRROUTE) PARM('U9')
PDC DESC('Reserved') UNAVAIL(ZUT9) MNEM(6) ACTION RUN(ISRROUTE) PARM('U10')
PDC DESC('Format') UNAVAIL(ZUT10) MNEM(1) ACC(ALT+F)
ACTION RUN(ISRROUTE) PARM('U11')
PDC DESC('Commands...') 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('SuperCE') UNAVAIL(ZUT13) MNEM(2) ACC(CTRL+F4)
ACTION RUN(ISRROUTE) PARM('U14')
PDC DESC('Search-For') 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('ISR40010')
PDC DESC('Assembler') MNEM(1) ACTION RUN(TUTOR) PARM('ISR41000')
PDC DESC('COBOL') MNEM(1) ACTION RUN(TUTOR) PARM('ISR42000')
PDC DESC('VS Fortran') MNEM(1) ACTION RUN(TUTOR) PARM('ISR43000')
PDC DESC('PL/I') MNEM(1) ACTION RUN(TUTOR) PARM('ISR44000')
PDC DESC('Index/Link editor') MNEM(1) ACTION RUN(TUTOR) PARM('ISR45000')
PDC DESC('SCRIPT VS') MNEM(1) ACTION RUN(TUTOR) PARM('ISR46000')
PDC DESC('VS COBOL II debug') MNEM(1) ACTION RUN(TUTOR) PARM('ISR4AA00')
PDC DESC('VS Pascal debug') MNEM(1) ACTION RUN(TUTOR) PARM('ISR49000')
PDC DESC('Member parts list') MNEM(1) ACTION RUN(TUTOR) PARM('ISR4A000')
PDC DESC('C/370') MNEM(3) ACTION RUN(TUTOR) PARM('ISR4B000')
PDC DESC('REXX/370') MNEM(1) ACTION RUN(TUTOR) PARM('ISR44000')
PDC DESC('ADA/370') MNEM(2) ACTION RUN(TUTOR) PARM('ISR45000')
PDC DESC('AD/Cycle C/370') MNEM(5) ACTION RUN(TUTOR) PARM('ISR4A000')
PDC DESC('ISPDLC') MNEM(5) ACTION RUN(TUTOR) PARM('ISR4I000')
PDC DESC('C/390 C/C++') MNEM(5) ACTION RUN(TUTOR) PARM('ISR4J000')
)ABCINIT
.ZVARS=FPAHELP
150 z/OS: ISPF Planning and Customizing
Customizing PDF 151
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. Foreground Assembler H definition (ISRFP01) shows the ISRFP01 option panel definition.

**Note:** In Foreground Assembler H definition (ISRFP01), attribute characters have been replaced by blanks.

**Foreground Assembler H definition (ISRFP01)**

```assembler
)PANEL KEYLIST(ISRSAB,ISR)
)ATTR DEFAULT(...) FORMAT(MIX) /* ISRFP01 - ENGLISH - 5.2 */
08 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('BRI')
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(ZPMS) 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') UNAVAIL(ZPM1) MNEM(1) ACC(CTRL+ALT+P)
ACTION RUN(ISRRROUTE) PARM('PL1')
PDC DESC('Current Library List &ZDSCURLT') MNEM(2) ACC(CTRL+SHIFT+P)
ACTION RUN(ISRRROUTE) PARM('PL2')
PDC DESC('List of Personal Data Set Lists') MNEM(1) PDSEP(ON) ACC(CTRL+ALT+0)
ACTION RUN(ISRRROUTE) PARM('PL1')
PDC DESC('List of Personal Library Lists') MNEM(2) ACC(CTRL+SHIFT+0)
ACTION RUN(ISRRROUTE) PARM('PL2')
)ABCINIT
.ZVARS=REFLOPT
VGET (ZCURTB ZCURLTB) PROFILE
IF (&ZCURTB = &Z) &ZDSCURT = &Z
ELSE &ZDSCURT = '(&ZCURTB)
IF (&ZCURLTB = &Z) &ZDSCURLT = &Z
ELSE &ZDSCURLT = '(&ZCURLTB)
```
&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
))REINIT
REFRESH(PRJ1 LIB1 LIB2 LIB3 LIB4 TYP1 MEM DSN FHAL1 FHAL2 FHAL3)
IF (&ZNXTMSG='ISRT') .CSRPOS = &ZCSRP /* AUTOTYPE */
  .CURSOR = &ZCSRV /* AUTOTYPE */
ELSE &ZXZX = &Z /* AUTOTYPE */
&ZUT6 = 1
&ZUT7 = 1
&ZUT11 = 1
&ZUT12 = 1
&ZUT13 = 1
&ZUT14 = 1
IF (&ZMLCSR ¬= ' ') /* @M1A*/
  .CURSOR = &ZMLCSR /* @M1A*/
)PROC
&ZCSRV = .CURSOR /* AUTOTYPE */
&ZCSRPOS = .CSRPOS /* AUTOTYPE */
&ZODSNLN = 0 /* AUTOPE */
&ZODSNMB = &Z /* AUTOTYPE */
&ZNAME='ZCSRV ZCSRPOS PRJ1 LIB1 LIB2 LIB3 LIB4 TYP1 MEM '
&ZNAME='&ZNAME *.&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 = '%'
&ZNAME='ZCSRV ZCSRPOS * * * * * &ZCSRV&ZODSNLN&ZODSNMB ZCMD'
PANEXIT((ZNAME),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 (ZRDNS) SHARED /* REFERENCE LIST CODE */
IF (&ZRDNS ¬= ' ') /* IF DATA SET SELECTED */
  &DSN = &ZRDNS /* PUT DSN VARIABLE */
&ZRDNS = ' ' /* INTO PANEL */
&ZRVOL = ' ' /* INTO PANEL */
VPUT (ZRDNS ZRVOL) SHARED /* */
  .CURSOR = DSN
  .MSG = ISRDS003 /* MSG PENDING */
VGET (DSALSEL) SHARED /* */
IF (&DSALSEL ¬= ' ') /* IF LIBRARY SELECTED */
VGET (DS1,DS2,DS3,DS4,DS5,DS6,DS7) SHARED
&PRJ1 = &DS1 /* PUT LIBRARY VARIABLES */
&LIB1 = &DS2 /* INTO PANEL */
&LIB2 = &DS3 /* */
&LIB3 = &DS4 /* */
&LIB4 = &DS5 /* */
&TYP1 = &DS6 /* */
&MEM = &DS7 /* */
Table 26 on page 155 lists the names of the ISPF-supplied panels and CLISTS for the Foreground processing option.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Panel ID</th>
<th>CLIST ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>FOREGROUND SELECTION MENU</td>
<td>ISRFPA</td>
<td>--</td>
</tr>
<tr>
<td>1</td>
<td>ASSEMBLER</td>
<td>ISRFP01</td>
<td>ISRFC01</td>
</tr>
<tr>
<td>2</td>
<td>VS COBOL II</td>
<td>ISRFP02</td>
<td>ISRFC02</td>
</tr>
<tr>
<td>3</td>
<td>FORTRAN COMPILE</td>
<td>ISRFP03</td>
<td>ISRFC03</td>
</tr>
<tr>
<td>5</td>
<td>PLI OPTIMIZER COMPILE</td>
<td>ISRFP05</td>
<td>ISRFC05</td>
</tr>
<tr>
<td>6</td>
<td>VS PASCAL COMPILE</td>
<td>ISRFP06</td>
<td>ISRFC06</td>
</tr>
<tr>
<td>7</td>
<td>BINDER/LINK EDIT</td>
<td>ISRFP07</td>
<td>ISRFC07</td>
</tr>
<tr>
<td>9</td>
<td>SCRIPT/VS</td>
<td>ISRFP09</td>
<td>ISRFC09</td>
</tr>
<tr>
<td>10</td>
<td>VS COBOL II DEBUG</td>
<td>--</td>
<td>--</td>
</tr>
</tbody>
</table>
Table 26. ISPF-supplied panels and CLISTs for foreground processing option (continued)

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Panel ID</th>
<th>CLIST ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>10A</td>
<td>COBOL INTERACTIVE DEBUG</td>
<td>ISRFP10A</td>
<td>ISRFC10A</td>
</tr>
<tr>
<td>11</td>
<td>FORTRAN INTERACTIVE DEBUG</td>
<td>ISRFP11</td>
<td>ISRFC11</td>
</tr>
<tr>
<td>12</td>
<td>MEMBER PARTS LIST</td>
<td>ISRFP12</td>
<td>ISRFC12</td>
</tr>
<tr>
<td>13</td>
<td>C/370 COMPILE</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>14</td>
<td>REXX/370 COMPILE</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>15</td>
<td>ADA/370 COMPILE</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>16</td>
<td>AD/CYCLE C/370 COMPILE</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>17</td>
<td>AD/CYCLE C/370 COBOL/370</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>18</td>
<td>ISPDTLC</td>
<td>ISPCP01</td>
<td>--</td>
</tr>
<tr>
<td>19</td>
<td>OS/390 C/C++</td>
<td>--</td>
<td>--</td>
</tr>
</tbody>
</table>

Figure 62 on page 156 shows the overall flow of control for foreground processing.

![Foreground processing flow diagram](image-url)
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 ISPPLIB 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 157. 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 redisplays 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 ISPPLIB 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 AFFFP11 is found in the ISPPLIB 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.

**PRJ1**
- ISPF project name

**LIB1**
- First ISPF library

**TYP1**
- ISPF library type (initially set to nulls in ISRFPR)

**MEM**
- ISPF member name (required if DSN is blank)

**DSN**
- "Other" data set name
ZSEL
Selection string used by ISRFPR to select either a CLIST (CMD) or a program (PGM) with parameters.

LIB2
Second ISPF library

LIB3
Third ISPF library

LIB4
Fourth ISPF library

This field is optional:

PSWD
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.

ZDSQMEM
fully qualified input data set name (without quotes), with the member name in parentheses.

ZDSQ
fully qualified input data set name (without quotes), for the first input data set (without member name)

ZDSQ2
Same as ZDSQ for the second input data set (library)

ZDSQ3
Same as ZDSQ for the third input data set (library)

ZDSQ4
Same as ZDSQ for the fourth input data set (library)

ZDS
Same as ZDSQ without the last qualifier (if ZDSQ has only a one level qualifier, then ZDSQ = ZDS.)

ZDS2
Same as ZDS for the second input data set (library)

ZDS3
Same as ZDS for the third input data set (library)

ZDS4
Same as ZDS for the fourth input data set (library)

ZMEM
Input member name (blank for a sequential data set)

ZORG
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 "ISRIBROB" (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, "S0C1").

**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 157.
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:

```plaintext
)PROC
 &ZSEL = TRANS( TRUNC(&ZCMD,'.'))
 . . .
 10,'PANEL(XYZ)'
```

2. Add new selection panel XYZ to the panel library. Use panel ISRFPA 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:

```plaintext
)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” on page 160, 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, ISRFPA. 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:

```plaintext
)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. The following code shows 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 ISRJPA and ISRJPAB, attribute characters have been replaced by blanks.

Batch selection panel definition (ISRJPA) (Part 1 of 4):

```plaintext
)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(CEF) PADC(USER)
 }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(ZPMS) MNEM(8) ACC(CTRL+T)
   ACTION RUN(ISRROUTE) PARM('DAL')
   PDC DESC('Other IBM Products...') UNAVAIL(ZPM6) MNEM(1) ACC(CTRL+F0)
   ACTION RUN(ISRROUTE) PARM('O1')
   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('Utilities') MNEM(1)
   PDC DESC('Library') UNAVAIL(ZUT1) MNEM(1) ACC(ALT+1)
   ACTION RUN(ISRROUTE) PARM('U1')
   PDC DESC('Data set') UNAVAIL(ZUT2) MNEM(1) ACC(ALT+2)
   ACTION RUN(ISRROUTE) PARM('U2')
   PDC DESC('Move/Copy') UNAVAIL(ZUT3) MNEM(1) ACC(ALT+3)
   ACTION RUN(ISRROUTE) PARM('U3')
   PDC DESC('Data Set List') UNAVAIL(ZUT4) MNEM(2) ACC(ALT+4)
   ACTION RUN(ISRROUTE) PARM('U4')
   PDC DESC('Reset Statistics') UNAVAIL(ZUT5) MNEM(5) ACC(ALT+5)
   ACTION RUN(ISRROUTE) PARM('U5')
   PDC DESC('Hardcopy') UNAVAIL(ZUT6) MNEM(1) ACC(ALT+6)
   ACTION RUN(ISRROUTE) PARM('U6')
 )ABCINIT
  .ZVARS=ISR@OPT
```

Batch selection panel definition (ISRJPA) (Part 2 of 4)

```plaintext
PDC DESC('Download...') UNAVAIL(ZUTDT) MNEM(2) ACC(ALT+7)
 ACTION RUN(ISRROUTE) PARM('UDT')
 PDC DESC('Outlist') UNAVAIL(ZUT7) MNEM(2) ACC(ALT+8)
 ACTION RUN(ISRROUTE) PARM('U8')
 PDC DESC('Commands...') UNAVAIL(ZUT8) MNEM(1) ACC(ALT+9)
 ACTION RUN(ISRROUTE) PARM('U9')
 PDC DESC('Reserved') UNAVAIL(ZUT9) MNEM(6) ACTION RUN(ISRROUTE) PARM('U10')
 PDC DESC('Format') UNAVAIL(ZUT10) MNEM(1) ACC(ALT+F1)
 ACTION RUN(ISRROUTE) PARM('U11')
 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-FoZe') 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('ISR50010')
PDC DESC('Assembler') MNEM(1) ACTION RUN(TUTOR) PARM('ISR51000')
PDC DESC('COBOL') MNEM(1) ACTION RUN(TUTOR) PARM('ISR52000')
PDC DESC('Fortran') MNEM(4) ACTION RUN(TUTOR) PARM('ISR53000')
PDC DESC('PL/I') MNEM(2) ACTION RUN(TUTOR) PARM('ISR54000')
PDC DESC('Member parts list') MNEM(1) ACTION RUN(TUTOR) PARM('ISR55000')
PDC DESC('C/370') MNEM(3) ACTION RUN(TUTOR) PARM('ISR56000')
PDC DESC('REXX/370') MNEM(1) ACTION RUN(TUTOR) PARM('ISR57000')
PDC DESC('Ada') MNEM(2) ACTION RUN(TUTOR) PARM('ISR58000')
PDC DESC('AD/Cycle C/370') MNEM(5) ACTION RUN(TUTOR) PARM('ISR59000')
PDC DESC('ISPDTLC') MNEM(5) ACTION RUN(TUTOR) PARM('ISR5A000')
PDC DESC('OS/390 C/C++') MNEM(1) ACTION RUN(TUTOR) PARM('ISR5B000')
PDC DESC('Appendices') MNEM(4) ACTION RUN(TUTOR) PARM('ISR5C000')

ABCINIT
.ZVARS=JPAHELP
)BODY CMD(ZCMD)
..--------------------------------------------------------------------------
..                           .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 .FORTRAN             ...12.*Member Parts List  ...18.*ISPDTLC
.. 5 .PLI                 ...13.*C/370             ...19.*OS/390 C/C++
.. 6 .VS PASCAL           ...14.*REXX/370
.....
..&multipmt                .*.No packed data support.
...$$..Source data online
...$$..Source data packed
.. .Job Statement Information:.Verify before proceeding.

.. .--> .Z
.. .--> .Z
.. .--> .Z
.. .--> .Z

Batch selection panel definition (ISRJPA) (Part 3 of 4)
**Batch selection panel definition (ISRJP) (Part 4 of 4)**

```plaintext
&ZBPKED = TRUNC(&ZBPKED,1)
&ZPACK = TRANS(TRUNC(&ZPACK,1),Y,YES,N,NO)
&DSN  = ' '                            /* INITIALIZE DATA SET NAME FIELD  */
VPUT (ZDSCHK,ZDSCKO,DSN) SHARED              /* PLACE IN SHARED POOL FOR ISRJB2 */
&TNPNL = 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((ISRJP13) 13) NEWPOOL'
  14, 'PGM(ISRJB2) PARM((ISRJP14) 14) NEWPOOL'
  15, 'PGM(ISRALTDI) PARM(EVGJP15,,ISRJP15,*) NOCHECK'
  16, 'PGM(ISRJB2) PARM((ISRJP16) 16) NEWPOOL'
  18, 'CMD(ISPDTLC (PANEL SUBMIT RETURN)) MODE(FSCR)'
  19, 'PGM(ISRALTDI) PARM(CBC3PE5A,+,ISRJP19,*,ISRJP19A) NOCHECK'
\n\* (**?
)PNTS
FIELD(ZPS01001) VAR(ZCMD) VAL(1)
FIELD(ZPS01002) VAR(ZCMD) VAL(7)
FIELD(ZPS01003) VAR(ZCMD) VAL(10)
FIELD(ZPS01004) VAR(ZCMD) VAL(15)
FIELD(ZPS01005) VAR(ZCMD) VAL(16)
FIELD(ZPS01006) VAR(ZCMD) VAL(17)
FIELD(ZPS01007) VAR(ZCMD) VAL(18)
FIELD(ZPS01008) VAR(ZCMD) VAL(19)
FIELD(ZPS01010) VAR(ZCMD) VAL(2)
FIELD(ZPS01011) VAR(ZCMD) VAL(5)
FIELD(ZPS01012) VAR(ZCMD) VAL(6)
FIELD(ZPS01013) VAR(ZCMD) VAL(12)
FIELD(ZPS01014) VAR(ZCMD) VAL(14)
END
/* 5647-A01 (C) COPYRIGHT IBM CORP 1980, 1999 */
/* ISPDTLC Release: 5.6. Level: PID */
/* z/OS 01.06.00. Created - Date: 19 Nov 2003, Time: 12:37 */
```

Customizing PDF 163
Batch JCL generation panel definition (ISRJPB) (Part 1 of 4)

```plaintext
)PANEL KEYLIST(ISRSAB,ISR)  /* ISRJPB - ENGLISH - 5.6 */
)ATTR DEFAULT(...) FORMAT(MIX)

0B TYPE(AB)
0D TYPE(PS)
04 TYPE(ABSL) GE(ON)
05 TYPE(PT)
09 TYPE(PP)
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(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('CL')
PDC DESC('Dialog Test...') UNAVAIL(ZPMS) MNEM(8) ACC(CTRL+T)
ACTION RUN(ISRROUTE) PARM('DAL')
PDC DESC('Other IBM Products...') UNAVAIL(ZPM6) MNEM(1) ACC(CTRL+D)
ACTION RUN(ISRROUTE) PARM('OIB')
PDC DESC('ISPF Workplace') UNAVAIL(ZPM7) MNEM(3) ACC(CTRL+L)
ACTION RUN(ISRROUTE) PARM('SCL')
PDC DESC('SuperC') UNAVAIL(ZPM8) MNEM(1) ACC(CTRL+X)
ACTION RUN(ISRROUTE) PARM('U12')
PDC DESC('SuperCE') UNAVAIL(ZPM9) MNEM(3) ACC(CTRL+F3)
ACTION RUN(ISRROUTE) PARM('U13')
PDC DESC('Search-For') UNAVAIL(ZUT2) MNEM(2) ACC(CTRL+F4)
ACTION RUN(ISRROUTE) PARM('U14')
PDC DESC('Search-ForE') UNAVAIL(ZUT3) MNEM(4) ACC(CTRL+F5)
ACTION RUN(ISRROUTE) PARM('U15')
ABCINIT
.ZVARS=ISR@OPT
.NEW
)`

Batch JCL generation panel definition (ISRJPB) (Part 2 of 4)

```plaintext
PDC DESC('Utilities') MNEM(1)
PDC DESC('Library') UNAVAIL(ZUT1) MNEM(1) ACC(ALT+1)
ACTION RUN(ISRROUTE) PARM('U1')
PDC DESC('Data set') UNAVAIL(ZUT2) MNEM(1) ACC(ALT+2)
ACTION RUN(ISRROUTE) PARM('U2')
PDC DESC('Move/Copy') UNAVAIL(ZUT3) MNEM(1) ACC(ALT+3)
ACTION RUN(ISRROUTE) PARM('U3')
PDC DESC('Data Set List') UNAVAIL(ZUT4) MNEM(2) ACC(ALT+4)
ACTION RUN(ISRROUTE) PARM('U4')
PDC DESC('Reset Statistics') UNAVAIL(ZUT5) MNEM(5) ACC(ALT+5)
ACTION RUN(ISRROUTE) PARM('U5')
PDC DESC('Hardcopy') UNAVAIL(ZUT6) MNEM(1) ACC(ALT+6)
ACTION RUN(ISRROUTE) PARM('U6')
PDC DESC('Download...') UNAVAIL(ZUT7) MNEM(2) ACC(ALT+7)
ACTION RUN(ISRROUTE) PARM('UDT')
PDC DESC('Outlist') UNAVAIL(ZUT8) MNEM(2) ACC(ALT+8)
ACTION RUN(ISRROUTE) PARM('U8')
PDC DESC('Commands...') UNAVAIL(ZUT9) MNEM(1) ACC(ALT+9)
ACTION RUN(ISRROUTE) PARM('U9')
PDC DESC('Reserved') UNAVAIL(ZUT10) MNEM(1) ACC(ALT+F1)
ACTION RUN(ISRROUTE) PARM('U10')
ABCINIT
.&zut9 = '1'
`
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. The following code shows the Batch Assembler H panel definition.
Note: In ISRJP01, attribute characters have been replaced by blanks.

Batch Assembler H definition (ISRJP01) (Part 1 of 5)

)`PANEL KEYLIST(ISRSAB,ISR) `)ATTR DEFAULT(...) FORMAT(MIX) /* ISRJP01 - ENGLISH - 5.2 */
08 TYPE(AB)
04 TYPE(ABSL) GE(ON)
05 TYPE(PT)
09 TYPE(FP)
0A TYPE(N)
0C TYPE(SAC)
11 TYPE(SAC) SKIP(ON)
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('CL1')
PDC DESC('Dialog Test...') UNAVAIL(ZPMS) 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('O1B')
PDC DESC('SCLM') UNAVAIL(ZPM7) MNEM(3) ACC(CTRL+L)
ACTION RUN(ISRROUTE) PARM('SCL')
PDC DESC('ISPF Workplace') UNAVAIL(ZPMB) 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(ISRROUTE) PARM('PL1')
PDC DESC('Current Library List &ZDSCURLT') MNEM(2) ACC(CTRL+SHIFT+P)
ACTION RUN(ISRROUTE) PARM('LL1')
PDC DESC('List of Personal Data Set Lists') MNEM(1) PDSEP(ON) ACC(CTRL+ALT+0)
ACTION RUN(ISRROUTE) PARM('PL2')
PDC DESC('List of Personal Library Lists') MNEM(2) ACC(CTRL+SHIFT+0)
ACTION RUN(ISRROUTE) PARM('LL2')

Batch Assembler H definition (ISRJP01) (Part 2 of 5)

)`ABCINIT
.ZVARS=ISR@OPT
)ABC DESC('Reflist') MNEM(1)
PDC DESC('Current Data Set List &ZDSCURT') MNEM(1) ACC(CTRL+ALT+P)
ACTION RUN(ISRROUTE) PARM('PL1')
PDC DESC('Current Library List &ZDSCURLT') MNEM(2) ACC(CTRL+SHIFT+P)
ACTION RUN(ISRROUTE) PARM('LL1')
PDC DESC('List of Personal Data Set Lists') MNEM(1) PDSEP(ON) ACC(CTRL+ALT+0)
ACTION RUN(ISRROUTE) PARM('PL2')
PDC DESC('List of Personal Library Lists') MNEM(2) ACC(CTRL+SHIFT+0)
ACTION RUN(ISRROUTE) PARM('LL2')
Batch Assembler H definition (ISRJP01) (Part 3 of 5)

)BODY CMD(ZCMD)
..------------------------------------------------------------------------------
.. Command ===>.Z
.. Command ===>.Z
.. Command ===>.Z
..------------------------------------------------------------------------------
.. Batch Assembler.
.. Project...
.. Group...
.. Type...
.. Member...
.. Data Set Name...
.. Other Partitioned or Sequential Data Set:..
.. List ID...
.. Assembler.
.. SYSOUT class...
.. Assembler options:..
.. Term...
.. Other...
.. Additional input libraries:..
..------------------------------------------------------------------------------
... ===>.Z
... ===>.Z
... ===>.Z
... ===>.Z
.)UNIT
.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 = '        '                        /*                        @M1A*/
&TYP1 = &ASMT                               /*ASSEMBLER TYPE VARIABLE NAME*/
IF (&DSN ¬= ' ')                           /*                        @M1A*/
&MEM = ' '                                /*                        @M1A*/
IF (.CURSOR = ' ') .CURSOR = DSN
IF (ZASMOPT ¬= 1)
   IF (ZASMOPT ¬= 2)
      &ZASMOPT = 1
&ZBTEMP = ''
&HASMT = NOTERM                           /* DEFAULT TO "NOTERM" */
IF (ZBTTEMP = '') &ZBPACK = &ZBPACK

168 z/OS: ISPF Planning and Customizing
Batch Assembler H definition (ISRJP01) (Part 4 of 5)

IF (ZTENMSG = 'ISRT') .CSRPOS = &ZCSR &CURSOR = &ZCSR
/* AUTOTYPE */
ELSE &ZUT6 = &Z
/* AUTOTYPE */
&ZUT7 = 1
&ZUT11 = 1
&ZUT12 = 1
&ZUT13 = 1
&ZUT14 = 1
IF (.MLCSR = ' ') .CSRPOS = &ZMLCSR
/* AUTOTYPE */
IF (.MSG = ISRDS003) REFRESH (PRJ1, LIB1, LIB2, LIB3, LIB4, TYP1, MEM, DSN)
PROC
&ZCSR = .CURSOR /* AUTOTYPE */
&ZCSR = .CSRPOS /* AUTOTYPE */
&ZDSNBLN = 0 /* AUTOTYPE */
&ZODSNMB = &Z /* AUTOTYPE */
&ZNAME='ZCSR ZCSRP PROF1 PRJ1 LIB2 LIB3 LIB4 TYP1 MEM '
&ZNAME='&ZNAME *.&ZDSNBLN &ZODSNMB ZCMD'
IF (.CURSOR = DSN, BHAL1, BHAL2, BHAL3)
&ZDSNBLN = 56
&ZODSNMB = &Z
IF (.CURSOR = BHAL1) &ZDSNBLN = 65
IF (.CURSOR = BHAL2) &ZODSNMB = ' %'
IF (.CURSOR = BHAL3) &ZDSNBLN = 65
&ZNAME='ZCSR ZCSRP ZCMD'
PANEXIT((ZNAME),LOAD,ISRAUTOT) /* AUTOTYPE */
IF (ZTENMSG = 'ISRT') EXIT /* AUTOTYPE */
VER(&ZASMOPT,NONBLANK)
VER(&ZASMOPT RANGE,1,2)
&ZUT6 = 0
&ZUT7 = 0
&ZUT11 = 0
&ZUT12 = 0
&ZUT13 = 0
&ZUT14 = 0
VGET (ZDSN) SHARED /* AUTOTYPE */
IF (&ZDSN = ' ') /* IF DATA SET SELECTED */
&DSN = &ZDSN /* PUT DSN VARIABLE */
&ZDSN = ' ' /* INTO PANEL */
&ZRVOL = ' ' /* INTO PANEL */
VPUT (ZDSN 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 /* */
/* */

Batch Assembler H definition (ISRJP01) (Part 5 of 5)

&LIB3 = &DSA4 /* */
&LIB4 = &DSA5 /* */
&LIB5 = &DSA6 /* */
&LIB6 = &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 */
The names of the ISPF-supplied panels, CLISTs, and skeletons for the Batch processing option are shown in the Table 27 on page 170.

Table 27. ISPF-supplied panels, CLISTs, and skeletons for Batch option

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Panel ID</th>
<th>CLIST ID</th>
<th>SKEL ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>BATCH SELECTION MENU</td>
<td>ISRJPA</td>
<td>-</td>
<td>ISRJSJC</td>
</tr>
<tr>
<td>--</td>
<td>BATCH TERMINATION MENU</td>
<td>ISRJPB</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>1</td>
<td>ASSEMBLER</td>
<td>ISRJP01</td>
<td>ISRJC01</td>
<td>ISRJS01</td>
</tr>
<tr>
<td>2</td>
<td>VS COBOL II</td>
<td>ISRJP02</td>
<td>ISRJC02</td>
<td>ISRJS02</td>
</tr>
<tr>
<td>3</td>
<td>FORTRAN COMPIL</td>
<td>ISRJP03</td>
<td>ISRJC03</td>
<td>ISRJS03</td>
</tr>
<tr>
<td>5</td>
<td>PLI OPTIMIZER COMPIL</td>
<td>ISRJP05</td>
<td>ISRJC05</td>
<td>ISRJS05</td>
</tr>
<tr>
<td>6</td>
<td>VS PASCAL COMPIL</td>
<td>ISRJP06</td>
<td>ISRJC06</td>
<td>ISRJS06</td>
</tr>
<tr>
<td>7</td>
<td>BINDER/LINK EDIT</td>
<td>ISRJP07</td>
<td>ISRJC07</td>
<td>ISRJS07</td>
</tr>
<tr>
<td>10</td>
<td>VS COBOL II DEBUG</td>
<td>ISRJP10</td>
<td>ISRJC10</td>
<td>ISRJS10</td>
</tr>
<tr>
<td>12</td>
<td>MEMBER PARTS LIST</td>
<td>ISRJP12</td>
<td>ISRJC12</td>
<td>ISRJS12</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
<td>Panel ID</td>
<td>CLIST ID</td>
<td>SKEL ID</td>
</tr>
<tr>
<td>--------</td>
<td>-------------------------------------------------</td>
<td>----------</td>
<td>----------</td>
<td>---------</td>
</tr>
<tr>
<td>13</td>
<td>C/370 Compile</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>14</td>
<td>REXX/370 Compile</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>15</td>
<td>ADA/370 Compile</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>16</td>
<td>AD/CYCLE C/370 Compile</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>17</td>
<td>AD/CYCLE C/370 COBOL/370</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>18</td>
<td>ISPDTLC</td>
<td>ISPCP01</td>
<td>--</td>
<td>ISPDTLB</td>
</tr>
<tr>
<td>19</td>
<td>OS/390 C C++</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
</tbody>
</table>

The overall flow of control for Batch processing is shown in Figure 63 on page 171.

![Figure 63. Batch processing flow](image)

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 149. 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 ZDSCKO 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 field, 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 157.

Generated JCL

The job control language (JCL) statements generated by the Batch processing option are accumulated in a temporary data set named user.id.SPFTEMPn.CNTL or prefix.user.id.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:

```plaintext
)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” on page 174.

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:

```plaintext
)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:

```plaintext
)PROC
 &ZSEL = TRANS (&ZCMD
 1,'PGM(ISRJB2) PARM(ISRJP01)'
 2,'PGM(ISPJB2) PARM(ISRJP02)'
 . .
```
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 ISRJP1, 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 ISRLEMX load modules are executed in both foreground and batch.

To submit jobs through the ISPF batch processing option, the load modules ISRSCAN, ISRSUPC, and ISRLEMX 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.SISPSLIB data set. Insert the statement following the EXEC PGM=ISRSCAN, EXEC PGM=ISRLEMX, or EXEC PGM=ISRSUPC statement, whichever is in the skeleton (see the comments in each of these members).

**ISRSCAN and ISRLEMX programs**

The ISPF-supplied Batch skeletons each generate one or more job steps. The first step executes either the ISRSCAN program or the ISRLEMX 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 ISRLEMX copies the primary member and expands any included members as part of the output data set. All members copied by ISRLEMX are unpacked in the output data set and ISRSCAN does not unpack the input while ISRLEMX does. The input to ISRSCAN must not be packed. ISRLEMX also creates a member parts list (see ISPF Options 4.12 and 5.12).

If your input is a sequential data set, ISRLEMX 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:

```plaintext
//*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.
//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.

```plaintext
/*ISRLEMX step
//LEMSTEP 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.
//ISRLXEXPDD 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)
/*
```

**Note:** DDNAME ISRLCODE can be a concatenation of more than 8 data sets, but only the first 8 will be listed in the member parts list.

**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.

**Note:**
1. DDNAME ISRLXEXPDD 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:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Length</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CHAR(3)</td>
<td>Language type of the input member to be processed:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ASM - Assembler</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COB - COBOL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FOR - FORTRAN</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PAS - Pascal</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PLI - PL/I</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SCR - Script</td>
</tr>
<tr>
<td>2</td>
<td>CHAR(8)</td>
<td>Member name of member to be expanded or of first member to be processed to create a member parts list.</td>
</tr>
<tr>
<td>3</td>
<td>CHAR(1)</td>
<td>‘B’ - The request is being run in batch</td>
</tr>
<tr>
<td></td>
<td></td>
<td>‘F’ - The request is being run in foreground</td>
</tr>
<tr>
<td>4</td>
<td>CHAR(1)</td>
<td>‘Y’ - Allocate a temporary sort data set if needed</td>
</tr>
<tr>
<td></td>
<td></td>
<td>‘N’ - Do not allocate a sort data set</td>
</tr>
<tr>
<td>5</td>
<td>CHAR(1)</td>
<td>‘E’ - Expand and unpack the specified member and all included members into one sequential data set</td>
</tr>
<tr>
<td></td>
<td></td>
<td>‘L’ - Create a member parts list starting with the specified member</td>
</tr>
<tr>
<td></td>
<td></td>
<td>blank - Copy the specified input member to a temporary data set, unpacking the member but not expanding. This is the default that is used when a comma is entered to skip this parameter.</td>
</tr>
<tr>
<td>6</td>
<td>CHAR(3)</td>
<td>Number of concatenated input libraries that should be scanned to find the specified input member. The value can be 1 - 255. If the number specified is equal to or larger than the number of concatenated input libraries, all concatenated input libraries are scanned.</td>
</tr>
<tr>
<td>7</td>
<td>CHAR(20)</td>
<td>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.</td>
</tr>
<tr>
<td>8</td>
<td>CHAR(2)</td>
<td>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.</td>
</tr>
</tbody>
</table>
### Table 28. Parameters in the ISRLEMX parameter string (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Length</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>CHAR(3)</td>
<td>Indicates the National language in use:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ENU - English</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DEU - German (Deutsch)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>KAN - Japanese (Kanji)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DES - Swiss German</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>10</td>
<td>CHAR(1)</td>
<td>Position of the month value in the date string; for example, <em>yy/mm/dd</em> (default '4')</td>
</tr>
<tr>
<td>11</td>
<td>CHAR(1)</td>
<td>Position of the day value in the date string; for example, <em>yy/mm/dd</em> (default '7')</td>
</tr>
<tr>
<td>12</td>
<td>CHAR(1)</td>
<td>Position of the year value in the date string; for example, <em>yy/mm/dd</em> (default '1')</td>
</tr>
<tr>
<td>13</td>
<td>CHAR(1)</td>
<td>Delimiter to use in the date string; for example, <em>yy/mm/dd</em> (default '/')</td>
</tr>
<tr>
<td>14</td>
<td>CHAR(8)</td>
<td>Unit name to be used for all temporary data sets used by ISRLEMX. The unit name must be specified.</td>
</tr>
<tr>
<td>15</td>
<td>CHAR(4)</td>
<td>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.</td>
</tr>
</tbody>
</table>

**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 function 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:*

```
ISPEEXEC SELECT PGM(ISRLEMX) +
  PARM('XXX,&ZMEM,F,N,E,4,,00,&ZFPRLANG,&ZFPRMMIX,+
  &ZFPRDDIX,&ZFPRYYIX,&ZFPRDURM,&Z4UNIT')
```
To:

```
ISPEXEC SELECT PGM(ISRLEMX) +
  PARM(‘XXX,&ZMEM,F,N,E,4,trigger,nn,&ZFPRLANG,&ZFPRMMIX,+ &ZFPRLANG,&ZFPRMMIX,&ZFPRDDIX,&ZFPRYYIX,&ZFPRDLIM,&Z4UNIT’)
```

where \( nn \) is the trigger start column.

The CLIST names are as follows:

<table>
<thead>
<tr>
<th>Option</th>
<th>CLIST Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>ISRFC01</td>
</tr>
<tr>
<td>4.2</td>
<td>ISRFC02</td>
</tr>
<tr>
<td>4.3</td>
<td>ISRFC03</td>
</tr>
<tr>
<td>4.5</td>
<td>ISRFC05</td>
</tr>
<tr>
<td>4.6</td>
<td>ISRFC06</td>
</tr>
<tr>
<td>4.9</td>
<td>ISRFC09</td>
</tr>
</tbody>
</table>

**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=ISRLEMX,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:

<table>
<thead>
<tr>
<th>Option</th>
<th>Skeleton Names</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1</td>
<td>ISRJS01I, ISRJS01J, ISRJS01P, ISRJS01S</td>
</tr>
<tr>
<td>5.2</td>
<td>ISRJS02I, ISRJS02J, ISRJS02P, ISRJS02S</td>
</tr>
<tr>
<td>5.3</td>
<td>ISRJS03I, ISRJS03J, ISRJS03P, ISRJS03S</td>
</tr>
<tr>
<td>5.5</td>
<td>ISRJS05I, ISRJS05J, ISRJS05P, ISRJS05S</td>
</tr>
<tr>
<td>5.6</td>
<td>ISRJS06I, ISRJS06J, ISRJS06P, ISRJS06S</td>
</tr>
</tbody>
</table>

**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 ISRFP12 (foreground) and ISRP12 (batch), as follows:

Change from:

```sql
)PROC
&SSPACE = '0100'
```

To:

```sql
)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 64 on page 181). Customized Edit panels must be patterned after these panels:

- ISREFR01 - Edit without action bars or extended highlighting (as shown in Figure 65 on page 182)
- 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 65 on page 182, 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 65 on page 182.

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 - 7.1 */
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)
08 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 = '(ZTITLB ZLINES ZCOLUMNS ZCMD ZSCBR)'
.HELP = ISRBROB000
&ZCMD = ' ')
VGET (ZSCBR) PROFILE /* Fill Scroll Vars if */
IF (&ZSCBR = ' ') &ZSCBR = 'PAGE' /* Blank with Page */
IF (&ZMEMB ^= ' ') &ZTITLB = '&ZDSNT(&ZMEMB)&ZLEVEL' /* OZ91708 */
IF (&ZMEMB = ' ') &ZTITLB = '&ZDSNT&ZLEVEL'
&MIXED = MIX
IF (&ZPDmix = N) &MIXED = EBCDIC
*REXX(ZSCR,ZTITLB)
if length(ztitlb) > 53-9-2
then zscr = 'ON'
else zscr = 'OFF'
+ENDREXX
)REINIT
REFRESH(ZCMD,ZSCBR,ZDATA,ZLINES,ZCOLUMNS,ZTITLB)
)PROC
&ZCURSOR = .CURSOR
&ZCSROFF = .CSRPOS
VPUT (ZSCBR) PROFILE /* */
&ZLVLINE = LVLLINE(ZDATA)
)FIELD
FIELD(ZTITLB) SCROLL(ZSCR)
)END
/* 5694-A01 COPYRIGHT IBM CORP 1995, 2011 */
/* ISPDOLC Release: 7.1. Level: PID */
```

Figure 64. Browse panel (ISRBROB)
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.

---

**Figure 65. Edit panel (ISREFR01)**

---
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 29 on page 183 and Table 30 on page 183 list the ISPF-provided output-only variables you can use on Browse and Edit panels.

### Table 29. Browse output-only variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th>Format/Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZCOLUMNS</td>
<td>First and last columns being displayed</td>
<td>CHAR 7</td>
</tr>
<tr>
<td>ZCSROFF</td>
<td>Cursor offset within cursor field</td>
<td>CHAR 4</td>
</tr>
<tr>
<td>ZCURSOR</td>
<td>Name of field where cursor is placed</td>
<td>CHAR 8</td>
</tr>
<tr>
<td>ZDADWD</td>
<td>Width of the dynamic area (ZDATA)</td>
<td>CHAR 4</td>
</tr>
<tr>
<td>ZDAMLN</td>
<td>Length of the dynamic area (ZDATA)</td>
<td>CHAR 4</td>
</tr>
<tr>
<td>ZDSN</td>
<td>Name of data set being displayed</td>
<td>CHAR 44</td>
</tr>
<tr>
<td>ZDSNT</td>
<td>Name of data set or file being displayed</td>
<td>CHAR 1023</td>
</tr>
<tr>
<td>ZLEVEL</td>
<td>Version and mod level of member</td>
<td>CHAR 8</td>
</tr>
<tr>
<td>ZLINES</td>
<td>Top line of the data display</td>
<td>CHAR 8</td>
</tr>
<tr>
<td>ZLVLINE</td>
<td>Last visible line of ZDATA after last interaction</td>
<td>CHAR 4</td>
</tr>
<tr>
<td>ZMEMB</td>
<td>Name of member being displayed</td>
<td>CHAR 8</td>
</tr>
<tr>
<td>ZWIDTH</td>
<td>Width of the panel</td>
<td>CHAR 4</td>
</tr>
<tr>
<td>ZDSVOL</td>
<td>Volume of the data set or first library in the concatenation</td>
<td>CHAR 6</td>
</tr>
</tbody>
</table>

### Table 30. Edit output-only variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th>Format/Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZCL</td>
<td>Left column of the data display</td>
<td>CHAR 5</td>
</tr>
</tbody>
</table>
### 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 Unicode Services User’s Guide and Reference](https://publib.boulder.ibm.com/infocenter/iseries/v2r13m1/index.jsp?topic=/com.ibm.iseries.doc.iseriesispf.doc/ihdgsunug.html).

### 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 66 on page 185 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](https://publib.boulder.ibm.com/infocenter/iseries/v2r13m1/index.jsp?topic=/com.ibm.iseries.doc.iseriesispf.doc/ihdgsdfdsug.html) for a discussion of these needs.

---

### Table 30. Edit output-only variables (continued)

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th>Format/Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZCR</td>
<td>Right column of the data display</td>
<td>CHAR 5</td>
</tr>
<tr>
<td>ZCSROFF</td>
<td>Cursor offset within cursor field</td>
<td>CHAR 4</td>
</tr>
<tr>
<td>ZCURSOR</td>
<td>Name of field where cursor is placed</td>
<td>CHAR 8</td>
</tr>
<tr>
<td>ZDADLN</td>
<td>Length of the dynamic area (ZDATA)</td>
<td>CHAR 4</td>
</tr>
<tr>
<td>ZDADWD</td>
<td>Width of the dynamic area (ZDATA)</td>
<td>CHAR 4</td>
</tr>
<tr>
<td>ZDSN</td>
<td>Name of data set being displayed</td>
<td>CHAR 44</td>
</tr>
<tr>
<td>ZLEVEL</td>
<td>Version and mod level of member</td>
<td>CHAR 8</td>
</tr>
<tr>
<td>ZLVLINE</td>
<td>Last visible line of ZDATA after last interaction</td>
<td>CHAR 4</td>
</tr>
<tr>
<td>ZMemb</td>
<td>Name of member being displayed</td>
<td>CHAR 8</td>
</tr>
<tr>
<td>ZTITLE</td>
<td>The title line DSN(MEMB) -VER.MOD</td>
<td>CHAR 1023</td>
</tr>
<tr>
<td>ZWIDTH</td>
<td>Width of the panel</td>
<td>CHAR 4</td>
</tr>
<tr>
<td>ZDSVOL</td>
<td>Volume of the data set or first library in the concatenation.</td>
<td>CHAR 6</td>
</tr>
</tbody>
</table>
Figure 66. Customized member list panel (Part 1 of 3)

Customized member list panel (Part 2 of 3)
Customized member list panel (Part 3 of 3)

```
. ZVARS=MEMLHELP
)BODY CMD(ZCMD)
## Menu# Functions# Utilities# Help#
#-------------------------------------------# # .-------------------------------------------# # # #
# MEMBER LIST #Z                 #Row#Z   #of#Z #
#Command ===>#Z                                               #Scroll ===>#Z   #
#ZMLCOLD                                                                       #
#ZDATA                                                                         #
#                                                                              #
)INIT
.ZVARS = '(ZDSN ZMLCR ZMLTR ZCMD ZSCML)'
.HELP = ISR01130
.ATTR(ZMLCR)='JUST(RIGHT) PAD(0)'
.ATTR(ZMLTR)='JUST(RIGHT) PAD(0)'
&zds = &zdsn
&zscr = 'OFF'
&zt = LENGTH(zds)
IF (&zt > 42)
&zscr = 'ON'
VGET (MLC1 MLC2 MLC3 MLH1 MLH2 MLH3 ZMLPD ZSCML) PROFILE
VGET (ZGUI)
&ZMLPAD = 'USER'
IF (&ZGUI = ' ')
/ * Init to user pad char OW18007*/
IF (&ZMLPD = ' ')
/ * User pad not wanted OW18007*/
&ZMLPAD = ' ',
ELSE
/ * GUI mode OW18007*/
IF (&ZGUI NE 'CLIENT')
/ * Not for Web */
.ATTRCHAR(01) = 'TYPE(DATAOUT)'
/ * Set to output OW18007*/
IF (&MLC1 = ' ') &MLC1 = 'TURQ'
IF (&MLC2 = ' ') &MLC2 = 'BLUE'
IF (&MLC3 = ' ') &MLC3 = 'GREEN'
IF (&ZSCML = ' ') &ZSCML = 'PAGE'
/ * Blank with page. */
)PROC
VPUT (ZSCML) PROFILE
IF (.CURSOR = ZDATA OR .CURSOR = ZMLCOLD) &ZMSCPOS = &ZCURPOS
ELSE &ZMSCPOS = '0000'
)FIELD
FIELD(ZDSN) SCROLL(ZSCR)
)PNTS
)END
/* 5650-ZOS     COPYRIGHT IBM CORP 1994, 2013 */
/* ISPDTLC Release: 7.1.  Level: PID                                  */
/* z/OS 02.01.00.  Created - Date: 1 Mar 2013, Time: 11:47            */
```
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 180 for a more complete discussion on customizing panels.

Table 31 on page 187 lists the ISPF-provided output-only variables you can use on member list panels.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th>Format/Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZDAZWD</td>
<td>Width of dynamic area (ZDATA)</td>
<td>CHAR 4</td>
</tr>
<tr>
<td>ZDAMLN</td>
<td>Length of dynamic area (ZDATA)</td>
<td>CHAR 4</td>
</tr>
<tr>
<td>ZDSN</td>
<td>First or only data set in concatenation</td>
<td>CHAR 44</td>
</tr>
<tr>
<td>ZMLCOLS</td>
<td>Member statistics column headings</td>
<td>CHAR 80</td>
</tr>
<tr>
<td>ZMLCR</td>
<td>Relative row number of top row</td>
<td>FIXED 4</td>
</tr>
<tr>
<td>ZMLTR</td>
<td>Total number of rows in member list</td>
<td>FIXED 4</td>
</tr>
<tr>
<td>ZDSVOL</td>
<td>Volume of the data set or first library in the concatenation.</td>
<td>CHAR 6</td>
</tr>
</tbody>
</table>

**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:

**APPLICATION ID**

- 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:

**APPLICATION ID**

- 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 |
Chapter 8. PDF installation-wide exits

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 99 for an example of a data set list filter exit.

### Analyze Dsname and Volume

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Len</th>
<th>Description</th>
<th>Modifiable</th>
</tr>
</thead>
<tbody>
<tr>
<td>CODE</td>
<td>FIXED</td>
<td>4</td>
<td>Code that indicates that this is the analyze dsname and volume call. Call=1.</td>
<td>No</td>
</tr>
<tr>
<td>DSNAME LEVEL</td>
<td>CHAR</td>
<td>44</td>
<td>Data set name level that was specified on the option 3.4 panel.</td>
<td>Yes</td>
</tr>
<tr>
<td>VOLUME</td>
<td>CHAR</td>
<td>6</td>
<td>Volume that was specified on the option 3.4 panel.</td>
<td>Yes</td>
</tr>
</tbody>
</table>

### Analyze data set name
Table 33. Details of the parameters used to analyze the data set name

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Len</th>
<th>Description</th>
<th>Modifiable</th>
</tr>
</thead>
<tbody>
<tr>
<td>CODE</td>
<td>FIXED</td>
<td>4</td>
<td>Code that indicates that this is the analyze data set</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>name call. Call=2</td>
<td></td>
</tr>
<tr>
<td>DSNAME LEVEL</td>
<td>CHAR</td>
<td>44</td>
<td>Data set name level that you specified on the option</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3.4 panel.</td>
<td></td>
</tr>
<tr>
<td>VOLUME</td>
<td>CHAR</td>
<td>6</td>
<td>Volume that you specified on the option 3.4 panel.</td>
<td>No</td>
</tr>
<tr>
<td>DATA SET NAME</td>
<td>CHAR</td>
<td>44</td>
<td>Data set name that is to be added to the list.</td>
<td>No</td>
</tr>
<tr>
<td>VOLUME NAME</td>
<td>CHAR</td>
<td>6</td>
<td>Volume on which this data set resides.</td>
<td>No</td>
</tr>
</tbody>
</table>

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 34 on page 195.

### Table 34. Parameter list during allocation

<table>
<thead>
<tr>
<th>Key</th>
<th>#Parms</th>
<th>Length</th>
<th>Parameter</th>
<th>Description</th>
<th>Exit may modify these parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>0001</td>
<td>1</td>
<td>8</td>
<td>DDNAME</td>
<td>ddname to allocate data set to</td>
<td>No</td>
</tr>
<tr>
<td>0002</td>
<td>1</td>
<td>44</td>
<td>DSNAME</td>
<td>Data set name to allocate</td>
<td>No</td>
</tr>
<tr>
<td>0004</td>
<td>1</td>
<td>1</td>
<td>STATUS</td>
<td>Data set status</td>
<td>Yes</td>
</tr>
<tr>
<td>0005</td>
<td>1</td>
<td>1</td>
<td>NDISP</td>
<td>Normal data set disposition</td>
<td>Yes</td>
</tr>
<tr>
<td>0006</td>
<td>1</td>
<td>1</td>
<td>CDISP</td>
<td>Conditional data set disposition</td>
<td>Yes</td>
</tr>
<tr>
<td>0010</td>
<td>1</td>
<td>6</td>
<td>VLSER</td>
<td>Volume serial</td>
<td>No</td>
</tr>
<tr>
<td>0015</td>
<td>1</td>
<td>8</td>
<td>UNIT</td>
<td>unit name</td>
<td>No</td>
</tr>
<tr>
<td>0050</td>
<td>1</td>
<td>8</td>
<td>PASSW</td>
<td>Password for protected data set if a password was specified</td>
<td>No</td>
</tr>
<tr>
<td>0057</td>
<td>1</td>
<td>8</td>
<td>RTORG</td>
<td>Return data set organization</td>
<td>No</td>
</tr>
<tr>
<td>005D</td>
<td>1</td>
<td>6</td>
<td>RTVOL</td>
<td>Return volume serial from allocation</td>
<td>No</td>
</tr>
<tr>
<td>8017</td>
<td>1</td>
<td>1023 (max)</td>
<td>PATHNAME</td>
<td>Path name of the z/OS UNIX file to allocate. Pathname is of the form: /dev/fdnnn</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>where nnn is the file descriptor number. The real path name can be obtained via the path name pointer and path name length parameters.</td>
<td></td>
</tr>
<tr>
<td>8018</td>
<td>1</td>
<td>4</td>
<td>PATHOPT</td>
<td>The file options for the z/OS UNIX file</td>
<td>No</td>
</tr>
<tr>
<td>801D</td>
<td>1</td>
<td>1</td>
<td>FILEORG</td>
<td>The organization of the z/OS UNIX file</td>
<td>No</td>
</tr>
</tbody>
</table>

If the request is to concatenate a data set, the parameter list contains the information in Table 35 on page 196.
Table 35. Parameter list during concatenation

<table>
<thead>
<tr>
<th>Key</th>
<th>#Parms</th>
<th>Length</th>
<th>Parameter</th>
<th>Description</th>
<th>Exit may modify these parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>0001</td>
<td>1</td>
<td>8</td>
<td>DDNAME</td>
<td>List of dnames corresponding to the data sets being concatenated</td>
<td>No</td>
</tr>
<tr>
<td>0004</td>
<td>0</td>
<td>0</td>
<td>PERMCC</td>
<td>Permanently concatenated attribute</td>
<td>No</td>
</tr>
</tbody>
</table>

If the request is to create a data set, the parameter list contains the information in Table 36 on page 196.

Table 36. Create data set allocation parameter list

<table>
<thead>
<tr>
<th>Key</th>
<th>#Parms</th>
<th>Length</th>
<th>Parameter</th>
<th>Description</th>
<th>Modifiable?</th>
</tr>
</thead>
<tbody>
<tr>
<td>0001</td>
<td>1</td>
<td>8</td>
<td>DDNAME</td>
<td>ddname to allocate data set to</td>
<td>No</td>
</tr>
<tr>
<td>0002</td>
<td>1</td>
<td>44</td>
<td>DSNAME</td>
<td>Data set name to allocate</td>
<td>No</td>
</tr>
<tr>
<td>0004</td>
<td>1</td>
<td>1</td>
<td>STATUS</td>
<td>Data set status</td>
<td>Yes</td>
</tr>
<tr>
<td>0005</td>
<td>1</td>
<td>1</td>
<td>NDISP</td>
<td>Normal data set disposition</td>
<td>Yes</td>
</tr>
<tr>
<td>0006</td>
<td>1</td>
<td>1</td>
<td>CDISP</td>
<td>Conditional data set disposition</td>
<td>Yes</td>
</tr>
<tr>
<td>0007</td>
<td>1</td>
<td>0</td>
<td>TRKS</td>
<td>Space allocated in tracks or Space allocated in cylinders, or Space allocated in blocks</td>
<td>Yes</td>
</tr>
<tr>
<td>0008</td>
<td>1</td>
<td>0</td>
<td>CYLS</td>
<td>Space allocated in tracks or Space allocated in cylinders, or Space allocated in blocks</td>
<td>Yes</td>
</tr>
<tr>
<td>0009</td>
<td>1</td>
<td>3</td>
<td>BLKS</td>
<td>Space allocated in tracks or Space allocated in cylinders, or Space allocated in blocks</td>
<td>Yes</td>
</tr>
<tr>
<td>000A</td>
<td>1</td>
<td>3</td>
<td>PSPACE</td>
<td>Primary space quantity</td>
<td>Yes</td>
</tr>
<tr>
<td>000B</td>
<td>1</td>
<td>3</td>
<td>SSPACE</td>
<td>Secondary space quantity</td>
<td>Yes</td>
</tr>
<tr>
<td>000C</td>
<td>1</td>
<td>3</td>
<td>DBLKS</td>
<td>Number of directory blocks</td>
<td>Yes</td>
</tr>
<tr>
<td>0010</td>
<td>1</td>
<td>6</td>
<td>VOLSER</td>
<td>Volume serial</td>
<td>Yes</td>
</tr>
<tr>
<td>0015</td>
<td>1</td>
<td>8</td>
<td>UNIT</td>
<td>Unit group (esoteric) name Device type Specific unit address</td>
<td>Yes</td>
</tr>
<tr>
<td>0020</td>
<td>1</td>
<td>1</td>
<td>PASPR</td>
<td>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.</td>
<td>Yes</td>
</tr>
<tr>
<td>0022</td>
<td>1</td>
<td>5</td>
<td>EXPDT</td>
<td>Expiration date - YYDDD or Expiration date - YYYYDDD or Retention period</td>
<td>Yes</td>
</tr>
<tr>
<td>0023</td>
<td>1</td>
<td>2</td>
<td>RETPD</td>
<td>Expiration date - YYDDD or Expiration date - YYYYDDD or Retention period</td>
<td>Yes</td>
</tr>
<tr>
<td>0030</td>
<td>1</td>
<td>2</td>
<td>BLKSZ</td>
<td>Block size</td>
<td>Yes</td>
</tr>
<tr>
<td>003C</td>
<td>1</td>
<td>2</td>
<td>DSORG</td>
<td>Data set organization</td>
<td>Yes</td>
</tr>
<tr>
<td>0042</td>
<td>1</td>
<td>2</td>
<td>LRECL</td>
<td>Logical record length</td>
<td>Yes</td>
</tr>
<tr>
<td>0049</td>
<td>1</td>
<td>1</td>
<td>RECFM</td>
<td>Record format</td>
<td>Yes</td>
</tr>
<tr>
<td>0050</td>
<td>1</td>
<td>8</td>
<td>PASSWORD</td>
<td>Data set password if specified</td>
<td>Yes</td>
</tr>
</tbody>
</table>
### Table 36. Create data set allocation parameter list (continued)

<table>
<thead>
<tr>
<th>Key</th>
<th>#Parms</th>
<th>Length</th>
<th>Parameter</th>
<th>Description</th>
<th>Modifiable?</th>
</tr>
</thead>
<tbody>
<tr>
<td>0052</td>
<td>0</td>
<td>0</td>
<td>PERM</td>
<td>Permanently allocated attribute</td>
<td>No</td>
</tr>
<tr>
<td>0057</td>
<td>1</td>
<td>2</td>
<td>RTORG</td>
<td>Return data set organization</td>
<td>No</td>
</tr>
<tr>
<td>005D</td>
<td>1</td>
<td>6</td>
<td>RTVOL</td>
<td>Return volume serial from allocation</td>
<td>No</td>
</tr>
<tr>
<td>8004</td>
<td>1</td>
<td>8</td>
<td>STORCLAS</td>
<td>Storage class used to allocate the data set</td>
<td>Yes</td>
</tr>
<tr>
<td>8005</td>
<td>1</td>
<td>8</td>
<td>MGMTCLAS</td>
<td>Management class used to allocate the data set</td>
<td>Yes</td>
</tr>
<tr>
<td>8006</td>
<td>1</td>
<td>8</td>
<td>DATACLAS</td>
<td>Data class used to allocate the data set</td>
<td>Yes</td>
</tr>
<tr>
<td>8010</td>
<td>1</td>
<td>1</td>
<td>AVGREC</td>
<td>Unit of allocation in terms of average record size:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X'80' Bytes</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X'40' Kilobytes</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X'20' Megabytes</td>
<td></td>
</tr>
<tr>
<td>8012</td>
<td>1</td>
<td>1</td>
<td>DSNTYPE</td>
<td>Data set name type used to allocate the data set:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X'80' Library</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X'40' PDS</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X'10' HFS</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X'08' Extreq</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X'04' Extpref</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X'02' Basic</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X'01' Large</td>
<td></td>
</tr>
</tbody>
</table>

If the request is to deallocate a data set, the parameter list contains the information in Table 37 on page 197.

### Table 37. Parameter list during deallocation

<table>
<thead>
<tr>
<th>Key</th>
<th>#Parms</th>
<th>Length</th>
<th>Parameter</th>
<th>Description</th>
<th>Exit may modify these parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>0001</td>
<td>1</td>
<td>8</td>
<td>DDNAME</td>
<td>ddname to free</td>
<td>No</td>
</tr>
<tr>
<td>0007</td>
<td>0</td>
<td>0</td>
<td>UNALC</td>
<td>Unallocate option</td>
<td>No</td>
</tr>
</tbody>
</table>

If the request is to delete a data set, the parameter list contains the information in Table 38 on page 197.

### Table 38. Parameter list during deletion

<table>
<thead>
<tr>
<th>Key</th>
<th>#Parms</th>
<th>Length</th>
<th>Parameter</th>
<th>Description</th>
<th>Exit may modify these parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>0002</td>
<td>1</td>
<td>44</td>
<td>DSNAME</td>
<td>DSNAME to free</td>
<td>No</td>
</tr>
</tbody>
</table>
Table 38. Parameter list during deletion (continued)

<table>
<thead>
<tr>
<th>Key</th>
<th>#Parms</th>
<th>Length</th>
<th>Parameter</th>
<th>Description</th>
<th>Exit may modify these parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>0007</td>
<td>0</td>
<td>0</td>
<td>UNALL</td>
<td>Unallocate option</td>
<td>No</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Len</th>
<th>Description</th>
<th>Modifiable</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXIT ID</td>
<td>FIXED</td>
<td>4</td>
<td>Numeric code for the exit point.</td>
<td>No</td>
</tr>
<tr>
<td>COMMAND/ SUB-FUNCTION ADDRESS</td>
<td>PTR</td>
<td>4</td>
<td>Storage location containing the command name or sub-function identifier.</td>
<td>No</td>
</tr>
<tr>
<td>COMMAND/ SUB-FUNCTION LENGTHS</td>
<td>FIXED</td>
<td>4</td>
<td>Length of the command name or sub-function identifier. The value ranges from 1-255.</td>
<td>No</td>
</tr>
</tbody>
</table>

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 control 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:

A
- Allocate data set
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:
<table>
<thead>
<tr>
<th>Identifier</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>Browse data set</td>
</tr>
<tr>
<td>C</td>
<td>Catalog data set</td>
</tr>
<tr>
<td>D</td>
<td>Delete data set</td>
</tr>
<tr>
<td>E</td>
<td>Edit data set</td>
</tr>
<tr>
<td>F</td>
<td>Free unused space</td>
</tr>
<tr>
<td>I</td>
<td>Data set information</td>
</tr>
<tr>
<td>M</td>
<td>Display member list</td>
</tr>
<tr>
<td>MB</td>
<td>Browse member</td>
</tr>
<tr>
<td>MD</td>
<td>Delete member</td>
</tr>
<tr>
<td>ME</td>
<td>Edit member</td>
</tr>
<tr>
<td>MI</td>
<td>Info member</td>
</tr>
<tr>
<td>MP</td>
<td>Print member</td>
</tr>
<tr>
<td>MR</td>
<td>Rename member</td>
</tr>
<tr>
<td>P</td>
<td>Print entire data set</td>
</tr>
<tr>
<td>R</td>
<td>Rename data set</td>
</tr>
<tr>
<td>S</td>
<td>Data set information (short)</td>
</tr>
<tr>
<td>U</td>
<td>Uncatalog data set</td>
</tr>
<tr>
<td>X</td>
<td>Print data set index listing</td>
</tr>
<tr>
<td>Z</td>
<td>Compress data set</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Len</th>
<th>Description</th>
<th>Modifiable</th>
</tr>
</thead>
<tbody>
<tr>
<td>LINE COMMAND</td>
<td>CHAR</td>
<td>255</td>
<td>A 255-line character field containing the data set list line command. The field is left-justified with trailing blanks.</td>
<td>Yes</td>
</tr>
</tbody>
</table>

### 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 “Print utility exit on ISPF termination and LOG/LIST commands” on page 206).

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 206. 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 42 on page 205 are available to the print exit routine. However, if variable ZPRLGLST has a value of LOG or LIST, then the variables shown in Table 43 on page 207 are available to the print exit routine.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Len</th>
<th>Description</th>
<th>Modifiable</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZPRDSN</td>
<td>CHAR</td>
<td>44</td>
<td>Fully qualified data set name (no quotes) of the data set to be printed</td>
<td>No</td>
</tr>
<tr>
<td>ZPRMEM</td>
<td>CHAR</td>
<td>8</td>
<td>Member to be printed, if the data set is a partitioned data set and a member is defined</td>
<td>No</td>
</tr>
<tr>
<td>ZPRPMD</td>
<td>CHAR</td>
<td>5</td>
<td>Print mode - BATCH, LOCAL, or TSO/E ICF</td>
<td>Yes</td>
</tr>
<tr>
<td>ZPROPT</td>
<td>CHAR</td>
<td>1</td>
<td>Disposition of the data set as specified by the user - K (keep) or D (delete)</td>
<td>Yes</td>
</tr>
<tr>
<td>ZPRDSORG</td>
<td>CHAR</td>
<td>8</td>
<td>Organization of the data set - PO or PS</td>
<td>No</td>
</tr>
<tr>
<td>ZPRRECFM</td>
<td>CHAR</td>
<td>6</td>
<td>Record format of the data set</td>
<td>No</td>
</tr>
<tr>
<td>ZPRLRECL</td>
<td>CHAR</td>
<td>5</td>
<td>Logical record length of the data set</td>
<td>No</td>
</tr>
<tr>
<td>ZPRBLKSZ</td>
<td>CHAR</td>
<td>5</td>
<td>Physical block size of the data set</td>
<td>No</td>
</tr>
<tr>
<td>ZPRVOLSE</td>
<td>CHAR</td>
<td>6</td>
<td>Volume serial of the data set</td>
<td>No</td>
</tr>
<tr>
<td>ZPRPASS</td>
<td>CHAR</td>
<td>8</td>
<td>Data set password (if it is password protected)</td>
<td>No</td>
</tr>
<tr>
<td>ZPRSYSO</td>
<td>CHAR</td>
<td>80</td>
<td>Batch SYSOUT class as specified by the user</td>
<td>Yes</td>
</tr>
<tr>
<td>ZPRSYSOL</td>
<td>CHAR</td>
<td>80</td>
<td>Local SYSOUT class as specified by the user</td>
<td>Yes</td>
</tr>
<tr>
<td>ZPRLPRT</td>
<td>CHAR</td>
<td>8</td>
<td>Local printer ID as specified by the user</td>
<td>Yes</td>
</tr>
<tr>
<td>ZRICFPL</td>
<td>CHAR</td>
<td>15</td>
<td>TSO/E Information Center Facility printer location as specified by the user</td>
<td>Yes</td>
</tr>
<tr>
<td>ZRICFPF</td>
<td>CHAR</td>
<td>8</td>
<td>TSO/E Information Center Facility printer format as specified by the user</td>
<td>Yes</td>
</tr>
<tr>
<td>ZRICFPT</td>
<td>CHAR</td>
<td>8</td>
<td>TSO/E Information Center Facility printer type as specified by the user</td>
<td>Yes</td>
</tr>
<tr>
<td>ZRICFNC</td>
<td>CHAR</td>
<td>3</td>
<td>TSO/E Information Center Facility number of copies as specified by the user</td>
<td>Yes</td>
</tr>
<tr>
<td>ZRICFEF</td>
<td>CHAR</td>
<td>1</td>
<td>TSO/E Information Center Facility enable font selection</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Table 42. Variables available to the print exit routine. (continued)

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Len</th>
<th>Description</th>
<th>Modifiable</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZPRJB1</td>
<td>CHAR</td>
<td>72</td>
<td>Print job card 1 as specified by the user</td>
<td>Yes</td>
</tr>
<tr>
<td>ZPRJB2</td>
<td>CHAR</td>
<td>72</td>
<td>Print job card 2 as specified by the user</td>
<td>Yes</td>
</tr>
<tr>
<td>ZPRJB3</td>
<td>CHAR</td>
<td>72</td>
<td>Print job card 3 as specified by the user</td>
<td>Yes</td>
</tr>
<tr>
<td>ZPRJB4</td>
<td>CHAR</td>
<td>72</td>
<td>Print job card 4 as specified by the user</td>
<td>Yes</td>
</tr>
</tbody>
</table>

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 cannot 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 204. 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 43 on page 207 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 43 on page 207 are available to the print exit routine. However, if variable ZPRLGLST is not available or blank, then the variables shown in Table 42 on page 205 are available to the print exit routine.

---

**Table 43. Print exit routine variables**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Len</th>
<th>Description</th>
<th>Modifiable</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZPRLGLST</td>
<td>CHAR</td>
<td>4</td>
<td>Value is LOG if processing the LOG, LIST if processing the LIST.</td>
<td>No</td>
</tr>
<tr>
<td>ZPRLGDSN</td>
<td>CHAR</td>
<td>44</td>
<td>Fully qualified log data set name (no quotes) of the data set to print.</td>
<td>No</td>
</tr>
<tr>
<td>ZPRLSDSN</td>
<td>CHAR</td>
<td>44</td>
<td>Fully qualified list data set name (no quotes) of the data set to print.</td>
<td>No</td>
</tr>
<tr>
<td>ZPRLGPMD</td>
<td>CHAR</td>
<td>5</td>
<td>Log print mode, BATCH or LOCAL.</td>
<td>Yes</td>
</tr>
<tr>
<td>ZPRLSPMD</td>
<td>CHAR</td>
<td>5</td>
<td>List print mode, BATCH or LOCAL.</td>
<td>Yes</td>
</tr>
<tr>
<td>*ZPRLGOPT</td>
<td>CHAR</td>
<td>1</td>
<td>Disposition of the log data set as specified by the user D (delete)</td>
<td>Yes</td>
</tr>
<tr>
<td>*ZPRLSOPT</td>
<td>CHAR</td>
<td>1</td>
<td>Disposition of the list data set as specified by the user D (delete)</td>
<td>Yes</td>
</tr>
<tr>
<td>*ZPRRECFCM</td>
<td>CHAR</td>
<td>6</td>
<td>Record format of the list data set.</td>
<td>No</td>
</tr>
<tr>
<td>*ZPRLRECL</td>
<td>CHAR</td>
<td>5</td>
<td>Logical record length of the list data set.</td>
<td>No</td>
</tr>
<tr>
<td>*ZPRLGSYS</td>
<td>CHAR</td>
<td>15</td>
<td>Batch SYSOUT class for the log data set as specified by the user.</td>
<td>Yes</td>
</tr>
<tr>
<td>ZPRLGSYL</td>
<td>CHAR</td>
<td>15</td>
<td>Local SYSOUT class for the log data set as specified by the user.</td>
<td>Yes</td>
</tr>
<tr>
<td>*ZPRLSSYS</td>
<td>CHAR</td>
<td>15</td>
<td>Batch SYSOUT class for the list data set as specified by the user.</td>
<td>Yes</td>
</tr>
<tr>
<td>ZPRLSSYL</td>
<td>CHAR</td>
<td>15</td>
<td>Local SYSOUT class for the list data set as specified by the user.</td>
<td>Yes</td>
</tr>
<tr>
<td>*ZPRLGPRT</td>
<td>CHAR</td>
<td>17</td>
<td>Local printer ID for the log data set as specified by the user.</td>
<td>Yes</td>
</tr>
<tr>
<td>*ZPRLSPRT</td>
<td>CHAR</td>
<td>17</td>
<td>Local printer ID for the list data set as specified by the user.</td>
<td>Yes</td>
</tr>
<tr>
<td>*ZPRJB1</td>
<td>CHAR</td>
<td>72</td>
<td>Print job card 1 as specified by the user</td>
<td>Yes</td>
</tr>
<tr>
<td>*ZPRJB2</td>
<td>CHAR</td>
<td>72</td>
<td>Print job card 2 as specified by the user</td>
<td>Yes</td>
</tr>
</tbody>
</table>
### Table 43. Print exit routine variables (continued)

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Len</th>
<th>Description</th>
<th>Modifiable</th>
</tr>
</thead>
<tbody>
<tr>
<td>*ZPRJB3</td>
<td>CHAR</td>
<td>72</td>
<td>Print job card 3 as specified by the user</td>
<td>Yes</td>
</tr>
<tr>
<td>*ZPRJB4</td>
<td>CHAR</td>
<td>72</td>
<td>Print job card 4 as specified by the user</td>
<td>Yes</td>
</tr>
<tr>
<td>ZPRMSGF</td>
<td>CHAR</td>
<td>1</td>
<td>Data set disposition message suppression. A value of 'S' stops ISPF setting a final disposition message.</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**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.

**Batch**
- 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.

**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.

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 ZPRSPRT 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_UTILITYPROGRAM_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:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Len</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZCMPDSN</td>
<td>CHAR</td>
<td>44</td>
<td>Fully qualified data set name with no quotes</td>
</tr>
<tr>
<td>ZCMPPVOL</td>
<td>CHAR</td>
<td>6</td>
<td>Volume serial</td>
</tr>
<tr>
<td>ZCMPPSWD</td>
<td>CHAR</td>
<td>8</td>
<td>Data set password</td>
</tr>
<tr>
<td>ZCMPOORIG</td>
<td>CHAR</td>
<td>8</td>
<td>Origin of the call.</td>
</tr>
</tbody>
</table>

- **LMCOMP**: The compress service (LMCOMP) called the exit routine.
- **OPTION31**: The compress exit routine is being called from option 3.1, the compress utility.
- **OPTION34**: 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
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:
Table 45. Details of the parameters used to communicate with the data set name change exit

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Len</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROJECT</td>
<td>CHAR</td>
<td>8</td>
<td>Project name as it was entered on the panel</td>
</tr>
<tr>
<td>GROUP</td>
<td>CHAR</td>
<td>8</td>
<td>Group name as it was entered on the panel</td>
</tr>
<tr>
<td>TYPE</td>
<td>CHAR</td>
<td>8</td>
<td>Type name as it was entered on the panel</td>
</tr>
<tr>
<td>MEMBER</td>
<td>CHAR</td>
<td>8</td>
<td>Member name as it was entered on the panel</td>
</tr>
<tr>
<td>OTHER DATA SET</td>
<td>CHAR</td>
<td>56</td>
<td>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.</td>
</tr>
<tr>
<td>REASON</td>
<td>CHAR</td>
<td>8</td>
<td>Reason the data set is being allocated:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>RECOVERY</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>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'.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>TEMP</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>The data set is a temporary PDF data set. This includes:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• The data set name specified on the member list SAVE command</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• The data set name specified on the data set list SAVE command</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• The data set used during the outlist utility processing (ISPF option 3.8, when using program ISRUOLP instead of CLIST ISRUOL).</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>blank</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Any data set entered on an PDF data entry panel</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Data sets specified on the LMINIT service</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Data sets specified on any service that does not need an LMINIT to have been previously performed.</td>
</tr>
</tbody>
</table>

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/TYP 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 or modification of the specified pattern (“Analyzing patterns with the member list filter exit”
on page 212), and then once for each member in the data set that matches the pattern (“Analyzing
member names with the member list filter exit” on page 213). 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, LMDISP, LMMOVE,
LMPROM, LMPRINT, and LMMSTATS). These invocations would result in a recursive call to the exit routine.

Analyzing patterns with the member list filter exit

Here are the exit parameters and return codes associated with analyzing 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.
Table 46. Details of the parameters used to communicate with the member list filter exit when it is invoked to verify or modify a pattern

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Len</th>
<th>Description</th>
<th>Modifiable</th>
</tr>
</thead>
<tbody>
<tr>
<td>CODE</td>
<td>FIXED</td>
<td>4</td>
<td>Code that indicates that this is the analyze pattern call. Call=1.</td>
<td>No</td>
</tr>
<tr>
<td>PATTERN</td>
<td>CHAR</td>
<td>8</td>
<td>Pattern as entered by the user</td>
<td>Yes</td>
</tr>
</tbody>
</table>

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.

Analyzing member names with the member list filter exit

Here are the exit parameters and return codes associated with analyzing member lists.

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.

Table 47. Details of the parameters used to communicate with the member list filter exit when it is invoked to match members

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Len</th>
<th>Description</th>
<th>Modifiable</th>
</tr>
</thead>
<tbody>
<tr>
<td>CODE</td>
<td>FIXED</td>
<td>4</td>
<td>Code indicating that this is the analyze member name call. Call=2.</td>
<td>No</td>
</tr>
<tr>
<td>MEMNAME</td>
<td>CHAR</td>
<td>8</td>
<td>Member name.</td>
<td>No</td>
</tr>
</tbody>
</table>

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.
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.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Len</th>
<th>Description</th>
<th>Modifiable</th>
</tr>
</thead>
<tbody>
<tr>
<td>LINE COMMAND</td>
<td>CHAR</td>
<td>1</td>
<td>A 1-character field containing the data set list line command.</td>
<td>No</td>
</tr>
<tr>
<td>DATA SET NAME AND MEMBER</td>
<td>CHAR</td>
<td>56</td>
<td>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.</td>
<td>No</td>
</tr>
<tr>
<td>VOLUME SERIAL</td>
<td>CHAR</td>
<td>6</td>
<td>A 6-character field containing the volume on which the data set specified in DATA SET NAME AND MEMBER can be found.</td>
<td>No</td>
</tr>
</tbody>
</table>

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.
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.

PRINTUTILITY_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.

PRINTUTILITY_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.

COMPRESSUTILITY_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.

COMPRESSUTILITY_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:

- **ANY**
  - Any unit may be used
- **UADS**
  - Indicates the UNIT parameter in the users UADS entry should control the unit used
- **unit-name**
  - 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_SUPERC_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_SUPERC_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_SUPERC_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_SUPERC_PROGRAM_INTERFACE field is set to YES.

USE_SUPERC_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.

SUPERC_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_SUPERC_PROGRAM_INTERFACE field is set to YES.

SUPERC_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_SUPERC_PROGRAM_INTERFACE field is set to YES.
SUPERC_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_SUPERC_PROGRAM_INTERFACE field is set to YES.

SUPERC_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_SUPERC_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.

---

**DSLIST removable media interface settings**

The keywords DSLIST_RM_ENABLED, DSLIST_RM_COMMAND, and DSLIST_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.

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 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 ? /.

Here are some examples. The 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 DLIST_RM_APPLID keyword is:

```
DSLIST_RM_APPLID = applid | NONE
```

Where **applid** is the application ID used to invoke the tape interface command defined by DLIST_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:

- **NONE**
  - No checking is done. SCLM-controlled members may be edited and processed outside SCLM.
- **WARN**
  - 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.
- **ERROR**
  - 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 that 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:

0
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). 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 FMNINV 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). 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 FMNINV 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). 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 FMNINV DSV /

**GLOBAL_LINE_COMMAND_TABLE**

Defines the line command table that is active when a line command table is not otherwise specified by the user or supplied as a parameter on the EDIF, EDIT, VIEW or VIIF service call.

**GLOBAL_DISABLE_PACK**

Disables the PACK operation used by the editor. Any currently packed data will be unpacked if saved. This option also disables the PACK operation from having any effect with COPY and MOVE services.

**PREV_GEN_DEFAULT_SAVE_ACTION**

This setting is applicable only when editing a member in a PDSE version 2 data set that is configured for member generations. The setting determines the default action that is taken when a previous generation of a member is saved.

Valid values are:

**NEWGEN**

Saves the member in a new generation. This new generation becomes the current generation. The generation being edited is left unchanged.
NOGEN
Saves the member to the same generation that is being edited.
The default is NOGEN.

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 ZDEFAULT 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 extended statistics to be generated when statistics are saved for a PDS member and any of the line count values for the member exceed 65535. Statistics can be saved under any of the following conditions:
- A member is saved from ISPF Edit
- A member is added or updated by using the LMMADD, LMMREP, or LMMSTATS service
- The reset member statistics panels are used
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
Determined 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
Determined 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 a 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 of 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®
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**

The RESET fields described in this section have no effect unless the VERSION_LEVEL_OF_SITEWIDE_DEFAULTS field is modified.

A RESET field causes ISPF to modify the associated value for each user. The value is reset to the value specified in the configuration field that is associated with the RESET field. For example, the RESET_TAB_TO_ACTIONS_BARS field causes the Tab to action bar choices setting to be reset using the value specified in the TAB_TO_ACTION_BARS field. This resetting 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 RESET 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 RESET values, increment this value by 1. For example, the first time you change any RESET values 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 in the configuration fields that are associated with the RESET fields to update the user’s values. You must increment this value every time you modify a value associated with a RESET field 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.
<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>TAB_TO_ACTION_BARS</strong></td>
<td>Enable tabbing to action bars. Valid values are YES and NO.</td>
<td>YES</td>
</tr>
<tr>
<td><strong>RESET_TAB_TO_ACTION_BARS</strong></td>
<td>Reset the value specified here.</td>
<td>NO</td>
</tr>
<tr>
<td><strong>USE_SESSION_MANAGER</strong></td>
<td>Value is used to set system variable ZSESS and is used to initialize system variable ZSM. Valid values are YES and NO.</td>
<td>NO</td>
</tr>
<tr>
<td><strong>RESET_USE_SESSION_MANAGER</strong></td>
<td>Reset the value of system variable ZSM to the value of the USE_SESSION_MANAGER keyword.</td>
<td>NO</td>
</tr>
<tr>
<td><strong>JUMP_FROM_LEADER_DOTS</strong></td>
<td>Enable the ISPF jump command (for example, =2) from fields with leader dots. Valid values are YES and NO.</td>
<td>YES</td>
</tr>
<tr>
<td><strong>RESET_JUMP_FROM_LEADER_DOTS</strong></td>
<td>Reset the value specified here.</td>
<td>NO</td>
</tr>
<tr>
<td><strong>SHOW_SPLIT_LINE</strong></td>
<td>Should ISPF show the split line when a user is running in split screen mode. Valid values are YES and NO.</td>
<td>YES</td>
</tr>
<tr>
<td><strong>RESET_SHOW_SPLIT_LINE</strong></td>
<td>Reset the value specified here.</td>
<td>NO</td>
</tr>
<tr>
<td><strong>LONG_MESSAGES_IN_POPUP</strong></td>
<td>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.</td>
<td>YES</td>
</tr>
<tr>
<td><strong>RESET_LONG_MESSAGES_IN_POPUP</strong></td>
<td>Reset the value specified here.</td>
<td>NO</td>
</tr>
<tr>
<td><strong>EDIT_PRINTDS_COMMAND</strong></td>
<td>Should the user be allowed to modify the PRINTDS command generated by ISPF before its submission. Valid values are YES and NO.</td>
<td>NO</td>
</tr>
<tr>
<td><strong>RESET_EDIT_PRINTDS_COMMAND</strong></td>
<td>Reset the value specified here.</td>
<td>NO</td>
</tr>
<tr>
<td><strong>RESTORE_TEST_TRACE_OPTIONS</strong></td>
<td>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.</td>
<td>YES</td>
</tr>
</tbody>
</table>
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:

NONE
   Process option not set

J
   Print and delete

D
   Delete without printing

K
   Keep

R
   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:

- **NONE**
  - Process option not set

- **J**
  - Print and delete

- **D**
  - Delete without printing

- **K**
  - Keep

- **R**
  - 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. This can be a 1 to 7 digit number.

The default is 0.

**SCROLL_MAX**
Select the maximum scroll value allowed. This can be a 1 to 7 digit number.

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.

**ISPCtl0_PRIMARY_QUANTITY**
Primary number of blocks to be allocated for file tailoring CNTL data sets.
The default is 10.

**ISPCtl0_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 backgound 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_SYMSTG_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_ISPSPROF_CONFLICT**
The initial action to be taken when the ISPF profile member, normally 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 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.

PDSE_RESERVE_PROCESSING
Option to specify whether ISPF issues a RESERVE to serialize access to PDSE. Valid values are OFF and ON. The default value is ON. See Appendix A, “ISPF enqueue processing for data integrity,” on page 327 for further information.

ISPF_INIT_COMMAND_TABLE_ENTRY
Identifies a command table entry to be invoked when the value of VERSION_LEVEL_OF_SITEWIDE_DEFAULTS has been changed. One possible use for this could be to reset the value of the ISPF ZSTART profile variable. Note that neither ZSTART nor any other start option is processed when this command table entry is invoked. Therefore the invoked function should process ZSTART or other option. ZSTART, or the option, is appended to any other parameters that may be passed via this keyword. The command table entry name plus all arguments is truncated if the length exceeds 255 bytes.

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
ISPF 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, ISPDPTRC, and ISPFTTRC 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
The RESET fields described in this section have no effect unless the VERSION_LEVEL_OF_SITEWIDE_DEFAULTS field is modified. See “ISPF site-wide profile customizations” on page 230 for more details. If you select any RESET fields here, you must increment the value in the VERSION_LEVEL_OF_SITEWIDE_DEFAULTS field.

A RESET field causes ISPF to modify the associated value for each user. The value is reset to the value specified in the configuration field that is associated with the RESET field. For example, the RESET_COLUMN_HEADING field causes the color, intensity, and highlighting used for column headers to be reset using the value specified in the COLUMN_HEADING field. This resetting 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_HEADER
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_SPLITScreens
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:

**Table 49. Search order details**

<table>
<thead>
<tr>
<th>Search Order = BEFORE</th>
<th>Search Order = AFTER</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Application</td>
<td>1. Application</td>
</tr>
<tr>
<td>2. USER (1 to 3)</td>
<td>2. USER (1 to 3)</td>
</tr>
</tbody>
</table>
Table 49. Search order details (continued)

<table>
<thead>
<tr>
<th>Search Order = BEFORE</th>
<th>Search Order = AFTER</th>
</tr>
</thead>
<tbody>
<tr>
<td>3. SITE (1 to 3)</td>
<td>3. System</td>
</tr>
<tr>
<td>4. System</td>
<td>4. SITE (1 to 3)</td>
</tr>
</tbody>
</table>

**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).

**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.

ENABLE_XTIOT  Indicates whether ISPF can process data sets dynamically allocated with an XTIOT and whether ISPF itself requests an XTIOT via dynamic allocation:

YES  XTIOT capability is requested. ISPF checks whether NON_VSAM_XTIOT support has been activated before enabling XTIOT support fully. For an explanation of this keyword in the DEVSUPxx member of parmlib, see Chapter 18, "Using BSAM, BPAM, and QSAM support for XTIOT, uncaptured UCBs, and DSAB above the 16 MB line" in z/OS DFSMS Using the New Functions.

The ISRDDN utility indicates whether an allocation uses XTIOT and disables line command input if XTIOT support is disabled.

NO  XTIOT data sets cannot be processed.

The default is NO.

USE_EIGHT_CHARACTER_USER_ID_LAYOUTS  Specify when to use the layouts that can display 8-character TSO user IDs. Several ISPF panels and saved lists require a modified layout to display 8-character TSO user IDs. The following values are supported:

1  Use the layouts that can display 8-character TSO user IDs when 8-character user IDs are enabled on the system.

2  Always use the layouts that can display 8-character TSO user IDs, whether or not 8-character user IDs are enabled on the system.

The default value is 1.

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).
VSAM_RESTRICTED_ALL_DATASET
The names of the data sets that are restricted from use in View, Edit, and Browse. Wildcards might 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
The RESET fields described in this section are not enabled until the VERSION_LEVEL_OF_SITEWIDE_DEFAULTS field is modified. See “ISPF site-wide profile customizations” on page 230 for more details. If you select any RESET fields here, you must increment the value in the VERSION_LEVEL_OF_SITEWIDE_DEFAULTS field.

A RESET field causes ISPF to modify the associated value for each user. The value is reset to the value specified in the configuration field that is associated with the RESET field. For example, the RESET_LOCAL_LOG_SYSOUT_CLASS field causes the Local SYSOUT class for the log data set to be reset using the value specified in the LOCAL_LOG_SYSOUT_CLASS field. This resetting 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. Set to zero to prevent allocation of the LOG data set.

**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_RECFM
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. Set to zero to prevent allocation of the LIST data set.

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® 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 alphanumerical 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**

<table>
<thead>
<tr>
<th>Workstation color</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blue</td>
</tr>
<tr>
<td>Dark Blue</td>
</tr>
<tr>
<td>Red</td>
</tr>
<tr>
<td>Dark Red</td>
</tr>
<tr>
<td>Pink</td>
</tr>
<tr>
<td>Light Magenta</td>
</tr>
<tr>
<td>Green</td>
</tr>
<tr>
<td>Dark Green</td>
</tr>
<tr>
<td>Turquoise</td>
</tr>
<tr>
<td>Dark Cyan</td>
</tr>
<tr>
<td>Yellow</td>
</tr>
<tr>
<td>Brown</td>
</tr>
<tr>
<td>White</td>
</tr>
<tr>
<td>Black</td>
</tr>
</tbody>
</table>

**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.

**Note:**
1. These options control a shared 17-byte information area that overlays any panel display. For a description and references to the system commands that also affect this area, see "Understanding ISPF panels" in *ISPF User's Guide Volume 1*.

---

**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_ADDRESS**
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.

CP & CS
Used by countries

0037 0697
United States, Canada, Netherlands, Portugal, Brazil, Australia, New Zealand
<table>
<thead>
<tr>
<th>Code Page</th>
<th>Character Set</th>
<th>Used by countries</th>
</tr>
</thead>
<tbody>
<tr>
<td>0273 0697</td>
<td>Austria, Germany</td>
<td></td>
</tr>
<tr>
<td>0280 0697</td>
<td>Italy</td>
<td></td>
</tr>
<tr>
<td>0277 0697</td>
<td>Denmark, Norway</td>
<td></td>
</tr>
<tr>
<td>0278 0697</td>
<td>Finland, Sweden</td>
<td></td>
</tr>
<tr>
<td>0284 0697</td>
<td>Spain, Latin America (Spanish)</td>
<td></td>
</tr>
<tr>
<td>0285 0697</td>
<td>United Kingdom</td>
<td></td>
</tr>
<tr>
<td>0297 0697</td>
<td>France</td>
<td></td>
</tr>
<tr>
<td>0500 0697</td>
<td>Belgium, Switzerland</td>
<td></td>
</tr>
<tr>
<td>0870 0959</td>
<td>Latin 2</td>
<td></td>
</tr>
<tr>
<td>0880 0960</td>
<td>Cyrillic</td>
<td></td>
</tr>
<tr>
<td>1025 1150</td>
<td>Cyrillic</td>
<td></td>
</tr>
<tr>
<td>290 1172</td>
<td>Japanese (Katakana)</td>
<td></td>
</tr>
<tr>
<td>1027 1172</td>
<td>Japanese (Latin)</td>
<td></td>
</tr>
<tr>
<td>833 1173</td>
<td>Korean</td>
<td></td>
</tr>
<tr>
<td>836 1174</td>
<td>Simplified Chinese</td>
<td></td>
</tr>
<tr>
<td>037 1175</td>
<td>Traditional Chinese</td>
<td></td>
</tr>
</tbody>
</table>

These Code Pages (CP) and the Character Sets (CS) support the Euro currency sign.

**CP & CS Used by countries**

<table>
<thead>
<tr>
<th>Code Page</th>
<th>Used by countries</th>
</tr>
</thead>
<tbody>
<tr>
<td>1140 0695</td>
<td>United States, Canada, Netherlands, Portugal, Brazil, Australia, New Zealand</td>
</tr>
<tr>
<td>1141 0695</td>
<td>Austria, Germany</td>
</tr>
<tr>
<td>1144 0695</td>
<td>Italy</td>
</tr>
<tr>
<td>1145 0695</td>
<td>Spain, Latin America (Spanish)</td>
</tr>
<tr>
<td>1146 0695</td>
<td>United Kingdom</td>
</tr>
<tr>
<td>1147 0695</td>
<td>France</td>
</tr>
<tr>
<td>1148 0695</td>
<td>Belgium, Canada, Switzerland</td>
</tr>
</tbody>
</table>
**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 is 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
2
   Reserved
3
   AIX®
4
   Reserved
5
   Reserved
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, ISPMEPT, and ISPMXEND macros define the exit entry definition section of the table, and the ISPMXDD and ISPMDAD 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.

**ISPMXDEF**
- Begins the definition of exit routines for a particular user exit.

**ISPMMEPT**
- Identifies an exit routine to call at a particular user exit.

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 137 for a list of the numeric codes.

**ISPMXDD**
- Defines the data areas that are used by the exit routines and their size specifications.

**ISPMDAD**
- Defines the data areas that are used by the exit routines and their size specifications.
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 ISPMDAD 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.

ISPMDAD
Defines a data area and its size. The data area can be used by one or more exit routines.

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 141, 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 141, 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 141, 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 141, 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 = PASLIB 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 141, the exit routines at this user exit receive these parameters:

cmdname
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.

If the command is a CLIST, this field contains the CLIST name. The CLIST name will not include the % prefix if one was used.

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 131 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. 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 131 and “ISPTCM usage notes” on page 134 for information about TCM flags.

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

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.
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 141, 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 131 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 141, 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.
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 327 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.

The RESERVE exit is still called if Configuration table keyword PDSE_RESERVE_PROCESSING is set to OFF.

Exit parameters

In addition to the standard exit parameters described in “Exit parameter list” on page 141, 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 1 = The resource being reserved is a PDSE.

5 1 = PDSE_RESERVE_PROCESSING setting. On by default

6-31 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.

The RELEASE exit is still called if Configuration table keyword PDSE_RESERVE_PROCESSING is set to OFF.

Exit parameters

In addition to the standard exit parameters described in “Exit parameter list” on page 141, 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 141, 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 141, the exit routines at this user exit receive these parameters:

Flags
4 bytes of bit flags defined as follows:

- 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 141, 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 141, 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 141,
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 141, 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:
   uuptpfx.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:

   uuptpx.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 141, 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 = ISPFTTTRC 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

PDF translation tables perform many functions for PDF. Each section describes one particular use.

Translation table for valid data set name characters

The translation table for valid data set name characters (Table 50 on page 281) 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 50. Example of translation table for valid data set characters

<table>
<thead>
<tr>
<th>Table</th>
<th>Hexadecimal Code</th>
<th>Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>TTVDSN</td>
<td>DC X'FFFFFFFFFFFFFFF'</td>
<td>(X'00' to X'07')</td>
</tr>
<tr>
<td></td>
<td>DC X'FFFFFFFFFFFFFFF'</td>
<td>(X'08' to X'0F')</td>
</tr>
<tr>
<td></td>
<td>DC X'FFFFFFFFFFFFFFF'</td>
<td>(X'10' to X'17')</td>
</tr>
<tr>
<td></td>
<td>...</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DC X'FFFF00000000000'</td>
<td>(X'78' to X'7F')</td>
</tr>
<tr>
<td></td>
<td>DC X'FF0000000000000'</td>
<td>(X'80' to X'87')</td>
</tr>
<tr>
<td></td>
<td>...</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DC X'000000000000000'</td>
<td>(X'E8' to X'EF')</td>
</tr>
<tr>
<td></td>
<td>DC X'000000000000000'</td>
<td>(X'F0' to X'F7')</td>
</tr>
<tr>
<td></td>
<td>DC X'000000000000000'</td>
<td>(X'F8' to X'FF')</td>
</tr>
</tbody>
</table>

Translation table for invalid data set name characters

The translation table for invalid data set name characters (Table 51 on page 281) 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 51. Example of translation table for invalid data set name characters

<table>
<thead>
<tr>
<th>Table</th>
<th>Hexadecimal Code</th>
<th>Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>TTIDSN</td>
<td>DC X'000000000000000'</td>
<td>(X'00' to X'07')</td>
</tr>
<tr>
<td></td>
<td>DC X'000000000000000'</td>
<td>(X'08' to X'0F')</td>
</tr>
<tr>
<td></td>
<td>DC X'000000000000000'</td>
<td>(X'10' to X'17')</td>
</tr>
<tr>
<td></td>
<td>...</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DC X'00007A7B7C000000'</td>
<td>(X'78' to X'7F')</td>
</tr>
<tr>
<td></td>
<td>DC X'001828384858687'</td>
<td>(X'80' to X'87')</td>
</tr>
<tr>
<td></td>
<td>...</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DC X'E8E900000000000'</td>
<td>(X'E8' to X'EF')</td>
</tr>
<tr>
<td></td>
<td>DC X'F1F2F3F4F5F6F700'</td>
<td>(X'F0' to X'F7')</td>
</tr>
<tr>
<td></td>
<td>DC X'F8F900000000000'</td>
<td>(X'F8' to X'FF')</td>
</tr>
</tbody>
</table>

Translation table for hexadecimal characters

The translation table for hexadecimal characters (Table 52 on page 282) specifies the valid hexadecimal characters as follows:
• Valid characters are represented with X'00'.
• Invalid characters are represented with X'FF'.

Table 52. Example of translation table for hexadecimal characters

<table>
<thead>
<tr>
<th>Table</th>
<th>Hexadecimal Code</th>
<th>Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>TTHEX</td>
<td>DC X'FFFFFFFFFFFFFFFF'</td>
<td>(X'00' to X'07')</td>
</tr>
<tr>
<td></td>
<td>DC X'FFFFFFFFFFFFFFFF'</td>
<td>(X'08' to X'0F')</td>
</tr>
<tr>
<td></td>
<td>DC X'FFFFFFFFFFFFFFFF'</td>
<td>(X'10' to X'17')</td>
</tr>
<tr>
<td></td>
<td>. . .</td>
<td>(X'78' to X'7F')</td>
</tr>
<tr>
<td></td>
<td>DC X'FFFFFFFFFFFFFFFF'</td>
<td>(X'80' to X'87')</td>
</tr>
<tr>
<td></td>
<td>DC X'FFFFFFFFFFFFFFFF'</td>
<td>(X'E8' to X'EF')</td>
</tr>
<tr>
<td></td>
<td>DC X'FFFFFFFFFFFFFFFF'</td>
<td>(X'F0' to X'F7')</td>
</tr>
<tr>
<td></td>
<td>DC X'FFFFFFFFFFFFFFFF'</td>
<td>(X'F8' to X'FF')</td>
</tr>
</tbody>
</table>

Translation table for numeric characters

The translation table for numeric characters (Table 53 on page 282) specifies the valid numeric characters as follows:

• Valid characters are represented with X'00'.
• Invalid characters are represented with X'FF'.

Table 53. Example of translation table for numeric characters

<table>
<thead>
<tr>
<th>Table</th>
<th>Hexadecimal Code</th>
<th>Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>TTNUM</td>
<td>DC X'FFFFFFFFFFFFFFFF'</td>
<td>(X'00' to X'07')</td>
</tr>
<tr>
<td></td>
<td>DC X'FFFFFFFFFFFFFFFF'</td>
<td>(X'08' to X'0F')</td>
</tr>
<tr>
<td></td>
<td>DC X'FFFFFFFFFFFFFFFF'</td>
<td>(X'10' to X'17')</td>
</tr>
<tr>
<td></td>
<td>. . .</td>
<td>(X'78' to X'7F')</td>
</tr>
<tr>
<td></td>
<td>DC X'FFFFFFFFFFFFFFFF'</td>
<td>(X'80' to X'87')</td>
</tr>
<tr>
<td></td>
<td>. . .</td>
<td>(X'E8' to X'EF')</td>
</tr>
<tr>
<td></td>
<td>DC X'FFFFFFFFFFFFFFFF'</td>
<td>(X'F0' to X'F7')</td>
</tr>
<tr>
<td></td>
<td>DC X'FFFFFFFFFFFFFFFF'</td>
<td>(X'F8' to X'FF')</td>
</tr>
</tbody>
</table>

Translation table for alphanumerical characters

The translation table for alphanumerical characters (Table 54 on page 283) specifies the valid alphanumerical characters as follows:

• Valid characters are represented with X'00'.
• Invalid characters are represented with X'FF'.
Table 54. Example of translation table for alphanumeric characters

<table>
<thead>
<tr>
<th>Table</th>
<th>Hexadecimal Code</th>
<th>Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>TTALN</td>
<td>DC X'FFFFFFFFFFFFFFFF'</td>
<td>(X'00' to X'07')</td>
</tr>
<tr>
<td></td>
<td>DC X'FFFFFFFFFFFFFFFF'</td>
<td>(X'08' to X'0F')</td>
</tr>
<tr>
<td></td>
<td>DC X'FFFFFFFFFFFFFFFF'</td>
<td>(X'10' to X'17')</td>
</tr>
<tr>
<td></td>
<td>DC X'EEEEE0000FFFFF'</td>
<td>(X'78' to X'7F')</td>
</tr>
<tr>
<td></td>
<td>DC X'EEEEE000000000000'</td>
<td>(X'80' to X'87')</td>
</tr>
<tr>
<td></td>
<td>DC X'FFFF000000000000'</td>
<td>(X'E8' to X'EF')</td>
</tr>
<tr>
<td></td>
<td>DC X'FFFF000000000000'</td>
<td>(X'F0' to X'F7')</td>
</tr>
<tr>
<td></td>
<td>DC X'FFFF000000000000'</td>
<td>(X'F8' to X'FF')</td>
</tr>
</tbody>
</table>

Translation table for edit terminal output characters

The translation table for edit terminal output characters (Table 55 on page 283) 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 55. Example of translation table for edit terminal output character

<table>
<thead>
<tr>
<th>Table</th>
<th>Hexadecimal Code</th>
<th>Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>TTETO</td>
<td>DC X'15151515151515'</td>
<td>(X'00' to X'07')</td>
</tr>
<tr>
<td></td>
<td>DC X'15151515151515'</td>
<td>(X'08' to X'0F')</td>
</tr>
<tr>
<td></td>
<td>DC X'15151515151515'</td>
<td>(X'10' to X'17')</td>
</tr>
<tr>
<td></td>
<td>DC X'15797A7B7C7D7E7F'</td>
<td>(X'78' to X'7F')</td>
</tr>
<tr>
<td></td>
<td>DC X'1581828384858687'</td>
<td>(X'80' to X'87')</td>
</tr>
<tr>
<td></td>
<td>DC X'EE9151515151515'</td>
<td>(X'E8' to X'EF')</td>
</tr>
<tr>
<td></td>
<td>DC X'F0F1F2F3F4F5F6F7'</td>
<td>(X'F0' to X'F7')</td>
</tr>
<tr>
<td></td>
<td>DC X'F8F91515151515'</td>
<td>(X'F8' to X'FF')</td>
</tr>
</tbody>
</table>

Translation table for generic string characters

The translation table for generic string characters (Table 56 on page 284) 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 57 on page 284.
**Table 56. Example of translation table for generic string characters**

<table>
<thead>
<tr>
<th>Table</th>
<th>Hexadecimal Code (Mask Value)</th>
<th>Code Mask Offset</th>
</tr>
</thead>
<tbody>
<tr>
<td>TTGSC</td>
<td>DC X'00'</td>
<td>X'00' Any character</td>
</tr>
<tr>
<td></td>
<td>DC X'01'</td>
<td>X'01' Invalid characters</td>
</tr>
<tr>
<td></td>
<td>DC X'02'</td>
<td>X'02' Special characters</td>
</tr>
<tr>
<td></td>
<td>DC X'04'</td>
<td>X'03' APL/TEXT Special</td>
</tr>
<tr>
<td></td>
<td>DC X'08'</td>
<td>X'04' APL/TEXT Alpha</td>
</tr>
<tr>
<td></td>
<td>DC X'10'</td>
<td>X'05' Lower alpha</td>
</tr>
<tr>
<td></td>
<td>DC X'20'</td>
<td>X'06' Upper alpha</td>
</tr>
<tr>
<td></td>
<td>DC X'40'</td>
<td>X'07' Numeric</td>
</tr>
<tr>
<td></td>
<td>DC X'80'</td>
<td>X'08' User defined character set</td>
</tr>
<tr>
<td></td>
<td>DC X'FF'</td>
<td>X'09' (RESERVED)</td>
</tr>
<tr>
<td></td>
<td>DC X'30'</td>
<td>X'0A' Alpha</td>
</tr>
<tr>
<td></td>
<td>DC X'7F'</td>
<td>X'0B' Nonblank</td>
</tr>
<tr>
<td></td>
<td>DC X'FF'</td>
<td>X'0C' (RESERVED)</td>
</tr>
<tr>
<td></td>
<td>DC X'FF'</td>
<td>X'0D' (RESERVED)</td>
</tr>
<tr>
<td></td>
<td>DC X'FF'</td>
<td>X'0E' (RESERVED)</td>
</tr>
<tr>
<td></td>
<td>DC X'FF'</td>
<td>X'0F' (RESERVED)</td>
</tr>
<tr>
<td></td>
<td>DC X'FF'</td>
<td>X'10' Not numeric</td>
</tr>
<tr>
<td></td>
<td>DC X'FF'</td>
<td>X'11' Not upper</td>
</tr>
<tr>
<td></td>
<td>DC X'FF'</td>
<td>X'12' Not lower</td>
</tr>
<tr>
<td></td>
<td>DC X'FF'</td>
<td>X'13' Not alpha</td>
</tr>
<tr>
<td></td>
<td>DC X'FF'</td>
<td>X'14' (RESERVED)</td>
</tr>
<tr>
<td></td>
<td>DC X'FF'</td>
<td>X'15' (RESERVED)</td>
</tr>
<tr>
<td></td>
<td>DC X'FF'</td>
<td>X'16' (RESERVED)</td>
</tr>
</tbody>
</table>

**Translation table for generic string special characters**

The translation table for generic string special characters (Table 57 on page 284) 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 56 on page 284) 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 57 on page 284, (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 57. Example of translation table for generic string special characters**

<table>
<thead>
<tr>
<th>Table</th>
<th>Hexadecimal Code</th>
<th>Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>TTGSS</td>
<td>DC X'FFFFFFFFFFFFF94240D75094FFD0DFFDDD'</td>
<td>(X'00' to X'07')</td>
</tr>
<tr>
<td></td>
<td>DC X'FFFFFFFFFFFFF94240D75094FFD0DFFDDD'</td>
<td>(X'08' to X'0F')</td>
</tr>
<tr>
<td></td>
<td>DC X'FFFFFFFFFFFFF94240D75094FFD0DFFDDD'</td>
<td>(X'10' to X'17')</td>
</tr>
<tr>
<td></td>
<td>DC X'FFFFFFFFFFFFF94240D75094FFD0DFFDDD'</td>
<td>(X'18' to X'1F')</td>
</tr>
<tr>
<td></td>
<td>DC X'FFFFFFFFFFFFF94240D75094FFD0DFFDDD'</td>
<td>(X'20' to X'27')</td>
</tr>
<tr>
<td></td>
<td>DC X'FFFFFFFFFFFFF94240D75094FFD0DFFDDD'</td>
<td>(X'28' to X'2F')</td>
</tr>
<tr>
<td></td>
<td>DC X'FFFFFFFFFFFFF94240D75094FFD0DFFDDD'</td>
<td>(X'30' to X'37')</td>
</tr>
<tr>
<td></td>
<td>DC X'FFFFFFFFFFFFF94240D75094FFD0DFFDDD'</td>
<td>(X'38' to X'3F')</td>
</tr>
</tbody>
</table>

**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 112). The hexadecimal code in the GSM is ANDed with the mask value from the GSC. Any nonzero result is considered a match.

In Table 57 on page 284, 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 58 on page 285) translates data as follows:

- Lowercase alphabetic characters translate to uppercase
- All other characters translate to themselves.

<table>
<thead>
<tr>
<th>Table</th>
<th>Hexadecimal Code</th>
<th>Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>TTUPP</td>
<td>X’0001020304050607’</td>
<td>(X’00’ to X’07’)</td>
</tr>
<tr>
<td></td>
<td>X’08090A0B0C0D0E0F’</td>
<td>(X’08’ to X’0F’)</td>
</tr>
<tr>
<td></td>
<td>X’1011121314151617’</td>
<td>(X’10’ to X’17’)</td>
</tr>
<tr>
<td></td>
<td>X’7B797A7B7C7D7E7F’</td>
<td>(X’78’ to X’7F’)</td>
</tr>
<tr>
<td></td>
<td>X’80C1C2C3C4C5C6C7’</td>
<td>(X’80’ to X’87’)</td>
</tr>
<tr>
<td></td>
<td>X’EB9EABECEDEEEFF’</td>
<td>(X’EB’ to X’EF’)</td>
</tr>
<tr>
<td></td>
<td>X’F0F1F2F3F4F5F6F7’</td>
<td>(X’F0’ to X’F7’)</td>
</tr>
<tr>
<td></td>
<td>X’FB9F9F9F9F9F9F9’</td>
<td>(X’FB’ to X’FF’)</td>
</tr>
</tbody>
</table>

**Translation table for lowercase characters**

The translation table for lowercase characters (Table 59 on page 285) translates data as follows:

- Uppercase alphabetic characters translate to lowercase
- All other characters translate to themselves.

<table>
<thead>
<tr>
<th>Table</th>
<th>Hexadecimal Code</th>
<th>Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>TTLOW</td>
<td>X’0001020304050607’</td>
<td>(X’00’ to X’07’)</td>
</tr>
<tr>
<td></td>
<td>X’08090A0B0C0D0E0F’</td>
<td>(X’08’ to X’0F’)</td>
</tr>
<tr>
<td></td>
<td>X’1011121314151617’</td>
<td>(X’10’ to X’17’)</td>
</tr>
<tr>
<td></td>
<td>X’7B797A7B7C7D7E7F’</td>
<td>(X’78’ to X’7F’)</td>
</tr>
<tr>
<td></td>
<td>X’80C1C2C3C4C5C6C7’</td>
<td>(X’80’ to X’87’)</td>
</tr>
<tr>
<td></td>
<td>X’9A9EABECEDEEEEFF’</td>
<td>(X’9B’ to X’EB’)</td>
</tr>
<tr>
<td></td>
<td>X’F0F1F2F3F4F5F6F7’</td>
<td>(X’F0’ to X’F7’)</td>
</tr>
<tr>
<td></td>
<td>X’FB9F9F9F9F9F9F9’</td>
<td>(X’FB’ to X’FF’)</td>
</tr>
</tbody>
</table>

**Modifying the GSM to use the user character subset**

The generic string master (GSM) translation table (Table 60 on page 286) 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 ISPOWNNT in ISP.SISPSAMP.
Generic string master translation table

The positions in the generic string master translation table (Table 60 on page 286) 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 60. Example of Generic String Master translation table

<table>
<thead>
<tr>
<th>Table</th>
<th>Hexadecimal Code</th>
<th>Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>TTGSM</td>
<td>DC X'0101010101010101'</td>
<td>(X'00' to X'07')</td>
</tr>
<tr>
<td></td>
<td>DC X'0101010101010101'</td>
<td>(X'08' to X'0F')</td>
</tr>
<tr>
<td></td>
<td>DC X'0101010101010101'</td>
<td>(X'10' to X'17')</td>
</tr>
<tr>
<td></td>
<td>DC X'0102020202020202'</td>
<td>(X'78' to X'7F')</td>
</tr>
<tr>
<td></td>
<td>DC X'0110101010101010'</td>
<td>(X'80' to X'87')</td>
</tr>
<tr>
<td></td>
<td>DC X'2020010101010101'</td>
<td>(X'E8' to X'EF')</td>
</tr>
<tr>
<td></td>
<td>DC X'4040404040404040'</td>
<td>(X'F0' to X'F7')</td>
</tr>
<tr>
<td></td>
<td>DC X'4040010101010101'</td>
<td>(X'F8' to X'FF')</td>
</tr>
</tbody>
</table>

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 61 on page 287. Compare to Table 57 on page 284.
2. Modify the GSM entries for hexadecimal codes X'10' through X'17' to turn the high order bit on as shown in Table 62 on page 287. Compare to Table 60 on page 286.

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 63 on page 287.

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".

---

### Table 61. Example of translation table for modified Generic String Special Characters

<table>
<thead>
<tr>
<th>Table</th>
<th>Hexadecimal Code</th>
<th>Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>TTGSS</td>
<td>DC X'FFFFFFFFFFFFF'</td>
<td>(X'00' to X'07')</td>
</tr>
<tr>
<td></td>
<td>DC X'FFFFFFFFFFFFF'</td>
<td>(X'08' to X'0F')</td>
</tr>
<tr>
<td></td>
<td>DC X'FFFFFFFFFFFFF'</td>
<td>(X'10' to X'17')</td>
</tr>
<tr>
<td></td>
<td>DC X'FFFFFFFFFFFFF0105FFFF08'</td>
<td>(X'78' to X'7F')</td>
</tr>
<tr>
<td></td>
<td>DC X'FFFFFFFFFFFFF'</td>
<td>(X'80' to X'87')</td>
</tr>
<tr>
<td></td>
<td>DC X'E8E9FFFFFFFFFFFF'</td>
<td>(X'E8' to X'EF')</td>
</tr>
<tr>
<td></td>
<td>DC X'F0F1F2F3F4F5F6F7'</td>
<td>(X'F0' to X'F7')</td>
</tr>
<tr>
<td></td>
<td>DC X'F8F9FFFFFFFFFFFF'</td>
<td>(X'F8' to X'FF')</td>
</tr>
</tbody>
</table>

---

### Table 62. Example of modified Generic String Master translation table

<table>
<thead>
<tr>
<th>Table</th>
<th>Hexadecimal Code</th>
<th>Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>TTGSM</td>
<td>DC X'0101010101010101'</td>
<td>(X'00' to X'07')</td>
</tr>
<tr>
<td></td>
<td>DC X'0101010101010101'</td>
<td>(X'08' to X'0F')</td>
</tr>
<tr>
<td></td>
<td>DC X'0101010101010101'</td>
<td>(X'10' to X'17')</td>
</tr>
<tr>
<td></td>
<td>DC X'0102020202020202'</td>
<td>(X'78' to X'7F')</td>
</tr>
<tr>
<td></td>
<td>DC X'0110101010101010'</td>
<td>(X'80' to X'87')</td>
</tr>
<tr>
<td></td>
<td>DC X'2020010101010101'</td>
<td>(X'E8' to X'EF')</td>
</tr>
<tr>
<td></td>
<td>DC X'4040404040404040'</td>
<td>(X'F0' to X'F7')</td>
</tr>
<tr>
<td></td>
<td>DC X'4040010101010101'</td>
<td>(X'F8' to X'FF')</td>
</tr>
</tbody>
</table>

---

Table 63. ISPF terminal translation table relationship

<table>
<thead>
<tr>
<th>ISP</th>
<th>PDF</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISP3277 ISP3277A ISP3277K ISP3278 ISP3278A ISP3278C ISP3278K ISP3278T NEW32TBL</td>
<td>ISR3277 ISR3277A ISR3277K ISR3278 ISR3278A ISR3278C ISR3278K ISR3278T ISR32TBL</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Display</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADDPOP</td>
<td>Display pop-up window</td>
<td>ISREMCD5</td>
</tr>
<tr>
<td>DISPLAY</td>
<td>Display option</td>
<td>ISREMCD1</td>
</tr>
<tr>
<td>PQUERY</td>
<td>Get panel information</td>
<td>ISREMCD4</td>
</tr>
<tr>
<td>REMPOP</td>
<td>Remove pop-up window</td>
<td>ISREMCD6</td>
</tr>
<tr>
<td>SETMSG</td>
<td>Set message display</td>
<td>ISREMCD3</td>
</tr>
<tr>
<td>TBDISPL</td>
<td>Table display information</td>
<td>ISREMCD2</td>
</tr>
<tr>
<td><strong>File tailoring</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FTCLOSE</td>
<td>End file tailoring</td>
<td>ISREMCF3</td>
</tr>
<tr>
<td>FTERASE</td>
<td>File tailor erase</td>
<td>ISREMCF4</td>
</tr>
<tr>
<td>FTINCL</td>
<td>File tailor include skeleton</td>
<td>ISREMCF2</td>
</tr>
<tr>
<td>FTOOPEN</td>
<td>File tailor open</td>
<td>ISREMCF1</td>
</tr>
<tr>
<td><strong>Library access</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DIRLIST</td>
<td>Displays z/OS UNIX directory list</td>
<td>ISREMCLD</td>
</tr>
<tr>
<td>LMCLOSE</td>
<td>Close a data set</td>
<td>ISREMCL1</td>
</tr>
<tr>
<td>LMCOMP</td>
<td>Compress a data set</td>
<td>ISREMCLU</td>
</tr>
<tr>
<td>LMCOPY</td>
<td>Copy a data set</td>
<td>ISREMCLQ</td>
</tr>
<tr>
<td>LMDDISP</td>
<td>Data set list</td>
<td>ISREMCLZ</td>
</tr>
<tr>
<td>LMDFREE</td>
<td>Release a data set list</td>
<td>ISREMCLW</td>
</tr>
<tr>
<td>LMDINIT</td>
<td>Establish a data set ID</td>
<td>ISREMCLX</td>
</tr>
<tr>
<td>LMDLIST</td>
<td>Obtain a list of data sets</td>
<td>ISREMCLV</td>
</tr>
<tr>
<td>LMERASE</td>
<td>Erase a data set or library</td>
<td>ISREMCL2</td>
</tr>
<tr>
<td>LMFREE</td>
<td>Release a data set</td>
<td>ISREMCL3</td>
</tr>
<tr>
<td>LMGET</td>
<td>Read a record</td>
<td>ISREMCL4</td>
</tr>
<tr>
<td>LMINIT</td>
<td>Establish a data ID</td>
<td>ISREMCL5</td>
</tr>
<tr>
<td>LMMADD</td>
<td>Add a member</td>
<td>ISREMCL6</td>
</tr>
</tbody>
</table>
Table 64. DM and PDF services in CLIST commands (continued)

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>LMMDEL</td>
<td>Delete a member</td>
<td>ISREMCL7</td>
</tr>
<tr>
<td>LMMDISP</td>
<td>Display member list</td>
<td>ISREMCLO</td>
</tr>
<tr>
<td>LMMFIND</td>
<td>Find a member</td>
<td>ISREMCL8</td>
</tr>
<tr>
<td>LMMLIST</td>
<td>Create a member list</td>
<td>ISREMCL9</td>
</tr>
<tr>
<td>LMMOVE</td>
<td>Move a data set or member</td>
<td>ISREMCLP</td>
</tr>
<tr>
<td>LMMREN</td>
<td>Rename a member</td>
<td>ISREMCLA</td>
</tr>
<tr>
<td>LMMREP</td>
<td>Replace a member</td>
<td>ISREMCLB</td>
</tr>
<tr>
<td>LMMSTATS</td>
<td>Set member statistics</td>
<td>ISREMCLR</td>
</tr>
<tr>
<td>LMOPEN</td>
<td>Open a data set</td>
<td>ISREMCLO</td>
</tr>
<tr>
<td>LMPRINT</td>
<td>Write member to list data set</td>
<td>ISREMCLT</td>
</tr>
<tr>
<td>LMPUT</td>
<td>Write a record</td>
<td>ISREMCLE</td>
</tr>
<tr>
<td>LMQUERY</td>
<td>Provide data set information</td>
<td>ISREMCLF</td>
</tr>
<tr>
<td>LMRENAME</td>
<td>Rename a library</td>
<td>ISREMCLG</td>
</tr>
<tr>
<td>MEMLIST</td>
<td>Displays Option 3.1 member list</td>
<td>ISREMCLH</td>
</tr>
</tbody>
</table>

**Miscellaneous**

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>BROWSE</td>
<td>Browse service</td>
<td>ISREMCM3</td>
</tr>
<tr>
<td>CONTROL</td>
<td>Control service</td>
<td>ISREMCM2</td>
</tr>
<tr>
<td>DSINFO</td>
<td>Returns data set information</td>
<td>ISREMCME</td>
</tr>
<tr>
<td>EDIT</td>
<td>Edit service</td>
<td>ISREMCM4</td>
</tr>
<tr>
<td>EDREC</td>
<td>Edit recovery services</td>
<td>ISREMCM7</td>
</tr>
<tr>
<td>GETMSG</td>
<td>Get message service</td>
<td>ISREMCM6</td>
</tr>
<tr>
<td>LIBDEF</td>
<td>LIBDEF service</td>
<td>ISREMCM8</td>
</tr>
<tr>
<td>LIST</td>
<td>Write data to list data set</td>
<td>ISREMCM5</td>
</tr>
<tr>
<td>LOG</td>
<td>Write message or log data set</td>
<td>ISREMCM8</td>
</tr>
<tr>
<td>QBASELIB</td>
<td>Query base library information</td>
<td>ISREMCMC</td>
</tr>
<tr>
<td>QLIBDEF</td>
<td>Query LIBDEF library information</td>
<td>ISREMCMD</td>
</tr>
<tr>
<td>SELECT</td>
<td>Select service</td>
<td>ISREMCM1</td>
</tr>
</tbody>
</table>

**Table functions (general)**

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBCLOSE</td>
<td>Table close</td>
<td>ISREMCG5</td>
</tr>
<tr>
<td>TBCREATE</td>
<td>Table create</td>
<td>ISREMCG1</td>
</tr>
<tr>
<td>TBEND</td>
<td>Table end</td>
<td>ISREMCG6</td>
</tr>
<tr>
<td>TBERASE</td>
<td>Table erase</td>
<td>ISREMCG7</td>
</tr>
<tr>
<td>TOPEN</td>
<td>Table open</td>
<td>ISREMCG2</td>
</tr>
<tr>
<td>TQUERY</td>
<td>Table query</td>
<td>ISREMCG3</td>
</tr>
</tbody>
</table>
### Table 64. DM and PDF services in CLIST commands (continued)

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBSAVE</td>
<td>Table save</td>
<td>ISREMCG4</td>
</tr>
<tr>
<td>TBSTATS</td>
<td>Table statistics</td>
<td>ISREMCG8</td>
</tr>
</tbody>
</table>

#### Table functions (row)

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBADD</td>
<td>Table row add</td>
<td>ISREMCR1</td>
</tr>
<tr>
<td>TBBOTTOM</td>
<td>Table row pointer to bottom</td>
<td>ISREMCRRA</td>
</tr>
<tr>
<td>TBDELETE</td>
<td>Table delete</td>
<td>ISREMCR2</td>
</tr>
<tr>
<td>TBEXIST</td>
<td>Table exist</td>
<td>ISREMCR6</td>
</tr>
<tr>
<td>TBGET</td>
<td>Table get</td>
<td>ISREMCR6</td>
</tr>
<tr>
<td>TBMOD</td>
<td>Table modify</td>
<td>ISREMCR5</td>
</tr>
<tr>
<td>TBPUT</td>
<td>Table put</td>
<td>ISREMCR4</td>
</tr>
<tr>
<td>TBSARG</td>
<td>Table search parameter</td>
<td>ISREMCR7</td>
</tr>
<tr>
<td>TBSCAN</td>
<td>Table scan</td>
<td>ISREMCR8</td>
</tr>
<tr>
<td>TBSKIP</td>
<td>Table skip</td>
<td>ISREMCRB</td>
</tr>
<tr>
<td>TBSORT</td>
<td>Table sort</td>
<td>ISREMCRD</td>
</tr>
<tr>
<td>TBTOP</td>
<td>Table top</td>
<td>ISREMCR9</td>
</tr>
<tr>
<td>TBVCLEAR</td>
<td>Table variable clear</td>
<td>ISREMCRC</td>
</tr>
</tbody>
</table>

#### Variables

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>VERASE</td>
<td>Variable erase</td>
<td>ISREMCV8</td>
</tr>
<tr>
<td>VGET</td>
<td>Variable get</td>
<td>ISREMCV1</td>
</tr>
<tr>
<td>VPUT</td>
<td>Variable put</td>
<td>ISREMCV2</td>
</tr>
</tbody>
</table>

#### Workstation

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>FILESTAT</td>
<td>Statistics for a file</td>
<td>ISREM CX1</td>
</tr>
<tr>
<td>FILEXFER</td>
<td>Upload or download a file</td>
<td>ISREM CX2</td>
</tr>
<tr>
<td>WS CON</td>
<td>Workstation connection</td>
<td>ISREM CX3</td>
</tr>
<tr>
<td>WSDISCON</td>
<td>Workstation disconnect</td>
<td>ISREM CX4</td>
</tr>
</tbody>
</table>

### DM and PDF services in COBOL programs

#### Table 65. DM and PDF services in COBOL programs

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADDPOP</td>
<td>Display pop-up window</td>
<td>ISREMBD5</td>
</tr>
<tr>
<td>DISPLAY</td>
<td>Display option</td>
<td>ISREMBD1</td>
</tr>
<tr>
<td>PQUERY</td>
<td>Get panel information</td>
<td>ISREMBD4</td>
</tr>
<tr>
<td>Model Name</td>
<td>Description</td>
<td>Member Name</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td>REMPOP</td>
<td>Remove pop-up window</td>
<td>ISREMBD6</td>
</tr>
<tr>
<td>SETMSG</td>
<td>Set message display</td>
<td>ISREMBD3</td>
</tr>
<tr>
<td>TBDISPL</td>
<td>Table display information</td>
<td>ISREMBD2</td>
</tr>
<tr>
<td><strong>File tailoring</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FTCLOSE</td>
<td>End file tailoring</td>
<td>ISREMBF3</td>
</tr>
<tr>
<td>FTERASE</td>
<td>File tailor erase</td>
<td>ISREMBF4</td>
</tr>
<tr>
<td>FTINCL</td>
<td>File tailor include skeleton</td>
<td>ISREMBF2</td>
</tr>
<tr>
<td>FTOOPEN</td>
<td>File tailor open</td>
<td>ISREMBF1</td>
</tr>
<tr>
<td><strong>Graphics</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GRERROR</td>
<td>Graphics error block service</td>
<td>ISREMBS3</td>
</tr>
<tr>
<td>GRINIT</td>
<td>Graphics initialization</td>
<td>ISREMBS1</td>
</tr>
<tr>
<td>GRTERM</td>
<td>Graphics completion service</td>
<td>ISREMBS2</td>
</tr>
<tr>
<td><strong>Library access</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DIRLIST</td>
<td>Displays z/OS UNIX directory list</td>
<td>ISREMBLD</td>
</tr>
<tr>
<td>LMCLOSE</td>
<td>Close a data set</td>
<td>ISREML1</td>
</tr>
<tr>
<td>LMCOMP</td>
<td>Compress a data set</td>
<td>ISREMBLS</td>
</tr>
<tr>
<td>LMCOPY</td>
<td>Copy a data set</td>
<td>ISREMBLQ</td>
</tr>
<tr>
<td>LMDISP</td>
<td>Data set list</td>
<td>ISREMBLZ</td>
</tr>
<tr>
<td>LMDFREE</td>
<td>Release a data set list</td>
<td>ISREMBLW</td>
</tr>
<tr>
<td>LMDINIT</td>
<td>Establish a data set ID</td>
<td>ISREMBLU</td>
</tr>
<tr>
<td>LMDLIST</td>
<td>Obtain a list of data sets</td>
<td>ISREMBLV</td>
</tr>
<tr>
<td>LMERASE</td>
<td>Erase a data set or library</td>
<td>ISREMBL2</td>
</tr>
<tr>
<td>LMFREE</td>
<td>Release a data set</td>
<td>ISREMBL3</td>
</tr>
<tr>
<td>LMGET</td>
<td>Read a record</td>
<td>ISREMBL4</td>
</tr>
<tr>
<td>LMINIT</td>
<td>Establish a data ID</td>
<td>ISREMBL5</td>
</tr>
<tr>
<td>LMMADD</td>
<td>Add a member</td>
<td>ISREMBL6</td>
</tr>
<tr>
<td>LMMDEL</td>
<td>Delete a member</td>
<td>ISREMBL7</td>
</tr>
<tr>
<td>LMM_DISP</td>
<td>Display member list</td>
<td>ISREMBLO</td>
</tr>
<tr>
<td>LMM_FIND</td>
<td>Find a member</td>
<td>ISREMBL8</td>
</tr>
<tr>
<td>LMM_LIST</td>
<td>Create a member list</td>
<td>ISREMBL9</td>
</tr>
<tr>
<td>LMM_MOVE</td>
<td>Move a data set or member</td>
<td>ISREMBLP</td>
</tr>
<tr>
<td>LMM_REN</td>
<td>Rename a member</td>
<td>ISREMBLA</td>
</tr>
<tr>
<td>LMM_REP</td>
<td>Replace a member</td>
<td>ISREMBLB</td>
</tr>
<tr>
<td>LMM_STATS</td>
<td>Set member statistics</td>
<td>ISREMBLR</td>
</tr>
<tr>
<td>Model Name</td>
<td>Description</td>
<td>Member Name</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td>LMOPEN</td>
<td>Open a data set</td>
<td>ISREMBLC</td>
</tr>
<tr>
<td>LMPRINT</td>
<td>Write member to list data set</td>
<td>ISREMBLT</td>
</tr>
<tr>
<td>LMPUT</td>
<td>Write a record</td>
<td>ISREMBLE</td>
</tr>
<tr>
<td>LMQUERY</td>
<td>Provide data set information</td>
<td>ISREMBLFI</td>
</tr>
<tr>
<td>LMRENAME</td>
<td>Rename a library</td>
<td>ISREMBLG</td>
</tr>
<tr>
<td>MEMLIST</td>
<td>Displays Option 3.1 member list</td>
<td>ISREMBLH</td>
</tr>
</tbody>
</table>

**Miscellaneous**

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>BRIF</td>
<td>Browse interface service</td>
<td>ISREMCB4</td>
</tr>
<tr>
<td>BROWSE</td>
<td>Browse service (MVS)</td>
<td>ISREMCB1</td>
</tr>
<tr>
<td>CONTROL</td>
<td>Control service</td>
<td>ISREMBM2</td>
</tr>
<tr>
<td>DSINFO</td>
<td>Returns data set information</td>
<td>ISREMBME</td>
</tr>
<tr>
<td>EDIF</td>
<td>Edit interface service</td>
<td>ISREMCE4</td>
</tr>
<tr>
<td>EDIREC</td>
<td>Edit recovery for EDIF</td>
<td>ISREMBM9</td>
</tr>
<tr>
<td>EDIT</td>
<td>Edit service (MVS)</td>
<td>ISREMCE1</td>
</tr>
<tr>
<td>EDREC</td>
<td>Edit recovery services</td>
<td>ISREMBM7</td>
</tr>
<tr>
<td>GETMSG</td>
<td>Get message service</td>
<td>ISREMBM6</td>
</tr>
<tr>
<td>LIBDEF</td>
<td>LIBDEF service</td>
<td>ISREMBM8</td>
</tr>
<tr>
<td>LIST</td>
<td>Write to list data set</td>
<td>ISREMBMA</td>
</tr>
<tr>
<td>LOG</td>
<td>Write message or log data set</td>
<td>ISREMBM5</td>
</tr>
<tr>
<td>QBASELIB</td>
<td>Query base library information</td>
<td>ISREMBMC</td>
</tr>
<tr>
<td>QLIBDEF</td>
<td>Query LIBDEF library information</td>
<td>ISREMBMD</td>
</tr>
<tr>
<td>SELECT</td>
<td>Select service</td>
<td>ISREMBM1</td>
</tr>
<tr>
<td>VIIF</td>
<td>View Interface service</td>
<td>ISREMCE6</td>
</tr>
</tbody>
</table>

**Table functions (general)**

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBCLOSE</td>
<td>Table close</td>
<td>ISREMBG5</td>
</tr>
<tr>
<td>TBCREATE</td>
<td>Table create</td>
<td>ISREMBG1</td>
</tr>
<tr>
<td>TBEND</td>
<td>Table end</td>
<td>ISREMBG6</td>
</tr>
<tr>
<td>TBERASE</td>
<td>Table erase</td>
<td>ISREMBG7</td>
</tr>
<tr>
<td>TBOPEN</td>
<td>Table open</td>
<td>ISREMBG2</td>
</tr>
<tr>
<td>TBQUERY</td>
<td>Table query</td>
<td>ISREMBG3</td>
</tr>
<tr>
<td>TBSAVE</td>
<td>Table save</td>
<td>ISREMBG4</td>
</tr>
<tr>
<td>TBSTATS</td>
<td>Table statistics</td>
<td>ISREMBG8</td>
</tr>
</tbody>
</table>

**Table functions (row)**

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBADD</td>
<td>Table row add</td>
<td>ISREMBR1</td>
</tr>
</tbody>
</table>
Table 65. DM and PDF services in COBOL programs (continued)

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBBOTTOM</td>
<td>Table row pointer to bottom</td>
<td>ISREMBRA</td>
</tr>
<tr>
<td>TBDELETE</td>
<td>Table delete</td>
<td>ISREMBR2</td>
</tr>
<tr>
<td>TBEXIST</td>
<td>Table exist</td>
<td>ISREMBR6</td>
</tr>
<tr>
<td>TBGET</td>
<td>Table get</td>
<td>ISREMBR3</td>
</tr>
<tr>
<td>TBMOD</td>
<td>Table modify</td>
<td>ISREMBR5</td>
</tr>
<tr>
<td>TBPUT</td>
<td>Table put</td>
<td>ISREMBR4</td>
</tr>
<tr>
<td>TBSARG</td>
<td>Table search parameter</td>
<td>ISREMBR7</td>
</tr>
<tr>
<td>TBSCAN</td>
<td>Table scan</td>
<td>ISREMBR8</td>
</tr>
<tr>
<td>TBSKIP</td>
<td>Table skip</td>
<td>ISREMBR8</td>
</tr>
<tr>
<td>TBSORT</td>
<td>Table sort</td>
<td>ISREMBRD</td>
</tr>
<tr>
<td>TBTOP</td>
<td>Table top</td>
<td>ISREMBR9</td>
</tr>
<tr>
<td>TBVCLEAR</td>
<td>Table variable clear</td>
<td>ISREMBRC</td>
</tr>
</tbody>
</table>

**Variables**

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>VCOPY</td>
<td>Copy variable</td>
<td>ISREMBV5</td>
</tr>
<tr>
<td>VDEFINE</td>
<td>Variable define</td>
<td>ISREMBV3</td>
</tr>
<tr>
<td>VDELETE</td>
<td>Variable delete</td>
<td>ISREMBV4</td>
</tr>
<tr>
<td>VERASE</td>
<td>Variable erase</td>
<td>ISREMBV8</td>
</tr>
<tr>
<td>VGET</td>
<td>Variable get</td>
<td>ISREMBV1</td>
</tr>
<tr>
<td>VMASK</td>
<td>Variable mask</td>
<td>ISREMBV9</td>
</tr>
<tr>
<td>VPUT</td>
<td>Variable put</td>
<td>ISREMBV2</td>
</tr>
<tr>
<td>VREPLACE</td>
<td>Variable replace</td>
<td>ISREMBV6</td>
</tr>
<tr>
<td>VRESET</td>
<td>Variable reset</td>
<td>ISREMBV7</td>
</tr>
</tbody>
</table>

**Working storage**

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>WORKSTOR</td>
<td>Working storage definition</td>
<td>ISREMBW1</td>
</tr>
</tbody>
</table>

**Workstation**

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>FILESTAT</td>
<td>Statistics for a file</td>
<td>ISREMBX1</td>
</tr>
<tr>
<td>FILEXFER</td>
<td>Upload or download a file</td>
<td>ISREMBX2</td>
</tr>
<tr>
<td>WSCON</td>
<td>Workstation connection</td>
<td>ISREMBX3</td>
</tr>
<tr>
<td>WSDISCON</td>
<td>Workstation disconnect</td>
<td>ISREMBX4</td>
</tr>
</tbody>
</table>

**DM and PDF services in EXEC commands**
<table>
<thead>
<tr>
<th>Model Name</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Display</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADDPOP</td>
<td>Display pop-up window</td>
<td>ISREMED5</td>
</tr>
<tr>
<td>DISPLAY</td>
<td>Display option</td>
<td>ISREMED1</td>
</tr>
<tr>
<td>PQURY</td>
<td>Get panel information</td>
<td>ISREMED4</td>
</tr>
<tr>
<td>REMPOP</td>
<td>Remove pop-up window</td>
<td>ISREMED6</td>
</tr>
<tr>
<td>SETMSG</td>
<td>Set message display</td>
<td>ISREMED3</td>
</tr>
<tr>
<td>TBDISPL</td>
<td>Table display information</td>
<td>ISREMED2</td>
</tr>
<tr>
<td><strong>File tailoring</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FTCLOSE</td>
<td>End file tailoring</td>
<td>ISREMEF3</td>
</tr>
<tr>
<td>FTERASE</td>
<td>File tailor erase</td>
<td>ISREMEF4</td>
</tr>
<tr>
<td>FTINCL</td>
<td>File tailor include skeleton</td>
<td>ISREMEF2</td>
</tr>
<tr>
<td>FTPOPEN</td>
<td>File tailor open</td>
<td>ISREMEF1</td>
</tr>
<tr>
<td><strong>Library access</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LMCLOSE</td>
<td>Close a data set</td>
<td>ISREMEL1</td>
</tr>
<tr>
<td>LMERASE</td>
<td>Erase a data set or library</td>
<td>ISREMEL2</td>
</tr>
<tr>
<td>LMFREE</td>
<td>Release a data set</td>
<td>ISREMEL3</td>
</tr>
<tr>
<td>LMGET</td>
<td>Read a record</td>
<td>ISREMEL4</td>
</tr>
<tr>
<td>LMINIT</td>
<td>Establish a data ID</td>
<td>ISREMEL5</td>
</tr>
<tr>
<td>LMMADD</td>
<td>Add a member</td>
<td>ISREMEL6</td>
</tr>
<tr>
<td>LMMDEL</td>
<td>Delete a member</td>
<td>ISREMEL7</td>
</tr>
<tr>
<td>LMMFIND</td>
<td>Find a member</td>
<td>ISREMEL8</td>
</tr>
<tr>
<td>LMMLIST</td>
<td>Create a member list</td>
<td>ISREMEL9</td>
</tr>
<tr>
<td>LMMREN</td>
<td>Rename a member</td>
<td>ISREMELA</td>
</tr>
<tr>
<td>LMMREP</td>
<td>Replace a member</td>
<td>ISREMELB</td>
</tr>
<tr>
<td>LMOPEN</td>
<td>Open a data set</td>
<td>ISREMELC</td>
</tr>
<tr>
<td>LMPUT</td>
<td>Write a record</td>
<td>ISREMELE</td>
</tr>
<tr>
<td>LMQUERY</td>
<td>Provide data set information</td>
<td>ISREMELF</td>
</tr>
<tr>
<td>LMRENAME</td>
<td>Rename a library</td>
<td>ISREMELG</td>
</tr>
<tr>
<td>LMSPEC</td>
<td>Specify a new ISPF library</td>
<td>ISREMELH</td>
</tr>
<tr>
<td>LMUNSPEC</td>
<td>Unspecify an ISPF library</td>
<td>ISREMEL1</td>
</tr>
<tr>
<td><strong>Miscellaneous</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BROWSE</td>
<td>Browse service</td>
<td>ISREMEM3</td>
</tr>
<tr>
<td>CONTROL</td>
<td>Control service</td>
<td>ISREMEM2</td>
</tr>
<tr>
<td>EDIT</td>
<td>Edit service</td>
<td>ISREMEM4</td>
</tr>
<tr>
<td>Model Name</td>
<td>Description</td>
<td>Member Name</td>
</tr>
<tr>
<td>------------</td>
<td>------------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>EDREC</td>
<td>Edit recovery services</td>
<td>ISREMEM7</td>
</tr>
<tr>
<td>GETMSG</td>
<td>Get message service</td>
<td>ISREMEM6</td>
</tr>
<tr>
<td>LIBDEF</td>
<td>LIBDEF service</td>
<td>ISREMEM8</td>
</tr>
<tr>
<td>LIST</td>
<td>Write list data set</td>
<td>ISREMEMA</td>
</tr>
<tr>
<td>LOG</td>
<td>Write message or log data set</td>
<td>ISREMEM5</td>
</tr>
<tr>
<td>QBASELIB</td>
<td>Query base library information</td>
<td>ISREMEMC</td>
</tr>
<tr>
<td>QLIBDEF</td>
<td>Query LIBDEF library information</td>
<td>ISREMEMD</td>
</tr>
<tr>
<td>SELECT</td>
<td>Select service</td>
<td>ISREMEM1</td>
</tr>
</tbody>
</table>

**Table functions (general)**

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBCLOSE</td>
<td>Table close</td>
<td>ISREMEG5</td>
</tr>
<tr>
<td>TBCREATE</td>
<td>Table create</td>
<td>ISREMEG1</td>
</tr>
<tr>
<td>TBEND</td>
<td>Table end</td>
<td>ISREMEG6</td>
</tr>
<tr>
<td>TBERASE</td>
<td>Table erase</td>
<td>ISREMEG7</td>
</tr>
<tr>
<td>TBOPEN</td>
<td>Table open</td>
<td>ISREMEG2</td>
</tr>
<tr>
<td>TBQUERY</td>
<td>Table query</td>
<td>ISREMEG3</td>
</tr>
<tr>
<td>TBSAVE</td>
<td>Table save</td>
<td>ISREMEG4</td>
</tr>
<tr>
<td>TBSTATS</td>
<td>Table statistics</td>
<td>ISREMEG8</td>
</tr>
</tbody>
</table>

**Table functions (row)**

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBADD</td>
<td>Table row add</td>
<td>ISREMER1</td>
</tr>
<tr>
<td>TBBOTTOM</td>
<td>Table row pointer to bottom</td>
<td>ISREMER4</td>
</tr>
<tr>
<td>TBDELETE</td>
<td>Table delete</td>
<td>ISREMER2</td>
</tr>
<tr>
<td>TBEXIST</td>
<td>Table exist</td>
<td>ISREMER6</td>
</tr>
<tr>
<td>TBGET</td>
<td>Table get</td>
<td>ISREMER3</td>
</tr>
<tr>
<td>TBMOD</td>
<td>Table modify</td>
<td>ISREMER5</td>
</tr>
<tr>
<td>TBPUT</td>
<td>Table put</td>
<td>ISREMER4</td>
</tr>
<tr>
<td>TBSARG</td>
<td>Table search parameter</td>
<td>ISREMER7</td>
</tr>
<tr>
<td>TBSCAN</td>
<td>Table scan</td>
<td>ISREMER8</td>
</tr>
<tr>
<td>TBSKIP</td>
<td>Table skip</td>
<td>ISREMER8</td>
</tr>
<tr>
<td>TBSORT</td>
<td>Table sort</td>
<td>ISREMER8</td>
</tr>
<tr>
<td>TBTOP</td>
<td>Table top</td>
<td>ISREMER9</td>
</tr>
<tr>
<td>TBVCLEAR</td>
<td>Table variable clear</td>
<td>ISREMERC</td>
</tr>
</tbody>
</table>

**Variables**

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGET</td>
<td>Variable get</td>
<td>ISREMEV1</td>
</tr>
<tr>
<td>VPUT</td>
<td>Variable put</td>
<td>ISREMEV2</td>
</tr>
</tbody>
</table>
### DM and PDF services in EXEC commands (continued)

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>VERASE</td>
<td>Variable erase</td>
<td>ISREMEV8</td>
</tr>
</tbody>
</table>

### DM and PDF services in FORTRAN programs

### Table 67. DM and PDF services in FORTRAN programs

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADDPOP</td>
<td>Display pop-up window</td>
<td>ISREMFD5</td>
</tr>
<tr>
<td>DISPLAY</td>
<td>Display option</td>
<td>ISREMFD1</td>
</tr>
<tr>
<td>PQUERY</td>
<td>Get panel information</td>
<td>ISREMFD4</td>
</tr>
<tr>
<td>REMPOPOP</td>
<td>Remove pop-up window</td>
<td>ISREMFD6</td>
</tr>
<tr>
<td>SETMSG</td>
<td>Set message display</td>
<td>ISREMFD3</td>
</tr>
<tr>
<td>TBDISPL</td>
<td>Table display information</td>
<td>ISREMFD2</td>
</tr>
<tr>
<td>File tailoring</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FTCLOSE</td>
<td>End file tailoring</td>
<td>ISREMFF3</td>
</tr>
<tr>
<td>FTERASE</td>
<td>File tailor erase</td>
<td>ISREMFF4</td>
</tr>
<tr>
<td>FTINCL</td>
<td>File tailor include skeleton</td>
<td>ISREMFF2</td>
</tr>
<tr>
<td>FTOOPEN</td>
<td>File tailor open</td>
<td>ISREMFF1</td>
</tr>
<tr>
<td>Graphics</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GRERROR</td>
<td>Graphics error block service</td>
<td>ISREMFS3</td>
</tr>
<tr>
<td>GRINIT</td>
<td>Graphics initialization</td>
<td>ISREMFS1</td>
</tr>
<tr>
<td>GRTERM</td>
<td>Graphics completion service</td>
<td>ISREMFS2</td>
</tr>
<tr>
<td>Library access</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DIRLIST</td>
<td>Displays z/OS UNIX directory list</td>
<td>ISREMFLD</td>
</tr>
<tr>
<td>LMCLOSE</td>
<td>Close a data set</td>
<td>ISREMFL1</td>
</tr>
<tr>
<td>LMCOMP</td>
<td>Compress a data set</td>
<td>ISREMFLS</td>
</tr>
<tr>
<td>LMCOPY</td>
<td>Copy a data set</td>
<td>ISREMFLQ</td>
</tr>
<tr>
<td>LMDDISP</td>
<td>Data set list</td>
<td>ISREMFLZ</td>
</tr>
<tr>
<td>LMDFREE</td>
<td>Release a data set list</td>
<td>ISREMFLW</td>
</tr>
<tr>
<td>LMDINIT</td>
<td>Establish a data set ID</td>
<td>ISREMFLU</td>
</tr>
<tr>
<td>LMDLIST</td>
<td>Obtain a list of data sets</td>
<td>ISREMFLV</td>
</tr>
<tr>
<td>LMERASE</td>
<td>Erase a data set or library</td>
<td>ISREMFL2</td>
</tr>
<tr>
<td>LMFREE</td>
<td>Release a data set</td>
<td>ISREMFL3</td>
</tr>
<tr>
<td>LMGET</td>
<td>Read a record</td>
<td>ISREMFL4</td>
</tr>
</tbody>
</table>
### Table 67. DM and PDF services in FORTRAN programs (continued)

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>LMINIT</td>
<td>Establish a data ID</td>
<td>ISREMFL5</td>
</tr>
<tr>
<td>LMMADD</td>
<td>Add a member</td>
<td>ISREMFL6</td>
</tr>
<tr>
<td>LMMDEL</td>
<td>Delete a member</td>
<td>ISREMFL7</td>
</tr>
<tr>
<td>LMM_DISP</td>
<td>Display member list</td>
<td>ISREMFLO</td>
</tr>
<tr>
<td>LMM_FIND</td>
<td>Find a member</td>
<td>ISREMFL8</td>
</tr>
<tr>
<td>LM_LIST</td>
<td>Create a member list</td>
<td>ISREMFL9</td>
</tr>
<tr>
<td>LM_MOVE</td>
<td>Move a data set or member</td>
<td>ISREMFLP</td>
</tr>
<tr>
<td>LM_REN</td>
<td>Rename a member</td>
<td>ISREMFLP</td>
</tr>
<tr>
<td>LM_REP</td>
<td>Replace a member</td>
<td>ISREMFLB</td>
</tr>
<tr>
<td>LMM_STATS</td>
<td>Set member statistics</td>
<td>ISREMFLR</td>
</tr>
<tr>
<td>LM_OPEN</td>
<td>Open a data set</td>
<td>ISREMFLC</td>
</tr>
<tr>
<td>LM_PRINT</td>
<td>Write member to list data set</td>
<td>ISREMFLT</td>
</tr>
<tr>
<td>LM_PUT</td>
<td>Write a record</td>
<td>ISREMFLE</td>
</tr>
<tr>
<td>LM_QUERY</td>
<td>Provide data set information</td>
<td>ISREMFLF</td>
</tr>
<tr>
<td>LM_RENAME</td>
<td>Rename a library</td>
<td>ISREMFLG</td>
</tr>
<tr>
<td>MEMLIST</td>
<td>Displays Option 3.1 member list</td>
<td>ISREMFLH</td>
</tr>
</tbody>
</table>

### Miscellaneous

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>BRIF</td>
<td>Browse interface service</td>
<td>ISREMFB4</td>
</tr>
<tr>
<td>BROWSE</td>
<td>Browse service (MVS)</td>
<td>ISREMFB1</td>
</tr>
<tr>
<td>CONTROL</td>
<td>Control service</td>
<td>ISREMFM2</td>
</tr>
<tr>
<td>DSINFO</td>
<td>Returns data set information</td>
<td>ISREMFM3</td>
</tr>
<tr>
<td>EDIF</td>
<td>Edit interface service</td>
<td>ISREMFE4</td>
</tr>
<tr>
<td>EDIREC</td>
<td>Edit recovery for EDIF</td>
<td>ISREMFM9</td>
</tr>
<tr>
<td>EDIT</td>
<td>Edit service (MVS)</td>
<td>ISREMFE1</td>
</tr>
<tr>
<td>EDREC</td>
<td>Edit recovery services</td>
<td>ISREMFM7</td>
</tr>
<tr>
<td>GETMSG</td>
<td>Get message service</td>
<td>ISREMFM6</td>
</tr>
<tr>
<td>LIBDEF</td>
<td>LIBDEF service</td>
<td>ISREMFM8</td>
</tr>
<tr>
<td>LIST</td>
<td>Write to list data set</td>
<td>ISREMFM9</td>
</tr>
<tr>
<td>LOG</td>
<td>Write message or log data set</td>
<td>ISREMFM5</td>
</tr>
<tr>
<td>QBASELIB</td>
<td>Query base library information</td>
<td>ISREMFM3</td>
</tr>
<tr>
<td>QLIBDEF</td>
<td>Query LIBDEF library information</td>
<td>ISREMFM4</td>
</tr>
<tr>
<td>SELECT</td>
<td>Select service</td>
<td>ISREMFM1</td>
</tr>
<tr>
<td>VIIF</td>
<td>View Interface service</td>
<td>ISREMFE6</td>
</tr>
</tbody>
</table>

Table functions (general)
<table>
<thead>
<tr>
<th>Model Name</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBCLOSE</td>
<td>Table close</td>
<td>ISREMFZ5</td>
</tr>
<tr>
<td>TBCREATE</td>
<td>Table create</td>
<td>ISREMFZ1</td>
</tr>
<tr>
<td>TBEND</td>
<td>Table end</td>
<td>ISREMFZ6</td>
</tr>
<tr>
<td>TBERASE</td>
<td>Table erase</td>
<td>ISREMFZ7</td>
</tr>
<tr>
<td>TBOPEN</td>
<td>Table open</td>
<td>ISREMFZ2</td>
</tr>
<tr>
<td>TBQUERY</td>
<td>Table query</td>
<td>ISREMFZ3</td>
</tr>
<tr>
<td>TBSAVE</td>
<td>Table save</td>
<td>ISREMFZ4</td>
</tr>
<tr>
<td>TBSTATS</td>
<td>Table statistics</td>
<td>ISREMFZ8</td>
</tr>
</tbody>
</table>

**Table functions (row)**

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBADD</td>
<td>Table row add</td>
<td>ISREMFR1</td>
</tr>
<tr>
<td>TBBOTTOM</td>
<td>Table row pointer to bottom</td>
<td>ISREMFRA</td>
</tr>
<tr>
<td>TBDELETE</td>
<td>Table delete</td>
<td>ISREMFR2</td>
</tr>
<tr>
<td>TBEXIST</td>
<td>Table exist</td>
<td>ISREMFR6</td>
</tr>
<tr>
<td>TBGET</td>
<td>Table get</td>
<td>ISREMFR3</td>
</tr>
<tr>
<td>TBMOD</td>
<td>Table modify</td>
<td>ISREMFR5</td>
</tr>
<tr>
<td>TBPUT</td>
<td>Table put</td>
<td>ISREMFR4</td>
</tr>
<tr>
<td>TBSARG</td>
<td>Table search parameter</td>
<td>ISREMFR7</td>
</tr>
<tr>
<td>TBSSCAN</td>
<td>Table scan</td>
<td>ISREMFR8</td>
</tr>
<tr>
<td>TBSKIP</td>
<td>Table skip</td>
<td>ISREMFR8</td>
</tr>
<tr>
<td>TBSORT</td>
<td>Table sort</td>
<td>ISREMFRD</td>
</tr>
<tr>
<td>TBTOP</td>
<td>Table top</td>
<td>ISREMFR9</td>
</tr>
<tr>
<td>TBVCLEAR</td>
<td>Table variable clear</td>
<td>ISREMFRC</td>
</tr>
</tbody>
</table>

**Variables**

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>VCOPY</td>
<td>Copy variable</td>
<td>ISREMFV5</td>
</tr>
<tr>
<td>VDEFINE</td>
<td>Variable define</td>
<td>ISREMFV3</td>
</tr>
<tr>
<td>VDELETE</td>
<td>Variable delete</td>
<td>ISREMFV4</td>
</tr>
<tr>
<td>VERASE</td>
<td>Variable erase</td>
<td>ISREMFV8</td>
</tr>
<tr>
<td>VGET</td>
<td>Variable get</td>
<td>ISREMFV1</td>
</tr>
<tr>
<td>VMASK</td>
<td>Variable mask</td>
<td>ISREMBV9</td>
</tr>
<tr>
<td>VPUT</td>
<td>Variable put</td>
<td>ISREMFV2</td>
</tr>
<tr>
<td>VREPLACE</td>
<td>Variable replace</td>
<td>ISREMFV6</td>
</tr>
<tr>
<td>VRESET</td>
<td>Variable reset</td>
<td>ISREMFV7</td>
</tr>
</tbody>
</table>

**Workstation**

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>FILESTAT</td>
<td>Statistics for a file</td>
<td>ISREMFX1</td>
</tr>
</tbody>
</table>
Table 67. DM and PDF services in FORTRAN programs (continued)

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>FILEXFER</td>
<td>Upload or download a file</td>
<td>ISREMFX2</td>
</tr>
<tr>
<td>WSCON</td>
<td>Workstation connection</td>
<td>ISREMFX3</td>
</tr>
<tr>
<td>WSDISCON</td>
<td>Workstation disconnect</td>
<td>ISREMFX4</td>
</tr>
</tbody>
</table>

Message format

Table 68. Message format

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSGS</td>
<td>Message member selection</td>
<td>ISREMMSG</td>
</tr>
</tbody>
</table>

Panel formats and statements

Table 69. Panel formats and statements

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Qualifier</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Panel formats</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACTION</td>
<td></td>
<td>Panel with action bar</td>
<td>ISREMMF6</td>
</tr>
<tr>
<td>ENTRY</td>
<td></td>
<td>Data entry panel</td>
<td>ISREMMF1</td>
</tr>
<tr>
<td>HELPSCR</td>
<td></td>
<td>Help panel with scrollable area</td>
<td>ISREMMF8</td>
</tr>
<tr>
<td>MULTIPLE</td>
<td>SELECT2</td>
<td>Double-column selection panel</td>
<td>ISREMSE1</td>
</tr>
<tr>
<td>MULTIPLE</td>
<td>ENTRY2</td>
<td>Double-column entry panel</td>
<td>ISREMSE2</td>
</tr>
<tr>
<td>SCROLL</td>
<td></td>
<td>Panel with scrollable area</td>
<td>ISREMMF7</td>
</tr>
<tr>
<td>SELECTION</td>
<td></td>
<td>Choice panel</td>
<td>ISREMMF3</td>
</tr>
<tr>
<td>SELECTION</td>
<td>CUA</td>
<td>Choice panel</td>
<td>ISREMMF9</td>
</tr>
<tr>
<td>TBDISPL</td>
<td></td>
<td>Table display table</td>
<td>ISREMMF4</td>
</tr>
<tr>
<td>TUTORIAL</td>
<td></td>
<td>Help panel</td>
<td>ISREMMF5</td>
</tr>
<tr>
<td>Panel statements</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ASSIGN</td>
<td>SIMPLE</td>
<td>Simple assignment statement</td>
<td>ISREMAS1</td>
</tr>
<tr>
<td>ASSIGN</td>
<td>TRANS</td>
<td>Trans assignment statement</td>
<td>ISREMAS2</td>
</tr>
<tr>
<td>ASSIGN</td>
<td>TRUNC</td>
<td>Trunc assignment statement</td>
<td>ISREMAS3</td>
</tr>
</tbody>
</table>
## Table 69. Panel formats and statements (continued)

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Qualifier</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASSIGN</td>
<td>TRANSTRU</td>
<td>Nested translate truncate statement</td>
<td>ISREMAS4</td>
</tr>
<tr>
<td>ASSIGN</td>
<td>PFKEY</td>
<td>Function key built-in function</td>
<td>ISREMAS5</td>
</tr>
<tr>
<td>ASSIGN</td>
<td>LVLINE</td>
<td>Last visible line built-in function</td>
<td>ISREMAS6</td>
</tr>
<tr>
<td>ATTR</td>
<td></td>
<td>Attribute section header</td>
<td>ISREMMS2</td>
</tr>
<tr>
<td>ATTRIB</td>
<td></td>
<td>New attribute character definition</td>
<td>ISREMMS3</td>
</tr>
<tr>
<td>ATTRIBA</td>
<td></td>
<td>New attrib char definition for area</td>
<td>ISREMMSB</td>
</tr>
<tr>
<td>BODY</td>
<td></td>
<td>Body section header</td>
<td>ISREMMS4</td>
</tr>
<tr>
<td>CCSID</td>
<td></td>
<td>CCSID section header</td>
<td>ISREMMPA</td>
</tr>
<tr>
<td>CONTROL</td>
<td>CURSOR</td>
<td>Control first cursor placement</td>
<td>ISREMCN1</td>
</tr>
<tr>
<td>CONTROL</td>
<td>HELP</td>
<td>Establish a tutorial panel</td>
<td>ISREMCN2</td>
</tr>
<tr>
<td>CONTROL</td>
<td>MSG</td>
<td>Identify message to be displayed</td>
<td>ISREMCN3</td>
</tr>
<tr>
<td>CONTROL</td>
<td>RESP</td>
<td>Show user response to panel</td>
<td>ISREMCN4</td>
</tr>
<tr>
<td>CONTROL</td>
<td>TRAIL</td>
<td>Contain remainder from TRUNC function</td>
<td>ISREMCN5</td>
</tr>
<tr>
<td>CONTROL</td>
<td>ALARM</td>
<td>Shows the alarm is to be sounded</td>
<td>ISREMCN6</td>
</tr>
<tr>
<td>CONTROL</td>
<td>ATTR</td>
<td>Override field attr by field name</td>
<td>ISREMCN7</td>
</tr>
<tr>
<td>CONTROL</td>
<td>ATTRCHAR</td>
<td>Override field attr by character</td>
<td>ISREMCN8</td>
</tr>
<tr>
<td>CONTROL</td>
<td>AUTOSEL</td>
<td>Control table display row selection</td>
<td>ISREMCN9</td>
</tr>
<tr>
<td>CONTROL</td>
<td>CSRPOS</td>
<td>Shows position of cursor in field</td>
<td>ISREMCNA</td>
</tr>
<tr>
<td>CONTROL</td>
<td>CSRROW</td>
<td>Row where cursor is positioned</td>
<td>ISREMCNB</td>
</tr>
<tr>
<td>CONTROL</td>
<td>PFKEY</td>
<td>Function key pressed by user</td>
<td>ISREMCND</td>
</tr>
<tr>
<td>CONTROL</td>
<td>ZVARS</td>
<td>Define names of placeholder fields</td>
<td>ISREMCNC</td>
</tr>
<tr>
<td>CUAATTR</td>
<td></td>
<td>CUA attributes</td>
<td>ISREMMSJ</td>
</tr>
<tr>
<td>END</td>
<td></td>
<td>END section header</td>
<td>ISREMMPP</td>
</tr>
<tr>
<td>GROUPBOX</td>
<td></td>
<td>TYPE(GRPBOX) attribute</td>
<td>ISREMAA4</td>
</tr>
<tr>
<td>HELP</td>
<td></td>
<td>HELP section</td>
<td>ISREMMPM</td>
</tr>
<tr>
<td>IF</td>
<td></td>
<td>IF statement</td>
<td>ISREMMS6</td>
</tr>
<tr>
<td>INIT</td>
<td></td>
<td>INIT section header</td>
<td>ISREMMPJ</td>
</tr>
<tr>
<td>KEYLIST</td>
<td></td>
<td>Keylist specification</td>
<td>ISREMMSF</td>
</tr>
<tr>
<td>LIST</td>
<td></td>
<td>LIST section</td>
<td>ISREMMPO</td>
</tr>
<tr>
<td>MODEL</td>
<td></td>
<td>Model section header</td>
<td>ISREMMS7</td>
</tr>
<tr>
<td>PANEXIT</td>
<td></td>
<td>Panel language exit</td>
<td>ISREMMSD</td>
</tr>
<tr>
<td>PDC</td>
<td></td>
<td>Action bar pull-down</td>
<td>ISREMMSH</td>
</tr>
<tr>
<td>PNTS</td>
<td></td>
<td>Point-and-shoot section</td>
<td>ISREMMPN</td>
</tr>
<tr>
<td>PROC</td>
<td></td>
<td>PROC section header</td>
<td>ISREMMPL</td>
</tr>
</tbody>
</table>
Table 69. Panel formats and statements (continued)

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Qualifier</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>REFRESH</td>
<td></td>
<td>Retrieve variables before redisplay</td>
<td>ISREMMSA</td>
</tr>
<tr>
<td>REINIT</td>
<td></td>
<td>REINIT section header</td>
<td>ISREMMPK</td>
</tr>
<tr>
<td>SC</td>
<td>ATTR</td>
<td>TYPE(SC) attribute</td>
<td>ISREMMA5</td>
</tr>
<tr>
<td>VEDIT</td>
<td></td>
<td>Validate available</td>
<td>ISREMMSI</td>
</tr>
<tr>
<td>VERIFY</td>
<td>ALPHA</td>
<td>Alphabetic or special characters</td>
<td>ISREMVE1</td>
</tr>
<tr>
<td>VERIFY</td>
<td>ALPHAB</td>
<td>Alphabetic characters</td>
<td>ISREMVE2</td>
</tr>
<tr>
<td>VERIFY</td>
<td>BIT</td>
<td>Binary characters</td>
<td>ISREMVE3</td>
</tr>
<tr>
<td>VERIFY</td>
<td>DSNAME</td>
<td>TSO data set name</td>
<td>ISREMVE4</td>
</tr>
<tr>
<td>VERIFY</td>
<td>DSNAMEF</td>
<td>TSO data set name with filters</td>
<td>ISREMVE5</td>
</tr>
<tr>
<td>VERIFY</td>
<td>DSNAMEFM</td>
<td>TSO data set name with filter member name only</td>
<td>ISREMVE6</td>
</tr>
<tr>
<td>VERIFY</td>
<td>DSNAMEPQ</td>
<td>TSO data set name' (adds missing end parenthesis and quote)</td>
<td>ISREMVE7</td>
</tr>
<tr>
<td>VERIFY</td>
<td>DSNAMEQ</td>
<td>TSO data set name' (adds missing end quote)</td>
<td>ISREMVE8</td>
</tr>
<tr>
<td>VERIFY</td>
<td>ENUM</td>
<td>Extended numeric</td>
<td>ISREMVE9</td>
</tr>
<tr>
<td>VERIFY</td>
<td>FILEID</td>
<td>CMS file ID</td>
<td>ISREMVEF</td>
</tr>
<tr>
<td>VERIFY</td>
<td>IPADDR4</td>
<td>IP Version 4 address</td>
<td>ISREMVEG</td>
</tr>
<tr>
<td>VERIFY</td>
<td>HEX</td>
<td>Hexadecimal characters</td>
<td>ISREMVEH</td>
</tr>
<tr>
<td>VERIFY</td>
<td>IDATE</td>
<td>International date</td>
<td>ISREMVEI</td>
</tr>
<tr>
<td>VERIFY</td>
<td>INCLUDE</td>
<td>Specify list of types</td>
<td>ISREMVEJ</td>
</tr>
<tr>
<td>VERIFY</td>
<td>ITIME</td>
<td>International time</td>
<td>ISREMVEK</td>
</tr>
<tr>
<td>VERIFY</td>
<td>JDATE</td>
<td>Julian date</td>
<td>ISREMVEA</td>
</tr>
<tr>
<td>VERIFY</td>
<td>JSTD</td>
<td>Julian standard date</td>
<td>ISREMVEB</td>
</tr>
<tr>
<td>VERIFY</td>
<td>LEN</td>
<td>Length of data stored in variable</td>
<td>ISREMVEC</td>
</tr>
<tr>
<td>VERIFY</td>
<td>LIST</td>
<td>List of valid values</td>
<td>ISREMVE6</td>
</tr>
<tr>
<td>VERIFY</td>
<td>LISTV</td>
<td>Specify list of values</td>
<td>ISREMVE7</td>
</tr>
<tr>
<td>VERIFY</td>
<td>LISTVX</td>
<td>Specify list of excluded values</td>
<td>ISREMVE8</td>
</tr>
<tr>
<td>VERIFY</td>
<td>LISTX</td>
<td>Specify list of excluded values</td>
<td>ISREMVE9</td>
</tr>
<tr>
<td>VERIFY</td>
<td>NAME</td>
<td>Data set member name</td>
<td>ISREMVEA</td>
</tr>
<tr>
<td>VERIFY</td>
<td>NAMEF</td>
<td>Data set member name with filters</td>
<td>ISREMVEB</td>
</tr>
<tr>
<td>VERIFY</td>
<td>NONBLANK</td>
<td>Verify nonblank field</td>
<td>ISREMVEC</td>
</tr>
<tr>
<td>VERIFY</td>
<td>NUM</td>
<td>Numeric characters</td>
<td>ISREMVEK</td>
</tr>
<tr>
<td>VERIFY</td>
<td>PICT</td>
<td>Mixed characters matching picture</td>
<td>ISREMVE6</td>
</tr>
<tr>
<td>VERIFY</td>
<td>PICTCN</td>
<td>Constants and mixed characters matching picture</td>
<td>ISREMVE7</td>
</tr>
</tbody>
</table>
### Table 69. Panel formats and statements (continued)

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Qualifier</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>VERIFY</td>
<td>RANGE</td>
<td>Numeric value within specified limits</td>
<td>ISREMVEB</td>
</tr>
<tr>
<td>VERIFY</td>
<td>STDDATE</td>
<td>Standard date</td>
<td>ISREMVEQ</td>
</tr>
<tr>
<td>VERIFY</td>
<td>STDTIME</td>
<td>Standard time</td>
<td>ISREMVEU</td>
</tr>
<tr>
<td>VGET</td>
<td></td>
<td>Variable get statement</td>
<td>ISREMMSC</td>
</tr>
<tr>
<td>VPUT</td>
<td></td>
<td>Variable put statement</td>
<td>ISREMMS9</td>
</tr>
</tbody>
</table>

### DM and PDF services in PL/I programs

#### Table 70. DM and PDF services in PL/I programs

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Display</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADDPOP</td>
<td>Display pop-up window</td>
<td>ISREMPD5</td>
</tr>
<tr>
<td>DISPLAY</td>
<td>Display option</td>
<td>ISREMPD1</td>
</tr>
<tr>
<td>PQUERY</td>
<td>Get panel information</td>
<td>ISREMPD4</td>
</tr>
<tr>
<td>REMPOP</td>
<td>Remove pop-up window</td>
<td>ISREMPD6</td>
</tr>
<tr>
<td>SETMSG</td>
<td>Set message display</td>
<td>ISREMPD3</td>
</tr>
<tr>
<td>TBDISPL</td>
<td>Table display information</td>
<td>ISREMPD2</td>
</tr>
<tr>
<td><strong>File tailoring</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FTCLOSE</td>
<td>End file tailoring</td>
<td>ISREMPF3</td>
</tr>
<tr>
<td>FTERASE</td>
<td>File tailor erase</td>
<td>ISREMPF4</td>
</tr>
<tr>
<td>FTINCL</td>
<td>File tailor include skeleton</td>
<td>ISREMPF2</td>
</tr>
<tr>
<td>FTOOPEN</td>
<td>File tailor open</td>
<td>ISREMPF1</td>
</tr>
<tr>
<td><strong>Graphics</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GRERROR</td>
<td>Graphics error block service</td>
<td>ISREMPSS3</td>
</tr>
<tr>
<td>GRINIT</td>
<td>Graphics initialization</td>
<td>ISREMPSS1</td>
</tr>
<tr>
<td>GRTERM</td>
<td>Graphics completion service</td>
<td>ISREMPSS2</td>
</tr>
<tr>
<td><strong>Library access</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DIRLIST</td>
<td>Displays z/OS UNIX directory list</td>
<td>ISREPLMD</td>
</tr>
<tr>
<td>LMCLOSE</td>
<td>Close a data set</td>
<td>ISREPL1</td>
</tr>
<tr>
<td>LMCOMP</td>
<td>Compress a data set</td>
<td>ISREPL8</td>
</tr>
<tr>
<td>LMCOPY</td>
<td>Copy a data set</td>
<td>ISREPLQ</td>
</tr>
<tr>
<td>LMDDISP</td>
<td>Data set list</td>
<td>ISREPLZ</td>
</tr>
<tr>
<td>LMDFREE</td>
<td>Release a data set list</td>
<td>ISREPLW</td>
</tr>
<tr>
<td>LMDINIT</td>
<td>Establish a data set ID</td>
<td>ISREPLU</td>
</tr>
<tr>
<td>Model Name</td>
<td>Description</td>
<td>Member Name</td>
</tr>
<tr>
<td>------------</td>
<td>------------------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>LMDLIST</td>
<td>Obtain a list of data sets</td>
<td>ISREMPPLV</td>
</tr>
<tr>
<td>LMERASE</td>
<td>Erase a data set or library</td>
<td>ISREMPPL2</td>
</tr>
<tr>
<td>LMFREE</td>
<td>Release a data set</td>
<td>ISREMPPL3</td>
</tr>
<tr>
<td>LMGET</td>
<td>Read a record</td>
<td>ISREMPPL4</td>
</tr>
<tr>
<td>LMINIT</td>
<td>Establish a data ID</td>
<td>ISREMPPL5</td>
</tr>
<tr>
<td>LMMADD</td>
<td>Add a member</td>
<td>ISREMPPL6</td>
</tr>
<tr>
<td>LMMDEL</td>
<td>Delete a member</td>
<td>ISREMPPL7</td>
</tr>
<tr>
<td>LMMDISP</td>
<td>Display member list</td>
<td>ISREMPLO</td>
</tr>
<tr>
<td>LMMFIND</td>
<td>Find a member</td>
<td>ISREMPPL8</td>
</tr>
<tr>
<td>LMMLIST</td>
<td>Create a member list</td>
<td>ISREMPPL9</td>
</tr>
<tr>
<td>LMMOVE</td>
<td>Move a data set or member</td>
<td>ISREMPLP</td>
</tr>
<tr>
<td>LMMREN</td>
<td>Rename a member</td>
<td>ISREMPLA</td>
</tr>
<tr>
<td>LMMREP</td>
<td>Replace a member</td>
<td>ISREMPLB</td>
</tr>
<tr>
<td>LMMSTATS</td>
<td>Set member statistics</td>
<td>ISREMPLR</td>
</tr>
<tr>
<td>LMOOPEN</td>
<td>Open a data set</td>
<td>ISREMPLC</td>
</tr>
<tr>
<td>LMPRINT</td>
<td>Write member to list data set</td>
<td>ISREMPLT</td>
</tr>
<tr>
<td>LMPUT</td>
<td>Write a record</td>
<td>ISREMPLE</td>
</tr>
<tr>
<td>LMQQUERY</td>
<td>Provide data set information</td>
<td>ISREMPLF</td>
</tr>
<tr>
<td>LMRNAME</td>
<td>Rename a library</td>
<td>ISREMPLG</td>
</tr>
<tr>
<td>MEMMLIST</td>
<td>Displays Option 3.1 member list</td>
<td>ISREMPLH</td>
</tr>
</tbody>
</table>

**Miscellaneous**

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>BRIF</td>
<td>Browse interface service</td>
<td>ISREMPB4</td>
</tr>
<tr>
<td>BROWSE</td>
<td>Browse service (MVS)</td>
<td>ISREMPB1</td>
</tr>
<tr>
<td>CONTROL</td>
<td>Control service</td>
<td>ISREMPM2</td>
</tr>
<tr>
<td>DSINFO</td>
<td>Returns data set information</td>
<td>ISREMPME</td>
</tr>
<tr>
<td>EDIF</td>
<td>Edit interface service</td>
<td>ISREMPE4</td>
</tr>
<tr>
<td>EDIREC</td>
<td>EDIF recovery service</td>
<td>ISREMPM9</td>
</tr>
<tr>
<td>EDIT</td>
<td>Edit service (MVS)</td>
<td>ISREMPE1</td>
</tr>
<tr>
<td>EDREC</td>
<td>Edit recovery services</td>
<td>ISREMPM7</td>
</tr>
<tr>
<td>GETMSG</td>
<td>Get message service</td>
<td>ISREMPM6</td>
</tr>
<tr>
<td>LIBDEF</td>
<td>LIBDEF service</td>
<td>ISREMPM8</td>
</tr>
<tr>
<td>LIST</td>
<td>Write to list data set</td>
<td>ISREMPMA</td>
</tr>
<tr>
<td>LOG</td>
<td>Write message or log data set</td>
<td>ISREMPM5</td>
</tr>
<tr>
<td>QBASELIB</td>
<td>Query base library information</td>
<td>ISREMPMC</td>
</tr>
<tr>
<td>Model Name</td>
<td>Description</td>
<td>Member Name</td>
</tr>
<tr>
<td>------------</td>
<td>------------------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>QLIBDEF</td>
<td>Query LIBDEF library information</td>
<td>ISREMPMD</td>
</tr>
<tr>
<td>SELECT</td>
<td>SELECT service</td>
<td>ISREMPM1</td>
</tr>
<tr>
<td>VIIF</td>
<td>View Interface service</td>
<td>ISREMPWE6</td>
</tr>
</tbody>
</table>

**Table functions (general)**

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBCLOSE</td>
<td>Table close</td>
<td>ISREMPZ5</td>
</tr>
<tr>
<td>TBCREATE</td>
<td>Table create</td>
<td>ISREMPZ1</td>
</tr>
<tr>
<td>TBEND</td>
<td>Table end</td>
<td>ISREMPZ6</td>
</tr>
<tr>
<td>TBERASE</td>
<td>Table erase</td>
<td>ISREMPZ7</td>
</tr>
<tr>
<td>TBOPEN</td>
<td>Table open</td>
<td>ISREMPZ2</td>
</tr>
<tr>
<td>TBQUERY</td>
<td>Table query</td>
<td>ISREMPZ3</td>
</tr>
<tr>
<td>TBSAVE</td>
<td>Table save</td>
<td>ISREMPZ4</td>
</tr>
<tr>
<td>TBSTATS</td>
<td>Table statistics</td>
<td>ISREMPZ8</td>
</tr>
</tbody>
</table>

**Table functions (row)**

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBADD</td>
<td>Table row add</td>
<td>ISREMPR1</td>
</tr>
<tr>
<td>TBBOTTOM</td>
<td>Table row pointer to bottom</td>
<td>ISREMPRA</td>
</tr>
<tr>
<td>TBDELETE</td>
<td>Table delete</td>
<td>ISREMPR2</td>
</tr>
<tr>
<td>TBEXIST</td>
<td>Table exist</td>
<td>ISREMPR6</td>
</tr>
<tr>
<td>TBGET</td>
<td>Table get</td>
<td>ISREMPR3</td>
</tr>
<tr>
<td>TBMOD</td>
<td>Table modify</td>
<td>ISREMPR5</td>
</tr>
<tr>
<td>TBPUSH</td>
<td>Table put</td>
<td>ISREMPR4</td>
</tr>
<tr>
<td>TBSARG</td>
<td>Table search parameter</td>
<td>ISREMPR7</td>
</tr>
<tr>
<td>TBSCAN</td>
<td>Table scan</td>
<td>ISREMPR8</td>
</tr>
<tr>
<td>TBSKIP</td>
<td>Table skip</td>
<td>ISREMPR8</td>
</tr>
<tr>
<td>TBSORT</td>
<td>Table sort</td>
<td>ISREMPRD</td>
</tr>
<tr>
<td>TBTOP</td>
<td>Table top</td>
<td>ISREMPR9</td>
</tr>
<tr>
<td>TBVCLEAR</td>
<td>Table variable clear</td>
<td>ISREMPRC</td>
</tr>
</tbody>
</table>

**Variables**

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>VCOPY</td>
<td>Copy variable</td>
<td>ISREMPV5</td>
</tr>
<tr>
<td>VDEFINE</td>
<td>Variable define</td>
<td>ISREMPV3</td>
</tr>
<tr>
<td>VDELETE</td>
<td>Variable delete</td>
<td>ISREMPV4</td>
</tr>
<tr>
<td>VERASE</td>
<td>Variable erase</td>
<td>ISREMPV8</td>
</tr>
<tr>
<td>VGET</td>
<td>Variable get</td>
<td>ISREMPV1</td>
</tr>
<tr>
<td>VMASK</td>
<td>Variable mask</td>
<td>ISREMPV9</td>
</tr>
<tr>
<td>VPUT</td>
<td>Variable put</td>
<td>ISREMPV2</td>
</tr>
</tbody>
</table>
Table 70. DM and PDF services in PL/I programs (continued)

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>VREPLACE</td>
<td>Variable replace</td>
<td>ISREMPV6</td>
</tr>
<tr>
<td>VRESET</td>
<td>Variable reset</td>
<td>ISREMPV7</td>
</tr>
</tbody>
</table>

Workstation

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>FILESTAT</td>
<td>Statistics for a file</td>
<td>ISREMPX1</td>
</tr>
<tr>
<td>FILEXFER</td>
<td>Upload or download a file</td>
<td>ISREMPX2</td>
</tr>
<tr>
<td>WCON</td>
<td>Workstation connection</td>
<td>ISREMPX3</td>
</tr>
<tr>
<td>WDISCON</td>
<td>Workstation disconnect</td>
<td>ISREMPX4</td>
</tr>
</tbody>
</table>

File tailoring control statements

Table 71. File tailoring control statements

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLANK</td>
<td>Create blank lines</td>
<td>ISREMSK8</td>
</tr>
<tr>
<td>CM</td>
<td>Define comment statement</td>
<td>ISREMSK1</td>
</tr>
<tr>
<td>DEFAULT</td>
<td>Change control character defaults</td>
<td>ISREMSK2</td>
</tr>
<tr>
<td>DOT</td>
<td>Define DO group for table row</td>
<td>ISREMSK3</td>
</tr>
<tr>
<td>IM</td>
<td>Imbed specified data set skeleton</td>
<td>ISREMSK4</td>
</tr>
<tr>
<td>SEL</td>
<td>Conditional processing definition</td>
<td>ISREMSK5</td>
</tr>
<tr>
<td>SET</td>
<td>Set dialog variable value</td>
<td>ISREMSK6</td>
</tr>
<tr>
<td>TB</td>
<td>Set tab stop position</td>
<td>ISREMSK7</td>
</tr>
</tbody>
</table>

DM and PDF services in Pascal programs

Table 72. DM and PDF services in Pascal programs

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADDPOP</td>
<td>Display pop-up window</td>
<td>ISREMQD5</td>
</tr>
<tr>
<td>DISPLAY</td>
<td>Display option</td>
<td>ISREMQD1</td>
</tr>
<tr>
<td>QUERY</td>
<td>Get panel information</td>
<td>ISREMQD4</td>
</tr>
<tr>
<td>REMPOP</td>
<td>Remove pop-up window</td>
<td>ISREMQD6</td>
</tr>
<tr>
<td>SETMSG</td>
<td>Set message display</td>
<td>ISREMQD3</td>
</tr>
<tr>
<td>TBDISPL</td>
<td>Table display information</td>
<td>ISREMQD2</td>
</tr>
</tbody>
</table>

File tailoring

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>FTCLOSE</td>
<td>End file tailoring</td>
<td>ISREMQF3</td>
</tr>
</tbody>
</table>
### Table 72. DM and PDF services in Pascal programs (continued)

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>FTERASE</td>
<td>File tailor erase</td>
<td>ISREMQF4</td>
</tr>
<tr>
<td>FTINCL</td>
<td>File tailor include skeleton</td>
<td>ISREMQF2</td>
</tr>
<tr>
<td>FTOPEN</td>
<td>File tailor open</td>
<td>ISREMQF1</td>
</tr>
</tbody>
</table>

#### Graphics

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRERROR</td>
<td>Graphics error block service</td>
<td>ISREMQS3</td>
</tr>
<tr>
<td>GRINIT</td>
<td>Graphics initialization</td>
<td>ISREMQS1</td>
</tr>
<tr>
<td>GRTERM</td>
<td>Graphics completion service</td>
<td>ISREMQS2</td>
</tr>
</tbody>
</table>

#### Library access

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIRLIST</td>
<td>Displays z/OS UNIX directory list</td>
<td>ISREMQLD</td>
</tr>
<tr>
<td>LMCLOSE</td>
<td>Close a data set</td>
<td>ISREMQL1</td>
</tr>
<tr>
<td>LMCOMP</td>
<td>Compress a data set</td>
<td>ISREMQLS</td>
</tr>
<tr>
<td>LMCOPY</td>
<td>Copy a data set</td>
<td>ISREMQLQ</td>
</tr>
<tr>
<td>LMDDISP</td>
<td>Data set list</td>
<td>ISREMQLZ</td>
</tr>
<tr>
<td>LMDFREE</td>
<td>Release a data set list</td>
<td>ISREMQLW</td>
</tr>
<tr>
<td>LMDINIT</td>
<td>Establish a data set ID</td>
<td>ISREMQLU</td>
</tr>
<tr>
<td>LMDLIST</td>
<td>Obtain a list of data sets</td>
<td>ISREMQLV</td>
</tr>
<tr>
<td>LMERASE</td>
<td>Erase a data set or library</td>
<td>ISREMQL2</td>
</tr>
<tr>
<td>LMFREE</td>
<td>Release a data set</td>
<td>ISREMQL3</td>
</tr>
<tr>
<td>LMGET</td>
<td>Read a record</td>
<td>ISREMQL4</td>
</tr>
<tr>
<td>LMINIT</td>
<td>Establish a data set ID</td>
<td>ISREMQL5</td>
</tr>
<tr>
<td>LMMADD</td>
<td>Add a member</td>
<td>ISREMQL6</td>
</tr>
<tr>
<td>LMMDEL</td>
<td>Delete a member</td>
<td>ISREMQL7</td>
</tr>
<tr>
<td>LMMDISP</td>
<td>Display member list</td>
<td>ISREMQLO</td>
</tr>
<tr>
<td>LMMFIND</td>
<td>Find a member</td>
<td>ISREMQL8</td>
</tr>
<tr>
<td>LMLIST</td>
<td>Create a member list</td>
<td>ISREMQL9</td>
</tr>
<tr>
<td>LMMMOVE</td>
<td>Move a data set or member</td>
<td>ISREMQLP</td>
</tr>
<tr>
<td>LMMREN</td>
<td>Rename a member</td>
<td>ISREMQLA</td>
</tr>
<tr>
<td>LMMREP</td>
<td>Replace a member</td>
<td>ISREMQLB</td>
</tr>
<tr>
<td>LMMSTATS</td>
<td>Set member statistics</td>
<td>ISREMQLR</td>
</tr>
<tr>
<td>LMOOPEN</td>
<td>Open a data set</td>
<td>ISREMQLC</td>
</tr>
<tr>
<td>LMPRINT</td>
<td>Write member to list data set</td>
<td>ISREMQLT</td>
</tr>
<tr>
<td>LMPUT</td>
<td>Write a record</td>
<td>ISREMQLE</td>
</tr>
<tr>
<td>LQUERY</td>
<td>Provide data set information</td>
<td>ISREMQLF</td>
</tr>
<tr>
<td>LRENAME</td>
<td>Rename a library</td>
<td>ISREMQLG</td>
</tr>
</tbody>
</table>
### Table 72. DM and PDF services in Pascal programs (continued)

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEMLIST</td>
<td>Displays Option 3.1 member list</td>
<td>ISREMQLH</td>
</tr>
</tbody>
</table>

**Miscellaneous**

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>BRI F</td>
<td>Browse interface service</td>
<td>ISREMQB4</td>
</tr>
<tr>
<td>BROWSE</td>
<td>Browse service (MVS)</td>
<td>ISREMQB1</td>
</tr>
<tr>
<td>CONTROL</td>
<td>Control service</td>
<td>ISREMQM2</td>
</tr>
<tr>
<td>DSINFO</td>
<td>Returns data set information</td>
<td>ISREMQM3</td>
</tr>
<tr>
<td>EDIF</td>
<td>Edit interface service</td>
<td>ISREMQME</td>
</tr>
<tr>
<td>EDIREC</td>
<td>Edit recovery for EDIF</td>
<td>ISREMQM4</td>
</tr>
<tr>
<td>EDIT</td>
<td>Edit service (MVS)</td>
<td>ISREMQE1</td>
</tr>
<tr>
<td>EDREC</td>
<td>Edit recovery services</td>
<td>ISREMQM7</td>
</tr>
<tr>
<td>GETMSG</td>
<td>Get message service</td>
<td>ISREMQM6</td>
</tr>
<tr>
<td>LIBDEF</td>
<td>LIBDEF service</td>
<td>ISREMQM8</td>
</tr>
<tr>
<td>LIST</td>
<td>Write to list data set</td>
<td>ISREMQMA</td>
</tr>
<tr>
<td>LOG</td>
<td>Write message or log data set</td>
<td>ISREMQM5</td>
</tr>
<tr>
<td>QBASELIB</td>
<td>Query base library information</td>
<td>ISREMQMC</td>
</tr>
<tr>
<td>QLIBDEF</td>
<td>Query LIBDEF library information</td>
<td>ISREMQMD</td>
</tr>
<tr>
<td>SELECT</td>
<td>Select service</td>
<td>ISREMQM1</td>
</tr>
<tr>
<td>VIIF</td>
<td>View Interface service</td>
<td>ISREMQE6</td>
</tr>
</tbody>
</table>

**Pascal definitions**

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>PASDEFS</td>
<td>Pascal Definitions</td>
<td>ISREMQPD</td>
</tr>
</tbody>
</table>

**Table functions (general)**

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBCLOSE</td>
<td>Table close</td>
<td>ISREMQZ5</td>
</tr>
<tr>
<td>TBCREATE</td>
<td>Table create</td>
<td>ISREMQZ1</td>
</tr>
<tr>
<td>TBEND</td>
<td>Table end</td>
<td>ISREMQZ6</td>
</tr>
<tr>
<td>TBERASE</td>
<td>Table erase</td>
<td>ISREMQZ6</td>
</tr>
<tr>
<td>TBOOPEN</td>
<td>Table open</td>
<td>ISREMQZ7</td>
</tr>
<tr>
<td>TBQUERY</td>
<td>Table query</td>
<td>ISREMQZ3</td>
</tr>
<tr>
<td>TBSAVE</td>
<td>Table save</td>
<td>ISREMQZ4</td>
</tr>
<tr>
<td>TBSTATS</td>
<td>Table statistics</td>
<td>ISREMQZ8</td>
</tr>
</tbody>
</table>

**Table functions (row)**

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBADD</td>
<td>Table row add</td>
<td>ISREMQR1</td>
</tr>
<tr>
<td>TBBOTTOM</td>
<td>Table row pointer to bottom</td>
<td>ISREMQRA</td>
</tr>
<tr>
<td>TBDELETE</td>
<td>Table delete</td>
<td>ISREMQR2</td>
</tr>
<tr>
<td>TBEXIST</td>
<td>Table exist</td>
<td>ISREMQR6</td>
</tr>
</tbody>
</table>
### Table 72. DM and PDF services in Pascal programs (continued)

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBGET</td>
<td>Table get</td>
<td>ISREMQR3</td>
</tr>
<tr>
<td>TBMOD</td>
<td>Table modify</td>
<td>ISREMQR5</td>
</tr>
<tr>
<td>TBPUT</td>
<td>Table put</td>
<td>ISREMQR4</td>
</tr>
<tr>
<td>TBSARG</td>
<td>Table search parameter</td>
<td>ISREMQR7</td>
</tr>
<tr>
<td>TBSCAN</td>
<td>Table scan</td>
<td>ISREMQR8</td>
</tr>
<tr>
<td>TBSKIP</td>
<td>Table skip</td>
<td>ISREMQRB</td>
</tr>
<tr>
<td>TBSORT</td>
<td>Table sort</td>
<td>ISREMQRD</td>
</tr>
<tr>
<td>TBTOP</td>
<td>Table top</td>
<td>ISREMQR9</td>
</tr>
<tr>
<td>TBVCLEAR</td>
<td>Table variable clear</td>
<td>ISREMQRc</td>
</tr>
</tbody>
</table>

#### Variables

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>VCOPY</td>
<td>Copy variable</td>
<td>ISREMqv5</td>
</tr>
<tr>
<td>VDEFINE</td>
<td>Variable define</td>
<td>ISREMqv3</td>
</tr>
<tr>
<td>VDELETE</td>
<td>Variable delete</td>
<td>ISREMqv4</td>
</tr>
<tr>
<td>VERASE</td>
<td>Variable erase</td>
<td>ISREMqv8</td>
</tr>
<tr>
<td>VGET</td>
<td>Variable get</td>
<td>ISREMqv1</td>
</tr>
<tr>
<td>VMASK</td>
<td>Variable mask</td>
<td>ISREMqv9</td>
</tr>
<tr>
<td>VPUT</td>
<td>Variable put</td>
<td>ISREMqv2</td>
</tr>
<tr>
<td>VREPLACE</td>
<td>Variable replace</td>
<td>ISREMqv6</td>
</tr>
<tr>
<td>VRESET</td>
<td>Variable reset</td>
<td>ISREMqv7</td>
</tr>
</tbody>
</table>

#### Workstation

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>FILESTAT</td>
<td>Statistics for a file</td>
<td>ISREMqx1</td>
</tr>
<tr>
<td>FILEXFER</td>
<td>Upload or download a file</td>
<td>ISREMqx2</td>
</tr>
<tr>
<td>WSCON</td>
<td>Workstation connection</td>
<td>ISREMqx3</td>
</tr>
<tr>
<td>WSDISCON</td>
<td>Workstation disconnect</td>
<td>ISREMqx4</td>
</tr>
</tbody>
</table>

### DM and PDF services in TSO/REXX commands

### Table 73. DM and PDF services in TSO/REXX commands

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADDPOP</td>
<td>Display pop-up window</td>
<td>ISREMrd5</td>
</tr>
<tr>
<td>DISPLAY</td>
<td>Display option</td>
<td>ISREMrd1</td>
</tr>
<tr>
<td>PQUERY</td>
<td>Get panel information</td>
<td>ISREMrd4</td>
</tr>
<tr>
<td>REMPOP</td>
<td>Remove pop-up window</td>
<td>ISREMrd6</td>
</tr>
<tr>
<td>Model Name</td>
<td>Description</td>
<td>Member Name</td>
</tr>
<tr>
<td>------------</td>
<td>-----------------------------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>SETMSG</td>
<td>Set message display</td>
<td>ISREMRD3</td>
</tr>
<tr>
<td>TBDISPL</td>
<td>Table display information</td>
<td>ISREMRD2</td>
</tr>
<tr>
<td><strong>File tailoring</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FTCLOSE</td>
<td>End file tailoring</td>
<td>ISREMRF3</td>
</tr>
<tr>
<td>FTERASE</td>
<td>File tailor erase</td>
<td>ISREMRF4</td>
</tr>
<tr>
<td>FTINCL</td>
<td>File tailor include skeleton</td>
<td>ISREMRF2</td>
</tr>
<tr>
<td>FTOOPEN</td>
<td>File tailor open</td>
<td>ISREMRF1</td>
</tr>
<tr>
<td><strong>Library access</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DIRLIST</td>
<td>Displays z/OS UNIX directory list</td>
<td>ISREMRLD</td>
</tr>
<tr>
<td>LMCLOSE</td>
<td>Close a data set</td>
<td>ISREMRL1</td>
</tr>
<tr>
<td>LMCOMP</td>
<td>Compress a data set</td>
<td>ISREMRLU</td>
</tr>
<tr>
<td>LMCOPY</td>
<td>Copy a data set</td>
<td>ISREMRLQ</td>
</tr>
<tr>
<td>LMDDISP</td>
<td>Data set list</td>
<td>ISREMRLZ</td>
</tr>
<tr>
<td>LMDFREE</td>
<td>Release a data set list</td>
<td>ISREMRLW</td>
</tr>
<tr>
<td>LMDINIT</td>
<td>Establish a data set ID</td>
<td>ISREMRLX</td>
</tr>
<tr>
<td>LMDLIST</td>
<td>Obtain a list of data sets</td>
<td>ISREMRLV</td>
</tr>
<tr>
<td>LMERASE</td>
<td>Erase a data set or library</td>
<td>ISREMRL2</td>
</tr>
<tr>
<td>LMFREE</td>
<td>Release a data set</td>
<td>ISREMRL3</td>
</tr>
<tr>
<td>LMGET</td>
<td>Read a record</td>
<td>ISREMRL4</td>
</tr>
<tr>
<td>LMINIT</td>
<td>Establish a data ID</td>
<td>ISREMRL5</td>
</tr>
<tr>
<td>LMMADD</td>
<td>Add a member</td>
<td>ISREMRL6</td>
</tr>
<tr>
<td>LMMDEL</td>
<td>Delete a member</td>
<td>ISREMRL7</td>
</tr>
<tr>
<td>LMMDISP</td>
<td>Display member list</td>
<td>ISREMRLO</td>
</tr>
<tr>
<td>LMMFIND</td>
<td>Find a member</td>
<td>ISREMRL8</td>
</tr>
<tr>
<td>LMMLIST</td>
<td>Create a member list</td>
<td>ISREMRL9</td>
</tr>
<tr>
<td>LMMMOVE</td>
<td>Move a data set or member</td>
<td>ISREMRLP</td>
</tr>
<tr>
<td>LMMREN</td>
<td>Rename a member</td>
<td>ISREMRLA</td>
</tr>
<tr>
<td>LMMREP</td>
<td>Replace a member</td>
<td>ISREMRLB</td>
</tr>
<tr>
<td>LMMSTATS</td>
<td>Set member statistics</td>
<td>ISREMRLR</td>
</tr>
<tr>
<td>LMOOPEN</td>
<td>Open a data set</td>
<td>ISREMRLC</td>
</tr>
<tr>
<td>LMPRINT</td>
<td>Write member to list data set</td>
<td>ISREMRLT</td>
</tr>
<tr>
<td>LMPUT</td>
<td>Write a record</td>
<td>ISREMRLE</td>
</tr>
<tr>
<td>LQUERY</td>
<td>Provide data set information</td>
<td>ISREMRLF</td>
</tr>
<tr>
<td>LRENAME</td>
<td>Rename a library</td>
<td>ISREMRLG</td>
</tr>
</tbody>
</table>
### Table 73. DM and PDF services in TSO/REXX commands (continued)

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEMLIST</td>
<td>Displays Option 3.1 member list</td>
<td>ISREMRLH</td>
</tr>
<tr>
<td></td>
<td><strong>Miscellaneous</strong></td>
<td></td>
</tr>
<tr>
<td>BROWSE</td>
<td>Browse service</td>
<td>ISREMRM3</td>
</tr>
<tr>
<td>CONTROL</td>
<td>Control service</td>
<td>ISREMRM2</td>
</tr>
<tr>
<td>DSINFO</td>
<td>Returns data set information</td>
<td>ISREMRME</td>
</tr>
<tr>
<td>EDIT</td>
<td>Edit service</td>
<td>ISREMRM4</td>
</tr>
<tr>
<td>EDREC</td>
<td>Edit recovery services</td>
<td>ISREMRM7</td>
</tr>
<tr>
<td>GETMSG</td>
<td>Get message service</td>
<td>ISREMRM6</td>
</tr>
<tr>
<td>LIBDEF</td>
<td>LIBDEF service</td>
<td>ISREMRM8</td>
</tr>
<tr>
<td>LIST</td>
<td>Write to list data set</td>
<td>ISREMRMA</td>
</tr>
<tr>
<td>LOG</td>
<td>Write message or log data set</td>
<td>ISREMRM5</td>
</tr>
<tr>
<td>QBASELIB</td>
<td>Query base library information</td>
<td>ISREMRMC</td>
</tr>
<tr>
<td>QLIBDEF</td>
<td>Query LIBDEF library information</td>
<td>ISREMRMD</td>
</tr>
<tr>
<td>SELECT</td>
<td>SELECT service</td>
<td>ISREMRM1</td>
</tr>
<tr>
<td></td>
<td><strong>Table functions (general)</strong></td>
<td></td>
</tr>
<tr>
<td>TBCLOSE</td>
<td>Table close</td>
<td>ISREMRG5</td>
</tr>
<tr>
<td>TBCREATE</td>
<td>Table create</td>
<td>ISREMRG1</td>
</tr>
<tr>
<td>TBEND</td>
<td>Table end</td>
<td>ISREMRG6</td>
</tr>
<tr>
<td>TBERASE</td>
<td>Table erase</td>
<td>ISREMRG7</td>
</tr>
<tr>
<td>TBOPEN</td>
<td>Table open</td>
<td>ISREMRG2</td>
</tr>
<tr>
<td>TBQUERY</td>
<td>Table query</td>
<td>ISREMRG3</td>
</tr>
<tr>
<td>TBSAVE</td>
<td>Table save</td>
<td>ISREMRG4</td>
</tr>
<tr>
<td>TBSARGS</td>
<td>Table statistics</td>
<td>ISREMRG8</td>
</tr>
<tr>
<td></td>
<td><strong>Table functions (row)</strong></td>
<td></td>
</tr>
<tr>
<td>TBADD</td>
<td>Table row add</td>
<td>ISREMRR1</td>
</tr>
<tr>
<td>TBBOTTOM</td>
<td>Table row pointer to bottom</td>
<td>ISREMRR4</td>
</tr>
<tr>
<td>TBDELETE</td>
<td>Table delete</td>
<td>ISREMRR2</td>
</tr>
<tr>
<td>TBBEXIST</td>
<td>Table exist</td>
<td>ISREMRR6</td>
</tr>
<tr>
<td>TBGET</td>
<td>Table get</td>
<td>ISREMRR3</td>
</tr>
<tr>
<td>TBMOD</td>
<td>Table modify</td>
<td>ISREMRR5</td>
</tr>
<tr>
<td>TBPUBL</td>
<td>Table put</td>
<td>ISREMRR4</td>
</tr>
<tr>
<td>TBSARG</td>
<td>Table search parameter</td>
<td>ISREMRR7</td>
</tr>
<tr>
<td>TBSBAR</td>
<td>Table scan</td>
<td>ISREMRR8</td>
</tr>
<tr>
<td>TBSKIP</td>
<td>Table skip</td>
<td>ISREMRRB</td>
</tr>
</tbody>
</table>
### Table 73. DM and PDF services in TSO/REXX commands (continued)

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBSORT</td>
<td>Table sort</td>
<td>ISREMRRRD</td>
</tr>
<tr>
<td>TBTOP</td>
<td>Table top</td>
<td>ISREMRR9</td>
</tr>
<tr>
<td>TBVCLEAR</td>
<td>Table variable clear</td>
<td>ISREMRRRC</td>
</tr>
</tbody>
</table>

**Variables**

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>VERASE</td>
<td>Variable erase</td>
<td>ISREMRV8</td>
</tr>
<tr>
<td>VGET</td>
<td>Variable get</td>
<td>ISREMRV1</td>
</tr>
<tr>
<td>VPUT</td>
<td>Variable put</td>
<td>ISREMRV2</td>
</tr>
</tbody>
</table>

**Workstation**

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>FILESTAT</td>
<td>Statistics for a file</td>
<td>ISREMRX1</td>
</tr>
<tr>
<td>FILEXFER</td>
<td>Upload or download a file</td>
<td>ISREMRX2</td>
</tr>
<tr>
<td>WSCON</td>
<td>Workstation connection</td>
<td>ISREMRX3</td>
</tr>
<tr>
<td>WSDISCON</td>
<td>Workstation disconnect</td>
<td>ISREMRX4</td>
</tr>
</tbody>
</table>

---

### SCLM architecture definition formats

### Table 74. SCLM architecture definition formats

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>CC</td>
<td>Compilation Control</td>
<td>ISREMHAC</td>
</tr>
<tr>
<td>LEC</td>
<td>Linkage Editor Control</td>
<td>ISREMHAL</td>
</tr>
<tr>
<td>HL</td>
<td>High Level Definition</td>
<td>ISREMHAH</td>
</tr>
<tr>
<td>GENERIC</td>
<td>Special processing control</td>
<td>ISREMHAG</td>
</tr>
</tbody>
</table>

---

### SCLM project definition macros and templates

### Table 75. SCLM project definition macros and templates

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>FLMABEG</td>
<td>Define project name</td>
<td>ISREMGAB</td>
</tr>
<tr>
<td>FLMAEND</td>
<td>End project definition</td>
<td>ISREMGAE</td>
</tr>
<tr>
<td>FLMAGRP</td>
<td>Define authcode group</td>
<td>ISREMGAG</td>
</tr>
<tr>
<td>FLMALLOC</td>
<td>Define ddname</td>
<td>ISREMGAL</td>
</tr>
<tr>
<td>FLMALTC</td>
<td>Define alternate control set</td>
<td>ISREMGAC</td>
</tr>
<tr>
<td>FLMATVER</td>
<td>Enable audit tracking and versioning</td>
<td>ISREMGAV</td>
</tr>
</tbody>
</table>
Table 75. SCLM project definition macros and templates (continued)

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>FLMCNTRL</td>
<td>Specify project controls</td>
<td>ISREMGCN</td>
</tr>
<tr>
<td>FLMCPYLB</td>
<td>Name data set for allocation</td>
<td>ISREMGCP</td>
</tr>
<tr>
<td>FLMEXLIB</td>
<td>Define external library name</td>
<td>ISREMGEX</td>
</tr>
<tr>
<td>FLMGROUP</td>
<td>Define group of libraries</td>
<td>ISREMGGR</td>
</tr>
<tr>
<td>FLMLANGL</td>
<td>Define a language</td>
<td>ISREMGLA</td>
</tr>
<tr>
<td>FLMSYSLB</td>
<td>Name system library</td>
<td>ISREMGSY</td>
</tr>
<tr>
<td>FLMTRNSL</td>
<td>Define language translator</td>
<td>ISREMGTR</td>
</tr>
<tr>
<td>FLMTYPE</td>
<td>Define a library type</td>
<td>ISREMGTY</td>
</tr>
</tbody>
</table>

Templates

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROJDEF</td>
<td>Project definition</td>
<td>ISREMGT1</td>
</tr>
<tr>
<td>LANGUAGE</td>
<td>Language definition</td>
<td>ISREMGT2</td>
</tr>
</tbody>
</table>

ISPF Dialog Tag Language models

Table 76. Dialog Tag Language models

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Panel formats</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACTION BAR</td>
<td>Action bar panel</td>
<td>ISREMDP2</td>
</tr>
<tr>
<td>ENTRY</td>
<td>Data entry panel</td>
<td>ISREMDP1</td>
</tr>
<tr>
<td>SELECTION</td>
<td>Choice panel</td>
<td>ISREMDP3</td>
</tr>
<tr>
<td>TABLE DISPLAY</td>
<td>Scrollable list</td>
<td>ISREMDP5</td>
</tr>
<tr>
<td>TUTORIAL</td>
<td>Help/Tutorial panel</td>
<td>ISREMDP4</td>
</tr>
</tbody>
</table>

Command table format

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMMAND TABLE</td>
<td>Command table application</td>
<td>ISREMDC1</td>
</tr>
</tbody>
</table>

DM and PDF services in C/370 programs

Table 77. DM and PDF services in C/370 programs

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADDPOP</td>
<td>Display pop-up window</td>
<td>ISREMWD5</td>
</tr>
<tr>
<td>DISPLAY</td>
<td>Display option</td>
<td>ISREMWD1</td>
</tr>
<tr>
<td>PQUERY</td>
<td>Get panel information</td>
<td>ISREMWD4</td>
</tr>
<tr>
<td><strong>Model Name</strong></td>
<td><strong>Description</strong></td>
<td><strong>Member Name</strong></td>
</tr>
<tr>
<td>----------------</td>
<td>----------------</td>
<td>----------------</td>
</tr>
<tr>
<td>REMPOPOP</td>
<td>Remove pop-up window</td>
<td>ISREMWD6</td>
</tr>
<tr>
<td>SETMSG</td>
<td>Set message display</td>
<td>ISREMWD3</td>
</tr>
<tr>
<td>TBDISPL</td>
<td>Table display information</td>
<td>ISREMWD2</td>
</tr>
</tbody>
</table>

### File tailoring

<table>
<thead>
<tr>
<th><strong>Model Name</strong></th>
<th><strong>Description</strong></th>
<th><strong>Member Name</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>FTCLOSE</td>
<td>End file tailoring</td>
<td>ISREMWF3</td>
</tr>
<tr>
<td>FTERASE</td>
<td>File tailor erase</td>
<td>ISREMWF4</td>
</tr>
<tr>
<td>FTINCL</td>
<td>File tailor include skeleton</td>
<td>ISREMWF2</td>
</tr>
<tr>
<td>FTOOPEN</td>
<td>File tailor open</td>
<td>ISREMWF1</td>
</tr>
</tbody>
</table>

### Graphics

<table>
<thead>
<tr>
<th><strong>Model Name</strong></th>
<th><strong>Description</strong></th>
<th><strong>Member Name</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>GRERROR</td>
<td>Graphics error block service</td>
<td>ISREMWS3</td>
</tr>
<tr>
<td>GRINIT</td>
<td>Graphics initialization</td>
<td>ISREMWS1</td>
</tr>
<tr>
<td>GRTERM</td>
<td>Graphics completion service</td>
<td>ISREMWS2</td>
</tr>
</tbody>
</table>

### Library access

<table>
<thead>
<tr>
<th><strong>Model Name</strong></th>
<th><strong>Description</strong></th>
<th><strong>Member Name</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>DIRLIST</td>
<td>Displays z/OS UNIX directory list</td>
<td>ISREMWLD</td>
</tr>
<tr>
<td>LMCLOSE</td>
<td>Close a data set</td>
<td>ISREMWL1</td>
</tr>
<tr>
<td>LMCOMP</td>
<td>Compress a data set</td>
<td>ISREMWLO</td>
</tr>
<tr>
<td>LMCOPY</td>
<td>Copy a data set</td>
<td>ISREMWLY</td>
</tr>
<tr>
<td>LMDDISP</td>
<td>Data set list</td>
<td>ISREMWLZ</td>
</tr>
<tr>
<td>LMDFREE</td>
<td>Release a data set list</td>
<td>ISREMWLW</td>
</tr>
<tr>
<td>LMDINIT</td>
<td>Establish a data set ID</td>
<td>ISREMWLU</td>
</tr>
<tr>
<td>LMDLIST</td>
<td>Obtain a list of data sets</td>
<td>ISREMWLV</td>
</tr>
<tr>
<td>LMERASE</td>
<td>Erase a data set or library</td>
<td>ISREMWL2</td>
</tr>
<tr>
<td>LMFREE</td>
<td>Release a data set</td>
<td>ISREMWL3</td>
</tr>
<tr>
<td>LMGET</td>
<td>Read a record</td>
<td>ISREMWL4</td>
</tr>
<tr>
<td>LMINIT</td>
<td>Establish a data ID</td>
<td>ISREMWL5</td>
</tr>
<tr>
<td>LMMADD</td>
<td>Add a member</td>
<td>ISREMWL6</td>
</tr>
<tr>
<td>LMMDEL</td>
<td>Delete a member</td>
<td>ISREMWL7</td>
</tr>
<tr>
<td>LMMDISP</td>
<td>Display member list</td>
<td>ISREMWLL</td>
</tr>
<tr>
<td>LMMFIND</td>
<td>Find a member</td>
<td>ISREMWL8</td>
</tr>
<tr>
<td>LMMLIST</td>
<td>Create a member list</td>
<td>ISREMWL9</td>
</tr>
<tr>
<td>LMOVE</td>
<td>Move a data set or member</td>
<td>ISREMWLM</td>
</tr>
<tr>
<td>LMMREN</td>
<td>Rename a member</td>
<td>ISREMWLO</td>
</tr>
<tr>
<td>LMMREP</td>
<td>Replace a member</td>
<td>ISREMWLB</td>
</tr>
<tr>
<td>LMMSTATS</td>
<td>Set member statistics</td>
<td>ISREMWLP</td>
</tr>
<tr>
<td>Model Name</td>
<td>Description</td>
<td>Member Name</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td>LMOPEN</td>
<td>Open a data set</td>
<td>ISREMWLC</td>
</tr>
<tr>
<td>LMPRINT</td>
<td>Write member to list data set</td>
<td>ISREMWLQ</td>
</tr>
<tr>
<td>LMPUT</td>
<td>Write a record</td>
<td>ISREMWLE</td>
</tr>
<tr>
<td>LMQUERY</td>
<td>Provide data set information</td>
<td>ISREMWLF</td>
</tr>
<tr>
<td>LMRNAME</td>
<td>Rename a library</td>
<td>ISREMWLG</td>
</tr>
<tr>
<td>MEMLIST</td>
<td>Displays Option 3.1 member list</td>
<td>ISREMWLH</td>
</tr>
</tbody>
</table>

**Miscellaneous**

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>BRIF</td>
<td>Browse interface service</td>
<td>ISREMWM9</td>
</tr>
<tr>
<td>BROWSE</td>
<td>Browse service</td>
<td>ISREMWM3</td>
</tr>
<tr>
<td>CONTROL</td>
<td>Control service</td>
<td>ISREMWM2</td>
</tr>
<tr>
<td>DSINFO</td>
<td>Returns data set information</td>
<td>ISREMWME</td>
</tr>
<tr>
<td>EDIF</td>
<td>Edit interface service</td>
<td>ISREMWMA</td>
</tr>
<tr>
<td>EDIREC</td>
<td>Edit recovery for EDIF</td>
<td>ISREMWMB</td>
</tr>
<tr>
<td>EDIT</td>
<td>Edit service</td>
<td>ISREMWM4</td>
</tr>
<tr>
<td>EDREC</td>
<td>Edit recovery services</td>
<td>ISREMWM7</td>
</tr>
<tr>
<td>GETMSG</td>
<td>Get message service</td>
<td>ISREMWM6</td>
</tr>
<tr>
<td>LIBDEF</td>
<td>LIBDEF service</td>
<td>ISREMWM8</td>
</tr>
<tr>
<td>LIST</td>
<td>Write to list data set</td>
<td>ISREMWMC</td>
</tr>
<tr>
<td>LOG</td>
<td>Write message or log data set</td>
<td>ISREMWM5</td>
</tr>
<tr>
<td>QBASELIB</td>
<td>Query base library information</td>
<td>ISREMWQ2</td>
</tr>
<tr>
<td>QLIBDEF</td>
<td>Query LIBDEF library information</td>
<td>ISREMWQ1</td>
</tr>
<tr>
<td>SELECT</td>
<td>Select service</td>
<td>ISREMWM1</td>
</tr>
<tr>
<td>VIIF</td>
<td>View Interface service</td>
<td>ISREMWMF</td>
</tr>
</tbody>
</table>

**Table functions (general)**

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBCLOSE</td>
<td>Table close</td>
<td>ISREMWG5</td>
</tr>
<tr>
<td>TBCREATE</td>
<td>Table create</td>
<td>ISREMWG1</td>
</tr>
<tr>
<td>TBEND</td>
<td>Table end</td>
<td>ISREMWG6</td>
</tr>
<tr>
<td>TBERASE</td>
<td>Table erase</td>
<td>ISREMWG7</td>
</tr>
<tr>
<td>TBOPEN</td>
<td>Table open</td>
<td>ISREMWG2</td>
</tr>
<tr>
<td>TBQUERY</td>
<td>Table query</td>
<td>ISREMWG3</td>
</tr>
<tr>
<td>TBSAVE</td>
<td>Table save</td>
<td>ISREMWG4</td>
</tr>
<tr>
<td>TBSTATS</td>
<td>Table statistics</td>
<td>ISREMWG8</td>
</tr>
</tbody>
</table>

**Table functions (row)**

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBADD</td>
<td>Table row add</td>
<td>ISREMWR1</td>
</tr>
</tbody>
</table>
### Table 77. DM and PDF services in C/370 programs (continued)

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBBOTTOM</td>
<td>Table row pointer to bottom</td>
<td>ISREMWRRA</td>
</tr>
<tr>
<td>TBDELETE</td>
<td>Table delete</td>
<td>ISREMWR2</td>
</tr>
<tr>
<td>TBEXIST</td>
<td>Table exist</td>
<td>ISREMWR6</td>
</tr>
<tr>
<td>TBGET</td>
<td>Table get</td>
<td>ISREMWR3</td>
</tr>
<tr>
<td>TBMOD</td>
<td>Table modify</td>
<td>ISREMWR5</td>
</tr>
<tr>
<td>TBPUT</td>
<td>Table put</td>
<td>ISREMWR4</td>
</tr>
<tr>
<td>TBSARG</td>
<td>Table search parameter</td>
<td>ISREMWR7</td>
</tr>
<tr>
<td>TBSCAN</td>
<td>Table scan</td>
<td>ISREMWR7</td>
</tr>
<tr>
<td>TBSKIP</td>
<td>Table skip</td>
<td>ISREMWR8</td>
</tr>
<tr>
<td>TBSORT</td>
<td>Table sort</td>
<td>ISREMWRD</td>
</tr>
<tr>
<td>TBTOP</td>
<td>Table top</td>
<td>ISREMWR9</td>
</tr>
<tr>
<td>TBVCLEAR</td>
<td>Table variable clear</td>
<td>ISREMWRC</td>
</tr>
</tbody>
</table>

**Variables**

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>VCOPY</td>
<td>Copy variable</td>
<td>ISREMWV5</td>
</tr>
<tr>
<td>VDEFINE</td>
<td>Variable define</td>
<td>ISREMWV3</td>
</tr>
<tr>
<td>VDELETE</td>
<td>Variable delete</td>
<td>ISREMWV4</td>
</tr>
<tr>
<td>VERASE</td>
<td>Variable erase</td>
<td>ISREMWV4</td>
</tr>
<tr>
<td>VGET</td>
<td>Variable get</td>
<td>ISREMWV8</td>
</tr>
<tr>
<td>VMASK</td>
<td>Variable mask</td>
<td>ISREMWV9</td>
</tr>
<tr>
<td>VPUT</td>
<td>Variable put</td>
<td>ISREMWV2</td>
</tr>
<tr>
<td>VREPLACE</td>
<td>Variable replace</td>
<td>ISREMWV6</td>
</tr>
<tr>
<td>VRESET</td>
<td>Variable reset</td>
<td>ISREMWV7</td>
</tr>
</tbody>
</table>

**Workstation**

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>FILESTAT</td>
<td>Statistics for a file</td>
<td>ISREMWX1</td>
</tr>
<tr>
<td>FILEXFER</td>
<td>Upload or download a file</td>
<td>ISREMWX2</td>
</tr>
<tr>
<td>WSCON</td>
<td>Workstation connection</td>
<td>ISREMWX3</td>
</tr>
<tr>
<td>WSDISCON</td>
<td>Workstation disconnect</td>
<td>ISREMWX4</td>
</tr>
</tbody>
</table>

### Dialog Tag Language models

**Table 78. Dialog Tag Language models**

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Description</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>AB</td>
<td>Action bar</td>
<td>ISREMDAB</td>
</tr>
<tr>
<td>AB (example)</td>
<td>Action bar with pull-down choice</td>
<td>ISREMDAG</td>
</tr>
<tr>
<td>Model Name</td>
<td>Description</td>
<td>Member Name</td>
</tr>
<tr>
<td>------------</td>
<td>------------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>ABC</td>
<td>Action bar choice</td>
<td>ISREMDA2</td>
</tr>
<tr>
<td>ACTION</td>
<td>Action</td>
<td>ISREMDAC</td>
</tr>
<tr>
<td>AREA</td>
<td>Area</td>
<td>ISREMDAR</td>
</tr>
<tr>
<td>ASSIGNI</td>
<td>Assignment List Item</td>
<td>ISREMDAI</td>
</tr>
<tr>
<td>ASSIGNL</td>
<td>Assignment list</td>
<td>ISREMDAL</td>
</tr>
<tr>
<td>ATTENTION</td>
<td>Attention</td>
<td>ISREMDAN</td>
</tr>
<tr>
<td>ATTR</td>
<td>Attribute</td>
<td>ISREMDAT</td>
</tr>
<tr>
<td>BOTINST</td>
<td>Bottom instruction</td>
<td>ISREMDBI</td>
</tr>
<tr>
<td>CAUTION</td>
<td>Caution</td>
<td>ISREMDCU</td>
</tr>
<tr>
<td>CHDIV</td>
<td>Choice divider</td>
<td>ISREMDCD</td>
</tr>
<tr>
<td>AB</td>
<td>Action bar</td>
<td>ISREMDAB</td>
</tr>
<tr>
<td>CHECKI</td>
<td>Validity Check Item</td>
<td>ISREMDCI</td>
</tr>
<tr>
<td>CHECKL</td>
<td>Check validity list</td>
<td>ISREMDCL</td>
</tr>
<tr>
<td>CHOFLD</td>
<td>Choice data field</td>
<td>ISREMDCF</td>
</tr>
<tr>
<td>CHOICE</td>
<td>Selection choice</td>
<td>ISREMDCH</td>
</tr>
<tr>
<td>CMD</td>
<td>Command definition</td>
<td>ISREMDCM</td>
</tr>
<tr>
<td>CMDACT</td>
<td>Command action</td>
<td>ISREMDCCC</td>
</tr>
<tr>
<td>CMDAREA</td>
<td>Command area</td>
<td>ISREMDCAA</td>
</tr>
<tr>
<td>CMDTBL</td>
<td>Command table</td>
<td>ISREMDCCT</td>
</tr>
<tr>
<td>CMDTBL example</td>
<td>Command table example</td>
<td>ISREMDC1</td>
</tr>
<tr>
<td>COMMENT</td>
<td>Comment</td>
<td>ISREMDCN</td>
</tr>
<tr>
<td>COMPOPT</td>
<td>Compiler options</td>
<td>ISREMDCO</td>
</tr>
<tr>
<td>COPYR</td>
<td>Copyright</td>
<td>ISREMDCR</td>
</tr>
<tr>
<td>DA</td>
<td>Dynamic Area</td>
<td>ISREMDDA</td>
</tr>
<tr>
<td>DD</td>
<td>Definition description</td>
<td>ISREMDDD</td>
</tr>
<tr>
<td>DDHD</td>
<td>Definition description header</td>
<td>ISREMDD1</td>
</tr>
<tr>
<td>DIVIDER</td>
<td>Area divider</td>
<td>ISREMDDI</td>
</tr>
<tr>
<td>DL</td>
<td>Definition list</td>
<td>ISREMDDL</td>
</tr>
<tr>
<td>DOCTYPE</td>
<td>Document type</td>
<td>ISREMDTD</td>
</tr>
<tr>
<td>DT</td>
<td>Definition term</td>
<td>ISREMDDT</td>
</tr>
<tr>
<td>DTACOL</td>
<td>Data column</td>
<td>ISREMDDC</td>
</tr>
<tr>
<td>DTAFLD</td>
<td>Data field</td>
<td>ISREMDDF</td>
</tr>
<tr>
<td>DTAFLDD</td>
<td>Data field description</td>
<td>ISREMDDX</td>
</tr>
<tr>
<td>Model Name</td>
<td>Description</td>
<td>Member Name</td>
</tr>
<tr>
<td>-----------</td>
<td>--------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>DTHD</td>
<td>Definition term header</td>
<td>ISREMMDD2</td>
</tr>
<tr>
<td>ENTITY</td>
<td>Entity</td>
<td>ISREMDEN</td>
</tr>
<tr>
<td>FIG</td>
<td>Figure</td>
<td>ISREMDFI</td>
</tr>
<tr>
<td>FIGCAP</td>
<td>Figure caption</td>
<td>ISREMDFC</td>
</tr>
<tr>
<td>GA</td>
<td>Graphic area</td>
<td>ISREMDGA</td>
</tr>
<tr>
<td>GRPHDR</td>
<td>Group header</td>
<td>ISREMDGH</td>
</tr>
<tr>
<td>HELP</td>
<td>Help panel</td>
<td>ISREMDHE</td>
</tr>
<tr>
<td>HELPDEF</td>
<td>Help default</td>
<td>ISREMDHD</td>
</tr>
<tr>
<td>HP</td>
<td>Highlighted phrase</td>
<td>ISREMDHP</td>
</tr>
<tr>
<td>H1</td>
<td>Heading 1</td>
<td>ISREMDH1</td>
</tr>
<tr>
<td>Hn</td>
<td>Heading (H2, H3, H4)</td>
<td>ISREMDHN</td>
</tr>
<tr>
<td>INFO</td>
<td>Information region</td>
<td>ISREMDIN</td>
</tr>
<tr>
<td>KEYI</td>
<td>Key item</td>
<td>ISREMDKI</td>
</tr>
<tr>
<td>KEYL</td>
<td>Key list</td>
<td>ISREMDKL</td>
</tr>
<tr>
<td>L1</td>
<td>List item</td>
<td>ISREMDLI</td>
</tr>
<tr>
<td>LINES</td>
<td>Lines</td>
<td>ISREMDLN</td>
</tr>
<tr>
<td>LIT</td>
<td>Literal</td>
<td>ISREMDLT</td>
</tr>
<tr>
<td>LP</td>
<td>List part</td>
<td>ISREMDLP</td>
</tr>
<tr>
<td>LSTCOL</td>
<td>List column</td>
<td>ISREMDLC</td>
</tr>
<tr>
<td>LSTFLD</td>
<td>List field</td>
<td>ISREMDLF</td>
</tr>
<tr>
<td>LSTGRP</td>
<td>List group</td>
<td>ISREMDLG</td>
</tr>
<tr>
<td>LSTVAR</td>
<td>List variable</td>
<td>ISREMDLV</td>
</tr>
<tr>
<td>M</td>
<td>Mnemonic</td>
<td>ISREMDMN</td>
</tr>
<tr>
<td>MSG</td>
<td>Message</td>
<td>ISREMDMS</td>
</tr>
<tr>
<td>MSGMBR</td>
<td>Message member</td>
<td>ISREMDMM</td>
</tr>
<tr>
<td>NOTE</td>
<td>Note</td>
<td>ISREMDNO</td>
</tr>
<tr>
<td>NOTE2</td>
<td>Note list</td>
<td>ISREMDNL</td>
</tr>
<tr>
<td>NT</td>
<td>Note</td>
<td>ISREMDNT</td>
</tr>
<tr>
<td>OL</td>
<td>Ordered list</td>
<td>ISREMDOL</td>
</tr>
<tr>
<td>Panel example</td>
<td>Panel example</td>
<td>ISREMDP1</td>
</tr>
<tr>
<td>Panel example</td>
<td>Panel example</td>
<td>ISREMDP2</td>
</tr>
<tr>
<td>Panel example</td>
<td>Panel example</td>
<td>ISREMDP3</td>
</tr>
<tr>
<td>Panel example</td>
<td>Panel example</td>
<td>ISREMDP4</td>
</tr>
<tr>
<td>Panel example</td>
<td>Panel example</td>
<td>ISREMDP5</td>
</tr>
<tr>
<td>Model Name</td>
<td>Description</td>
<td>Member Name</td>
</tr>
<tr>
<td>------------</td>
<td>---------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>PANDEF</td>
<td>Panel default</td>
<td>ISREMDPB</td>
</tr>
<tr>
<td>PANEL</td>
<td>Panel</td>
<td>ISREMDPA</td>
</tr>
<tr>
<td>PARML</td>
<td>Parameter list</td>
<td>ISREMDPL</td>
</tr>
<tr>
<td>PD</td>
<td>Parameter description</td>
<td>ISREMDPD</td>
</tr>
<tr>
<td>PDC</td>
<td>Pull-down choice</td>
<td>ISREMDA3</td>
</tr>
<tr>
<td>PDSEP</td>
<td>Pull-down separator</td>
<td>ISREMDA4</td>
</tr>
<tr>
<td>PG</td>
<td>Paragraph</td>
<td>ISREMDPG</td>
</tr>
<tr>
<td>PNLINST</td>
<td>Panel instruction</td>
<td>ISREMDPI</td>
</tr>
<tr>
<td>PS</td>
<td>Point-and-shoot</td>
<td>ISREMDPS</td>
</tr>
<tr>
<td>PT</td>
<td>Parameter term</td>
<td>ISREMDPT</td>
</tr>
<tr>
<td>Region</td>
<td>Region</td>
<td>ISREMDRE</td>
</tr>
<tr>
<td>RP</td>
<td>Reference phrase</td>
<td>ISREMDRP</td>
</tr>
<tr>
<td>SF</td>
<td>Selection field</td>
<td>ISREMDSF</td>
</tr>
<tr>
<td>SL</td>
<td>Simple list</td>
<td>ISREMDSL</td>
</tr>
<tr>
<td>SOURCE</td>
<td>Source</td>
<td>ISREMDSO</td>
</tr>
<tr>
<td>T</td>
<td>Truncation</td>
<td>ISREMDTR</td>
</tr>
<tr>
<td>TOPINST</td>
<td>Top instruction</td>
<td>ISREMDTI</td>
</tr>
<tr>
<td>UL</td>
<td>Unordered list</td>
<td>ISREMDUL</td>
</tr>
<tr>
<td>VARCLASS</td>
<td>Variable class</td>
<td>ISREMDVC</td>
</tr>
<tr>
<td>VARDCL</td>
<td>Variable declaration</td>
<td>ISREMDVD</td>
</tr>
<tr>
<td>VARLIST</td>
<td>Variable list</td>
<td>ISREMDVL</td>
</tr>
<tr>
<td>VARSUB</td>
<td>Variable substitution</td>
<td>ISREMDVS</td>
</tr>
<tr>
<td>WARNING</td>
<td>Warning</td>
<td>ISREMDWA</td>
</tr>
<tr>
<td>XLATI</td>
<td>Translate item</td>
<td>ISREMDXI</td>
</tr>
<tr>
<td>XLATL</td>
<td>Translate list</td>
<td>ISREMDXL</td>
</tr>
<tr>
<td>XMP</td>
<td>Example</td>
<td>ISREMDXM</td>
</tr>
</tbody>
</table>
Chapter 13. Programming interface macros for customers

This topic describes the macros ISPF and SCLM provide as programming interfaces.

Attention: Do not use any ISPF macros other than those described in this topic as programming interfaces.

DM component general-use programming macros

<table>
<thead>
<tr>
<th>ISPCCSID</th>
<th>ISPXMDEF</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISPMDAD</td>
<td>ISPMXED</td>
</tr>
<tr>
<td>ISPMEPT</td>
<td>ISPMXEND</td>
</tr>
<tr>
<td>ISPMTCM</td>
<td>ISPMXLST</td>
</tr>
<tr>
<td>ISPMXDD</td>
<td>ISPDSSTAT</td>
</tr>
</tbody>
</table>


SCLM general-use programming macros

<table>
<thead>
<tr>
<th>FLMABEG</th>
<th>FLMGROUP</th>
</tr>
</thead>
<tbody>
<tr>
<td>FLMAEND</td>
<td>FLMINCLS</td>
</tr>
<tr>
<td>FLMAGRP</td>
<td>FLMLANGL</td>
</tr>
<tr>
<td>FLMALLOC</td>
<td>FLMSYSLB</td>
</tr>
<tr>
<td>FLMALTC</td>
<td>FLMTRNSL</td>
</tr>
<tr>
<td>FLMATVER</td>
<td>FLMTSEX  T</td>
</tr>
<tr>
<td>FLMCNTRL</td>
<td>FLMTYPE</td>
</tr>
<tr>
<td>FLMCOPYLB</td>
<td></td>
</tr>
</tbody>
</table>

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>
<thead>
<tr>
<th>Library</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SISPALIB</td>
<td>APL2 workspace library</td>
</tr>
<tr>
<td>SISPCLIB</td>
<td>CLIST library</td>
</tr>
<tr>
<td>SISPEXEC</td>
<td>REXX exec library</td>
</tr>
<tr>
<td>SISPGDES</td>
<td>Swiss German Language-specific panel source library</td>
</tr>
<tr>
<td>SISPGDEU</td>
<td>German Language-specific panel source library</td>
</tr>
<tr>
<td>SISPGENP</td>
<td>Uppercase English Language-specific panel source library</td>
</tr>
<tr>
<td>SISPGENU</td>
<td>English Language-specific panel source library</td>
</tr>
<tr>
<td>SISPGJPN</td>
<td>Japanese Language-specific panel source library</td>
</tr>
<tr>
<td>SISPGMLI</td>
<td>Non-language-specific panel source library</td>
</tr>
<tr>
<td>SISPGUI</td>
<td>ISPF Client/Server workstation code library</td>
</tr>
<tr>
<td>SISHELP</td>
<td>Help library</td>
</tr>
<tr>
<td>SISPLOAD</td>
<td>Linklist load library</td>
</tr>
<tr>
<td>SISPLPA</td>
<td>LPA load library</td>
</tr>
<tr>
<td>SISPMACS</td>
<td>ISPF Macro library</td>
</tr>
<tr>
<td>SISPMDES</td>
<td>Swiss German Language-specific message library</td>
</tr>
<tr>
<td>SISPMDEU</td>
<td>German Language-specific message library</td>
</tr>
<tr>
<td>SISPMENP</td>
<td>Uppercase English Language-specific message library</td>
</tr>
<tr>
<td>SISPMENU</td>
<td>English Language-specific message library</td>
</tr>
<tr>
<td>SISPMJPN</td>
<td>Japanese Language-specific message library</td>
</tr>
<tr>
<td>SISPPDES</td>
<td>Swiss German Language-specific panel library</td>
</tr>
<tr>
<td>SISPPDEU</td>
<td>German Language-specific panel library</td>
</tr>
<tr>
<td>SISPPENP</td>
<td>Uppercase English Language-specific panel library</td>
</tr>
<tr>
<td>SISPPENU</td>
<td>English Language-specific panel library</td>
</tr>
<tr>
<td>SISPPJPN</td>
<td>Japanese Language-specific panel library</td>
</tr>
<tr>
<td>SISPSAMP</td>
<td>Sample library</td>
</tr>
<tr>
<td>SISPSDES</td>
<td>Swiss German Language-specific skeleton library</td>
</tr>
<tr>
<td>SISPSDEU</td>
<td>German Language-specific skeleton library</td>
</tr>
<tr>
<td>SISPPSEN</td>
<td>Uppercase English Language-specific skeleton library</td>
</tr>
<tr>
<td>SISPSENU</td>
<td>English Language-specific skeleton library</td>
</tr>
</tbody>
</table>
### Table 79. Target data set descriptions (continued)

<table>
<thead>
<tr>
<th>Library</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SISPSJPN</td>
<td>Japanese Language-specific skeleton library</td>
</tr>
<tr>
<td>SISPSLIB</td>
<td>Non-language-specific skeleton library</td>
</tr>
<tr>
<td>SISPTDES</td>
<td>Swiss German Language-specific table library</td>
</tr>
<tr>
<td>SISPTDEU</td>
<td>German Language-specific table library</td>
</tr>
<tr>
<td>SISPTENP</td>
<td>Uppercase English Language-specific table library</td>
</tr>
<tr>
<td>SISPTENU</td>
<td>English Language-specific table library</td>
</tr>
<tr>
<td>SISPTJPN</td>
<td>Japanese Language-specific table library</td>
</tr>
</tbody>
</table>

### Table 80. Distribution data set descriptions

<table>
<thead>
<tr>
<th>Library</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AISPALIB</td>
<td>APL2 workspace library</td>
</tr>
<tr>
<td>AISPCLIB</td>
<td>CLIST library</td>
</tr>
<tr>
<td>AISPEXEC</td>
<td>REXX exec library</td>
</tr>
<tr>
<td>AISPGDES</td>
<td>Swiss German Language-specific panel source library</td>
</tr>
<tr>
<td>AISPGDEU</td>
<td>German Language-specific panel source library</td>
</tr>
<tr>
<td>AISPGENP</td>
<td>Uppercase English Language-specific panel source library</td>
</tr>
<tr>
<td>AISPGENU</td>
<td>English Language-specific panel source library</td>
</tr>
<tr>
<td>AISPGJPN</td>
<td>Japanese Language-specific panel source library</td>
</tr>
<tr>
<td>AISPGMDEU</td>
<td>Non-language-specific panel source library</td>
</tr>
<tr>
<td>AISPGGUI</td>
<td>ISPF Client/Server workstation code library</td>
</tr>
<tr>
<td>AISPHelp</td>
<td>Help library</td>
</tr>
<tr>
<td>AISPMACS</td>
<td>ISPF Macro library</td>
</tr>
<tr>
<td>AISPMDES</td>
<td>Swiss German Language-specific message library</td>
</tr>
<tr>
<td>AISPDEU</td>
<td>German Language-specific message library</td>
</tr>
<tr>
<td>AISPMENP</td>
<td>Uppercase English Language-specific message library</td>
</tr>
<tr>
<td>AISPMENU</td>
<td>English Language-specific message library</td>
</tr>
<tr>
<td>AISPMJPN</td>
<td>Japanese Language-specific message library</td>
</tr>
<tr>
<td>AISPMOD1</td>
<td>Executable modules library</td>
</tr>
<tr>
<td>AISPPDES</td>
<td>Swiss German Language-specific panel library</td>
</tr>
<tr>
<td>AISPPDEU</td>
<td>German Language-specific panel library</td>
</tr>
<tr>
<td>AISPPENP</td>
<td>Uppercase English Language-specific panel library</td>
</tr>
<tr>
<td>AISPPENU</td>
<td>English Language-specific panel library</td>
</tr>
<tr>
<td>AISPPJPN</td>
<td>Japanese Language-specific panel library</td>
</tr>
<tr>
<td>AISPSAMP</td>
<td>Sample library</td>
</tr>
<tr>
<td>Library</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>------------------------------------------------------------------</td>
</tr>
<tr>
<td>AISPSDES</td>
<td>Swiss German Language-specific skeleton library</td>
</tr>
<tr>
<td>AISPSDEU</td>
<td>German Language-specific skeleton library</td>
</tr>
<tr>
<td>AISPSENP</td>
<td>Uppercase English Language-specific skeleton library</td>
</tr>
<tr>
<td>AISPSENU</td>
<td>English Language-specific skeleton library</td>
</tr>
<tr>
<td>AISPSJPN</td>
<td>Japanese Language-specific skeleton library</td>
</tr>
<tr>
<td>AISPSLIB</td>
<td>Non-language-specific skeleton library</td>
</tr>
<tr>
<td>AISPTDES</td>
<td>Swiss German Language-specific table library</td>
</tr>
<tr>
<td>AISPTDEU</td>
<td>German Language-specific table library</td>
</tr>
<tr>
<td>AISPTENP</td>
<td>Uppercase English Language-specific table library</td>
</tr>
<tr>
<td>AISPTENU</td>
<td>English Language-specific table library</td>
</tr>
<tr>
<td>AISPTJPN</td>
<td>Japanese Language-specific table library</td>
</tr>
</tbody>
</table>
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.

**RESERVE processing for partitioned data set extended**

ISPF uses RESERVE to protect data sets residing on a shared volume. However, PDSEs are designed to be shared in a sysplex, so Configuration Table keyword PDSE_RESERVE_PROCESSING can be used to disable the use of RESERVE.

Assuming a PDSE is being processed and ISPF services are used to access the PDSE then:

**PDSE_RESERVE_PROCESSING=ON (default)**

RESERVE issued prior to STOW and released afterwards. LMOPEN with ENQ(SHRW) issues a RESERVE at LMOPEN and OPTION(OUTPUT). If batch job issues RESERVE before an interactive user, then the interactive user waits on SAVE until the batch dialog releases the RESERVE. Another batch job will also wait. See the description of LMCLOSE in *z/OS ISPF Services Guide*.

**PDSE_RESERVE_PROCESSING=OFF**

No RESERVE is issued. DFSMS preserves the integrity of the PDSE. Interactive users and batch jobs can access the PDSE simultaneously. ISPF dialogs continue to use Member name enqueue to serialize access to individual members. In environments where there may be more than one Configuration Table, the dialog that issues a RESERVE will only wait for another dialog that also issues RESERVE. See chapter 3.8.9.3.3 "Choosing Volumes for PDSEs in a Sysplex" in *z/OS DFSMS Using Data Sets*.

**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.
For sequential and PDS data sets, ISPF does a reserve to protect an entire data set on DASD that you are updating through edit, utilities (such as move, copy, delete, and rename), or services (such as TBSAVE, FTCLOSE, FTERASE, EDIT, LMCOPY, LMOVE, and LOPEN depending on whether the option is INPUT or OUTPUT). For PDSE data sets, whether the reserve is done depends on the configuration table setting as discussed in “RESERVE processing for partitioned data set extended” on page 327.

ISPF issues this macro to protect the entire partitioned data set:

```sql
RESERVE SPFEDIT,dsname,E,44,SYSTEMS
```

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:

```sql
RESERVE SYSIEWLP,dsname,E,44,SYSTEMS
```

On a volume that is not shared:

```sql
ENQ SYSIEWLP,dsname,E,44,SYSTEM
```

To restrict concurrent use of a member of a partitioned data set while still allowing ISPF users to concurrently use different members of the same data set (such as through EDIT, Table Processing, File Tailoring, and library access services), ISPF issues this ENQ macro for the member:

```sql
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

**Note:** When a member or a member generation in a PDSE version 2 data set that is configured for member generations is edited, enqueue processing restricts access to the member and all generations of the member.

To restrict concurrent use of a z/OS UNIX file, ISPF edit issues this ENQ macro for the file:

```sql
ENQ SPFEDIT,\%name,E,12,SYSTEMS  (z/OS UNIX in sysplex; OextSysplexActv = '1'b)
```

or

```sql
ENQ SPFEDIT,\%name,E,12,SYSTEM   (z/OS UNIX not in sysplex)
```

where

- **\%name**
  - Is 12 bytes in length, made up of three 4-byte values:
    - A binary integer, which is the inode number of the file.
    - A binary integer, which is the device number of the file.
    - Four flag bytes (0-3).
Bit 2 (of bits 0-7) of byte 3 is the Sysplex indicator. It can have one of these values:

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

Accessible publications for this product are offered through IBM Knowledge Center (www.ibm.com/support/knowledgecenter/SSLTBW/welcome).

If you experience difficulty with the accessibility of any z/OS information, send a detailed email message to mhvrdfs@us.ibm.com.

Accessibility features

Accessibility features help users who have physical disabilities such as restricted mobility or limited vision use software products successfully. The accessibility features in z/OS can help users do the following tasks:

- Run assistive technology such as screen readers and screen magnifier software.
- Operate specific or equivalent features by using the keyboard.
- Customize display attributes such as color, contrast, and font size.

Consult assistive technologies

Consult assistive technology products such as screen readers function with the user interfaces found in z/OS. Consult the product information for the specific assistive technology product that is used to access z/OS interfaces.

Keyboard navigation of the user interface

You can access z/OS user interfaces with TSO/E or ISPF. The following information describes how to use TSO/E and ISPF, including the use of keyboard shortcuts and function keys (PF keys). Each guide includes the default settings for the PF keys.

- z/OS TSO/E Primer
- z/OS TSO/E User's Guide
- z/OS ISPF User's Guide Vol I

Dotted decimal syntax diagrams

Syntax diagrams are provided in dotted decimal format for users who access IBM Knowledge Center with a screen reader. In dotted decimal format, each syntax element is written on a separate line. If two or more syntax elements are always present together (or always absent together), they can appear on the same line because they are considered a single compound syntax element.

Each line starts with a dotted decimal number; for example, $3$ or $3.1$ or $3.1.1$. To hear these numbers correctly, make sure that the screen reader is set to read out punctuation. All the syntax elements that have the same dotted decimal number (for example, all the syntax elements that have the number $3.1$) are mutually exclusive alternatives. If you hear the lines $3.1$ USERID and $3.1$ SYSTEMID, your syntax can include either USERID or SYSTEMID, but not both.

The dotted decimal numbering level denotes the level of nesting. For example, if a syntax element with dotted decimal number $3$ is followed by a series of syntax elements with dotted decimal number $3.1$, all the syntax elements numbered $3.1$ are subordinate to the syntax element numbered $3$. 
Certain words and symbols are used next to the dotted decimal numbers to add information about the syntax elements. Occasionally, these words and symbols might occur at the beginning of the element itself. For ease of identification, if the word or symbol is a part of the syntax element, it is preceded by the backslash (\) character. The * symbol is placed next to a dotted decimal number to indicate that the syntax element repeats. For example, syntax element *FILE with dotted decimal number 3 is given the format 3 \* FILE. Format 3* FILE indicates that syntax element FILE repeats. Format 3* \* FILE indicates that syntax element * FILE repeats.

Characters such as commas, which are used to separate a string of syntax elements, are shown in the syntax just before the items they separate. These characters can appear on the same line as each item, or on a separate line with the same dotted decimal number as the relevant items. The line can also show another symbol to provide information about the syntax elements. For example, the lines 5.1*, 5.1 LASTRUN, and 5.1 DELETE mean that if you use more than one of the LASTRUN and DELETE syntax elements, the elements must be separated by a comma. If no separator is given, assume that you use a blank to separate each syntax element.

If a syntax element is preceded by the % symbol, it indicates a reference that is defined elsewhere. The string that follows the % symbol is the name of a syntax fragment rather than a literal. For example, the line 2.1 %OP1 means that you must refer to separate syntax fragment OP1.

The following symbols are used next to the dotted decimal numbers.

? indicates an optional syntax element
The question mark (?) symbol indicates an optional syntax element. A dotted decimal number followed by the question mark symbol (?) indicates that all the syntax elements with a corresponding dotted decimal number, and any subordinate syntax elements, are optional. If there is only one syntax element with a dotted decimal number, the ? symbol is displayed on the same line as the syntax element, (for example 5? NOTIFY). If there is more than one syntax element with a dotted decimal number, the ? symbol is displayed on a line by itself, followed by the syntax elements that are optional. For example, if you hear the lines 5 ?, 5 NOTIFY, and 5 UPDATE, you know that the syntax elements NOTIFY and UPDATE are optional. That is, you can choose one or none of them. The ? symbol is equivalent to a bypass line in a railroad diagram.

! indicates a default syntax element
The exclamation mark (!) symbol indicates a default syntax element. A dotted decimal number followed by the ! symbol and a syntax element indicate that the syntax element is the default option for all syntax elements that share the same dotted decimal number. Only one of the syntax elements that share the dotted decimal number can specify the ! symbol. For example, if you hear the lines 2? FILE, 2.1! (KEEP), and 2.1 (DELETE), you know that (KEEP) is the default option for the FILE keyword. In the example, if you include the FILE keyword, but do not specify an option, the default option KEEP is applied. A default option also applies to the next higher dotted decimal number. In this example, if the FILE keyword is omitted, the default FILE (KEEP) is used. However, if you hear the lines 2? FILE, 2.1, 2.1.1! (KEEP), and 2.1.1 (DELETE), the default option KEEP applies only to the next higher dotted decimal number, 2.1 (which does not have an associated keyword), and does not apply to 2? FILE. Nothing is used if the keyword FILE is omitted.

* indicates an optional syntax element that is repeatable
The asterisk or glyph (*) symbol indicates a syntax element that can be repeated zero or more times. A dotted decimal number followed by the * symbol indicates that this syntax element can be used zero or more times; that is, it is optional and can be repeated. For example, if you hear the line 5.1* data area, you know that you can include one data area, more than one data area, or no data area. If you hear the lines 3* , 3 HOST, 3 STATE, you know that you can include HOST, STATE, both together, or nothing.

Notes:
1. If a dotted decimal number has an asterisk (*) next to it and there is only one item with that dotted decimal number, you can repeat that same item more than once.
2. If a dotted decimal number has an asterisk next to it and several items have that dotted decimal number, you can use more than one item from the list, but you cannot use the items more than once each. In the previous example, you can write HOST STATE, but you cannot write HOST HOST.
3. The * symbol is equivalent to a loopback line in a railroad syntax diagram.

**+ indicates a syntax element that must be included**

The plus (+) symbol indicates a syntax element that must be included at least once. A dotted decimal number followed by the + symbol indicates that the syntax element must be included one or more times. That is, it must be included at least once and can be repeated. For example, if you hear the line 6.1+ data area, you must include at least one data area. If you hear the lines 2+, 2 HOST, and 2 STATE, you know that you must include HOST, STATE, or both. Similar to the * symbol, the + symbol can repeat a particular item if it is the only item with that dotted decimal number. The + symbol, like the * symbol, is equivalent to a loopback line in a railroad syntax diagram.
Notices

This information was developed for products and services that are offered in the USA or elsewhere.

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 grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
United States of America

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, 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.

This information could include missing, incorrect, or broken hyperlinks. Hyperlinks are maintained in only the HTML plug-in output for the Knowledge Centers. Use of hyperlinks in other output formats of this information is at your own risk.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites 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
Site Counsel
2455 South Road

© Copyright IBM Corp. 1984, 2019
Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

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 illustrate 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. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.
reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

**Rights**
Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

---

**IBM Online Privacy Statement**

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user, or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

Depending upon the configurations deployed, this Software Offering may use session cookies that collect each user's name, email address, phone number, or other personally identifiable information for purposes of enhanced user usability and single sign-on configuration. These cookies can be disabled, but disabling them will also eliminate the functionality they enable.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM's Privacy Policy at ibm.com/privacy and IBM’s Online Privacy Statement at ibm.com/privacy/details in the section entitled “Cookies, Web Beacons and Other Technologies,” and the “IBM Software Products and Software-as-a-Service Privacy Statement” at ibm.com/software/info/product-privacy.

---

**Policy for unsupported hardware**

Various z/OS elements, such as DFSMS, JES2, JES3, and MVS, contain code that supports specific hardware servers or devices. In some cases, this device-related element support remains in the product even after the hardware devices pass their announced End of Service date. z/OS may continue to service element code; however, it will not provide service related to unsupported hardware devices. Software problems related to these devices will not be accepted for service, and current service activity will cease if a problem is determined to be associated with out-of-support devices. In such cases, fixes will not be issued.
Minimum supported hardware

The minimum supported hardware for z/OS releases identified in z/OS announcements can subsequently change when service for particular servers or devices is withdrawn. Likewise, the levels of other software products supported on a particular release of z/OS are subject to the service support lifecycle of those products. Therefore, z/OS and its product publications (for example, panels, samples, messages, and product documentation) can include references to hardware and software that is no longer supported.

- For information about software support lifecycle, see: IBM Lifecycle Support for z/OS (www.ibm.com/software/support/systemsz/lifecycle)
- For information about currently-supported IBM hardware, contact your IBM representative.

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

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at Copyright and Trademark information (www.ibm.com/legal/copytrade.shtml).
Index

A
accessibility
  contact IBM 331
  features 331
action bars
  customizing 37
  extended color in ISPF panels 94
alternate option 7.1 panels 135
APL2 workspace, loading 131
application command table 112
assistive technologies 331

B
BACKUP temporary data sets, preallocating 101
backup, Edit 145
base code pages for terminals 123
batch
  CLIST 173
  modifications 173
  options
    adding to PDF 174
    customizing 149
  processing
    control program (ISRJB1) 171
    flow of control 171
    generated JCL 172
    panels, CLIST, and skeletons 161
    required variables 172
  SCLM 147
  skeletons 173
  suboptions, adding to PDF 174
  using to serialize 327
block size, edit recovery data set 93
Browse
  automatic 159
  customizing 180
  panel
    customizing 180
    output-only variables 183
    PDF invocation of 159
    providing Unicode support 184
build
  Configuration Table Load Module 31
  SMP/E USERMOD 33

C
CCSID
  coded character set identifier 121
  SBCS/DBCS 122
  Unicode support in Browse 184
changing DTL source
  ISPISMMN 117
  ISPOPTxx 119

CLIST
  batch 173
  batch processing 161
  foreground 159
  foreground processing 157
  ISRFPR 159
  panel names 155
  with foreground processing panels 149
CNTL temporary data sets, preallocating 102
code page translation tables, creating 120
coding errors, skeleton, debugging 173
collating sequence translation table (TTCOL) 116
command accounting 106
commands
  reading syntax diagrams x
  compress request exit 209
  compressing listings 103
configuration table
  example 9
  utility, ISPF 9
configuration table, ISPF
  configuration table fields, ISPF
    Data Set Allocation Settings 218
    Default CUA Color Settings 240
    edit recovery data set 220
    Edit Site-wide Profile Customizations 226
    Edit-related Settings 222
    EDIT/BROWSE/VIEW 248
    ISPF Site-wide Profile Customizations 230
    ISPSPROF General Values 249
    ISPSPROF Workstation Defaults 256
    ISPSPROF Workstation Download Defaults 260
    LMF 220
    Miscellaneous Settings 244
    Move/Copy Settings 220
    Outlist data set specifications 218
    PDF Exits 217
    Removable media interface settings 221
    SuperC Data set specifications 219
configuration table, ISPF 9
contact
  z/OS 331
CONVACTB exec 42
Convert Assembler Configuration Table to Keyword File 32
calculating user profile data 120
Create/Modify Settings and Regenerate Keyword File 11
customization options
  ISPF 105
  PDF 143
customizing
  browse and edit 180
  ISPF/PDF product options 187
  ISPTCM 131
  member list panels 184
  z/OS UNIX directory list panel 187
D

daemon name, TCP/IP 131

data set
  allocation exit 194
  allocation settings 218
  integrity enqueue
    DASD 327
    destroying PDSs 328
    member name enqueue 327
    updating data sets 327
  list
    exit 193
    line command exit 203
    utility 203
  name change exit 210
  print 145
  temporary sort data sets 178
Data Set Allocation Settings 218
  default CUA Color Settings 240
  defaults, ISPF site-wide 110
  dialog development model listings
    file tailoring control statements 306
    ISPF Dialog Tag Language panel models 313
    ISPF services
      in CLIST commands 289
      in CLIST programs 313
      in COBOL programs 291
      in EXEC commands 294
      in FORTRAN programs 297
      in Pascal programs 306
      in PLI programs 303
      in TSO/REXX commands 309
    message format 300
    panel formats and statements 300
  Dialog Tag Language Models 316
  disabling high-level qualifiers 99
  DISPLAY service exit 276
  double-byte character set (DBCS) 146
  DLIST removable media interface settings 221
  dynamic area, Edit and Browse panels 182

E

edit
  backup and recovery 145
  customizing 180
  defaults 143
  extended highlighting, disabling 94
  functions, removing or tailoring 93
  keyword file configuration table 29
  panel
    customizing 180
    output-only variables 183
    profile 143
  recovery data set
    changing the block size 93
    ISPF Configuration Table fields 220
    restrictions on 145
    related settings 222
    site-wide profile customizations 226
    terminal output character translation table 283
  error processing for exit routines 142
  execution data sets, allocating 99
  exit macros
    exit data area definitions 263
    exit entry definitions 263
    ISPMDAD 264
    ISPMEPT 263
    ISPMXDD 264
    ISPMXDEF 263
    ISPMXED 263
    ISPMXEND 264
    ISPMXLST 263
  exit parameter list 141
  exits 137, 193
  expansion triggers 178
  extended
    code pages
      adding translation tables for 124
      supported by ISPF 121
      translation tables 125
    color in Edit 144

F

feedback xvii

file tailoring
  allocating 37
  control statements, model listings for 306
  image library 37
  FMID for USERMOD 33
  foreground
    CLIST 159
    ISRFPR 159
  modifications 160
  options
    adding new option to PDF 160
    adding new suboption to PDF 160
    customizing 149
    invoking ISRFPR 157
    unauthorized commands 160
  processing
    flow of control 156
    panel and CLIST names 155
    required variables 157
  FORTRAN and COBOL interactive CLIST 159
  fragments, syntax diagrams x

G

gateway, TSO/ISPF
  customizing 70
  description 46, 69, 82
  HTTP server, customizing 82
  HTTP server, verifying customization 83, 87
  installing 70
  other environments, configuration for 89
  Remote Systems Explorer, customizing 89
  using 71
Generalized Markup Language (GML) 37
  generic string
    character translation table 283
    master translation table 114, 284, 286
    special character translation table 284
H

hard copy utility, using DBCS data sets 146
hexadecimal character translation table 281
high-level qualifiers, disabling 99
HILITE 94

I

IEEMB846 module 107
image library, allocating 37
initialization, site-wide edit profile 143
installation exits
  DISPLAY service 270, 276
  ISPF session
    initialization 264
    termination 265
  LIBDEF service 270
log, list, and temporary data set allocation 277
logical screen
  end 274
  start 273
PDF service
  end 275
  start 274
RELEASE 273
RESERVE 272
SELECT service 267
SWAP 275
TSO command
  end 270
  start 269
installation-wide exits
  activity monitoring exits 198, 273
  calling the exit routine 210
  common parameter list 141
  compress request exit 209
data set exits
  allocation 194
  list 193
  list line 203, 204
  name change 210
error processing 142
exit
  definition table 139
  macros 263
  parameters 142, 210
installing 138
  possible uses for the routine 137
print utility exit 204
SELECT service 265
TSO/E information center facility 205
variables 204, 209
interactive CLIST, FORTRAN, and COBOL 159
invalid data set name character translation table 281
IPV6 3
ISPAFTTT, APL2 terminal type dialog 130
ISPCCSID translation load module
description 124
  generation macro 126
ISPCFIG 9
ISPEEXEC, external entry point in ISPLINK 129
ISPEXITS, sample JCL 139
ISPF

ISPF (continued)

APL2 terminal type mappings 130
customizing
  DM 105
  ISPTCM 131
data set integrity enqueue 327
default table, generating 131
installation-wide exits 137
ISPPREP preprocessed panel utility 95
list data set allocation 109
load modules 5
log data set allocation 109
module names 107
preprocessed panel utility 95
session initialization exit 264
session termination exit 265
system command table, overriding commands 111
tailoring defaults 131
terminal translation table, ISPF/PDF 265, 287
translation tables, creating 112
using with modified batch options 149
using with modified foreground options 149
ISPF client/server
  changing default name 106
  Prohibiting connections to the ISPF Workstation Agent 105
ISPF client/server feature 4
ISPF configuration table 111
ISPF Configuration Table 9
ISPF configuration table fields
  Data Set Allocation Settings 218
  Default CUA Color Settings 240
  DLIST interface settings 221
  edit recovery data set 220
  Edit Site-wide Profile Customizations 226
  Edit-related Settings 222
  EDIT/BROWSE/VIEW 248
  ISPF Site-wide Profile Customizations 230
  ISPSPROF General Values 249
  ISPSPROF Workstation Defaults 256
  ISPSPROF Workstation Download Defaults 260
  LMF 220
  Miscellaneous Settings 244
  Move/Copy Settings 220
  Outlist data set specifications 218
  PDF Exits 217
  Removable media interface settings 221
  SuperC Data set specifications 219
ISPF configuration utility
  Build Configuration Table Load Module 31
  Build SMP/E USERMOD 33
  Convert Assembler Configuration Table to Keyword File 32
  Create/Modify Settings and Regenerate Keyword File 11
  Edit Keyword File Configuration Table 29
  Verify Keyword Table Contents 30
ISPF Dialog Tag Language panel models 313
ISPF services
  in C/370 programs 313
  in CLIST commands 289
  in COBOL programs 291
  in EXEC commands 294
  in FORTRAN programs 297
  in Pascal programs 306
ISPF services (continued)
in PLI programs 303
in TSO/REXX commands 309
ISPF Site-wide Profile Customizations 230
ISPF termination and LOG/LIST commands, Print utility exit 206
ISPF/PDF
  installation-wide exits 193
  Product Option customizing 187
ISPILIB 7
  ISPILIB image input library 37
  ISPILIB in logon proc 7
ISP LIST, default values for 110
ISPMDAD macro 264
ISPMEPT macro 263
ISPMTCM macro
  how to use 131
  sample definition 133
  syntax 131
ISPMXDD macro 264
ISPMXDEF macro 263
ISPMXED macro 263
ISPMXLST macro 263
ISPPREP preprocessed panel utility 95
ISPTCM (TSO command table)
  ISPMTCM macro 131
  sample definition 133
  usage notes 134
ISPTTDEF 120
ISPXDT (exit definition table)
  how to define 139
  main entry point of ISPEXITS 137
  sample definition 140
ISR@PRIM, PDF Primary Option Menu 149
ISRBROB, distributed Browse panel 180
ISREDIT, edit profiles set 144
ISREDRT, Edit backup and recovery table 145
ISREFR01, distributed Edit panel 180
ISRFPR 157
ISRJB1
  batch processing control program 171
  jobnames algorithm 172
ISRJB2 172
ISRLEMX
  customizing 175
  job steps 176
  parameters 177
  return codes 176
ISRLWKnn, temporary data sets 178
ISRSCAN
  customizing 175
  job steps 175
  return codes 175
ISRUOL, print data set attributes 146

K

keyboard
  navigation 331
  PF keys 331
  shortcut keys 331
Keyword File, Configuration Table 217
keywords, syntax diagrams x

L

language default value, changing for session 136
Language Environment 4
LIBDEF service exit 134, 270
line mode support, setup 106
link-pack area 96
LINKLST 96
list data set
  default values 110
  preallocating to SYSOUT 110
LIST temporary data sets, preallocating 101, 102
listings, compressing 103
load modules
  adding to the LPA 96
  descriptions 97
  moving 97
  recommended for the LPA 97–99
  search order 5
loading APL2 workspace 131
LoadMod
  converted to keyword file 35
log
  data set, preallocating to SYSOUT 110
  list, and temporary data set allocation exit 277
logical screen exit
  end 274
  start 273
LOGON procedure, TSO
  list and temporary data set allocation exit 277
lowercase character translation table (TTLOW) 113, 285
LPA 96
LPALST 96

M

macro names, complete list 321
mappings, ISPF-to-APL2 terminal type 130
member
  expansion function, adding user-defined trigger 178, 179
  list built-in line command exit 214
  list filter exit
    analyze member names 213
    analyze patterns 212
  list panels, customizing 184
  parts list function, adding user-defined trigger 179
menu
  PDF Primary Option Menu 149
message format, model listings for 300
Miscellaneous Settings 244
model listings, complete list of
  file tailoring control statements 306

J

JCL generated by batch processing 172
jobnames, algorithm for uniqueness 172
model listings, complete list of (continued)
ISPF Dialog Tag Language panel models 313
ISPF services
  in C/370 programs 313
  in CLIST commands 289
  in COBOL programs 291
  in EXEC commands 294
  in FORTRAN programs 297
  in Pascal programs 306
  in PLI programs 303
  in TSO/REXX commands 309
message format 300
panel formats and statements 300
SCLM
  architecture definition formats 312
  project definition macros and templates 312
model names, complete list 289
Models, Dialog Tag Language 316
Move/Copy Settings 220
multicultural support
  installation 136
  maintenance considerations 5
multivolume data sets
  edit recovery data set 145
  temporary data set 103

N
National Language Support, See multicultural support
navigation
  keyboard 331
NLS, See multicultural support
non-ISPF TSO and BATCH 327
numeric character translation table 282

O
option
  IBM products 187
  panel, redisplay of in foreground processing 159
optional ISPF libraries 37
Outlist data set specifications 218
OUTLIST temporary data sets, preallocating 101
outlist utility, allocation defaults for 145

P
panels
  batch processing 161
  display sequence, changing 159, 161
  Edit and Browse, customizing 180
  foreground processing 149
  formats and statements, model listings for 300
parameter list for exit routines 141
partitioned data set
  RESERVE processing 327
PDF
  automatic browse and display, bypassing 159
  batch, customizing 149
  customizing 143
  FORTRAN Interactive Debug option 157
  installation-wide exits 193
  Primary Option Menu 149
PDF (continued)
  sample assembler modules 148
  service end exit 275
  service start exit 274
  terminal translation table, ISPF/PDF 287
  translation tables, creating 148
  VS COBOL II Interactive Debug 157
PDF Exits 217
performance, improving
  adding load modules to the LPA 96
  changing the block size of the edit recovery data set 93
  disabling edit extended highlighting 94
  disabling SETUNDO storage 94
  preprocessing ISPF panels 95
  removing action bars from ISPF panels 94
  removing scrollable areas from ISPF panels 95
  VIO 93
planning
  ISPF installation 3
  preprocessing ISPF panels 95
Primary Option Menu
  PDF 149
print
  data set attributes 146
  function 159
  utility exit 204
Print utility exit
  ISPF termination and LOG/LIST commands 206
product option 187
profile
  Edit 143
  sharing, customizing 43
programming
  interface macros for customers 321

R
recovery
  data set, restrictions on 145
  Edit 145
redisplay of option panel, modifying 159
RELEASE exit 273
removing edit functions 93
repeatable items, syntax diagrams x
requirements
  system 3
RESERVE exit 272
RESERVE processing
  partitioned data set extended 327
Resource Access Control Facility (RACF)
  protects SCLM project data sets 4
return codes
  ISRLEMX 176
  ISRSCLM 175
RTNPNL, PDF batch processing variable 172

S
SBCS/DBCS CCSIDs 122
SCEERUN run-time library data sets 4
SCLM
  architecture definition formats 312
SCLM (continued)
batch considerations 147
listings, compressing 103
project definition macros and templates 312
scrollable areas, removing from ISPF panels 95
search sequence for attaching commands 134
SELECT service
end exit 267
start exit 265
sending to IBM
reader comments xvii
serializing 327
SETUNDO STORAGE, disabling 94
shared variable pool, foreground processing 158
shortcut keys 331
site command tables 112
site-wide
defaults, setting
CUA panel elements 110
KEYLIST 110
PFSHOW 110
SCROLL AREA 110
STATUS 110
edit profile initialization 143
skeletons, batch processing 161, 173
SMF 106
SMP/E
++MOVE statements 97
data set 33
USERMOD 33
software components used by SCLM 4
Software Configuration and Library Manager (SCLM)
migrating from LMF
data changes 4
library structure changes 4
project
accounting data 312
database 4
definition and control, contained in PROJDEFS 4
project accounting data 4
project cross-reference 4
RACF to protect SCLM project data sets 4
SCLM project definition macros and templates 312
started task 4
VSAM clusters 4
SPFTEMP temporary data sets, preallocating 101
split screens, specifying number of 110
square brackets, displaying in C programs 129
STATUS AREA 110
SUBMIT command 172
suboption panel, redisplay of, in batch processing 173
suboptions
PDF batch option 166
PDF foreground option 152
summary of changes xix
SuperC Data set specifications 219
SVCs
SVC 109 106, 107
SWAP exit 275
syntax diagrams, how to read x
SYSLIB 34
SYSMOD Identifier for USERMOD 33
system
command table 112
system (continued)
requirements 3
System Management Facility (SMF)
command accounting 106
description 106
T
tailoring
edit functions 93
ISPF defaults 131
PDF defaults 9
tape data sets, configure support for 221
TCP/IP daemon name 131
temporary data set names 44
temporary data sets
default names 101
preallocating 101
restrictions on 103
terminal type
associating with translation tables 116
mappings, ISPF-to-APL2 130
terminals, base code pages for 123
trademarks 338
translation load
module generation macro, ISPCCSID 126
modules, ISPCCSID 124
translation tables
alphabetic character (TTALB) 115
alphabetic character (TTALP) 115
alphanumeric character 282
collating sequence (TTCOL) 116
creating for ISPF 112
creating for ISPF/PDF 148edit terminal output character 283
generic string character
translation table 283
usage notes 284
generic string master (TTGSM) 114
generic string special character
translation table 284
usage notes 284
hexadecimal character 281
invalid data set name character 281
lowercase character (TTLOW) 113, 285
master, generic string 286
modifying GSM 285
numeric character 282
sample assembler modules 148
uppercase character (TTUPP) 113, 285
valid data set name character 281
valid terminal output (TTVAL) 114
trigger, user-defined
adding to member expansion function 178
adding to member parts list function 179
TSO
command
end exit 270
table (ISPTCM) 131
line mode 7
line mode support setup 106
logon procedure example 5, 7
SUBMIT command 172
TSO line mode restriction 7
TSO/ISPF client gateway,  
  description 46, 69, 82  
TSO/ISPF gateway  
  customizing 70  
  HTTP server, customizing 82  
  HTTP server, verifying customization 83, 87  
  installing 70  
  other environments, configuration for 89  
Remote Systems Explorer, customizing 89  
  using 71

U

Unicode  
  providing support in Browse 184  
  uppercase character translation table  
    ISPF 113  
    ISPF/PDF 285  
user command tables 112  
user interface  
  ISPF 331  
  TSO/E 331  
user-defined trigger  
  batch options 179  
  foreground options 178  
  ISRLEMX parameter description 177  
  member expansion function 178  
  member parts list function 179  
usermod 97  
utility  
  hard copy, with DBCS support 146  
  ISPF configuration 9  
  outlist (3.8), changing data set allocation defaults 145  
  preprocessed panel utility (ISPPREP) 95

V

valid  
  data set name character translation table 281  
  terminal output translation table (TTVAL) 114  
variables, syntax diagrams x  
Verify Keyword Table Contents 30  
VIO 93  
virtual I/O 93  
VSAM clusters  
  SCLM accounting data sets 4  
  SCLM project cross-reference 4

W

wildcard characters 99  
WORK temporary data sets, preallocating 102

Z

z/OS UNIX directory list panel, customizing 187  
z/OS UNIX Directory List utility 4  
ZDEFAULT  
  default edit profile 144  
  profile, creating 144  
ZDSCKO, PDF batch processing variable 172  
ZFBROWS, foreground automatic browse variable 159  
ZFPRINT, foreground automatic print variable 159

ZNEXTPN, controlling panel display sequence  
  batch 173  
  foreground 159  
ZSEL, foreground processing restriction for 157