Interactive System Productivity Facility (ISPF)

Planning and Customizing

z/OS Version 1 Release 7.0

GC34-4814-04

Interactive System Productivity Facility (ISPF)

Planning and Customizing

z/OS Version 1 Release 7.0

GC34-4814-04

Note
Before using this document, read the general information under "Notices" on page 285.

Fifth Edition (September 2005)

This edition applies to ISPF for Version 1 Release 7.0 of the licensed program z/OS (program number 5694-A01) and to all subsequent releases and modifications until otherwise indicated in new editions.

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

IBM Corporation
Department J87/D325
555 Bailey Avenue
San Jose, CA 95141-1003
U.S.A.

Internet: comments@us.ibm.com

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

Make sure to include the following in your comment or note:
- Title and order number of this document
- Page number or topic related to your comment

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

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

Contents

Preface

This document provides planning and customizing information for the Interactive System Productivity Facility (ISPF) product. ISPF 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 manual contains detailed information you need to:

- Plan, install, and customize ISPF under z/OS with the Time Sharing Option Extensions (TSO/E)
- Select installation options
- Modify the distributed release

It also provides information about tailoring menus and options, and the online tutorial.

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 the following chapters and appendixes:

- Chapter 1, "z/OS V1R7.0 ISPF Planning," on page 1 describes planning procedures you need to know before installing ISPF. It also identifies the hardware and software requirements for running ISPF.
- Chapter 2, "Customizing ISPF," on page 7 describes how to modify items that affect both the DM and PDF components to suit the needs of your installation.
- Chapter 3, "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 4, "Customizing PDF," on page 161 describes how to modify the distributed release of PDF to suit the needs of your installation.
- Appendix A, "ISPF Enqueue Processing for Data Integrity," on page 247 explains how ISPF ensures data integrity.
- Appendix B, "Dialog Development Model Listings," on page 251 lists all of the models shipped with PDF, any qualifiers, a short description, and the internal member name in the SKELS library shipped with PDF.
- Appendix C, "Programming Interface Macros For Customers," on page 277 lists the macros provided by ISPF as programming interfaces.
- Appendix D, "ISPF Data Set Descriptions," on page 279 lists the target and distribution data sets used by ISPF, and their contents.

Using LookAt to look up message explanations

LookAt is an online facility that lets you look up explanations for most of the IBM® messages you encounter, as well as for some system abends and codes. Using LookAt to find information is faster than a conventional search because in most cases LookAt goes directly to the message explanation.

You can use LookAt from these locations to find IBM message explanations for z/OS® elements and features, z/VM®, VSE/ESA™, and Clusters for AIX® and Linux™:

- The Internet. You can access IBM message explanations directly from the LookAt Web site at http://www.ibm.com/servers/eserver/zseries/zos/bkserv/lookat/.
- Your z/OS TSO/E host system. You can install code on your z/OS or z/OS.e systems to access IBM message explanations using LookAt from a TSO/E command line (for example: TSO/E prompt, ISPF, or z/OS UNIX® System Services).
- Your Microsoft® Windows® workstation. You can install LookAt directly from the *z/OS Collection* (SK3T-4269) or the *z/OS and Software Products DVD Collection* (SK3T4271) and use it from the resulting Windows graphical user interface (GUI). The command prompt (also known as the DOS > command line) version can still be used from the directory in which you install the Windows version of LookAt.
- Your wireless handheld device. You can use the LookAt Mobile Edition from http://www.ibm.com/servers/eserver/zseries/zos/bkserv/lookat/lookatm.html with a handheld device that has wireless access and an Internet browser (for example: Internet Explorer for Pocket PCs, Blazer or Eudora for Palm OS, or Opera for Linux handheld devices).

You can obtain code to install LookAt on your host system or Microsoft Windows workstation from:

- A CD-ROM in the *z/OS Collection* (SK3T-4269).
- The *z/OS and Software Products DVD Collection* (SK3T4271).
- The LookAt Web site (click **Download** and then select the platform, release, collection, and location that suit your needs). More information is available in the LOOKAT.ME files available during the download process.

Using IBM Health Checker for z/OS

IBM Health Checker for z/OS is a z/OS component that installations can use to gather information about their system environment and system parameters to help identify potential configuration problems before they impact availability or cause outages. Individual products, z/OS components, or ISV software can provide checks that take advantage of the IBM Health Checker for z/OS framework. This book refers to checks or messages associated with this component.

For additional information about checks and about IBM Health Checker for z/OS, see *IBM Health Checker for z/OS and Sysplex: User's Guide*. z/OS V1R4, V1R5, and V1R6 users can obtain the IBM Health Checker for z/OS from the z/OS Downloads page at http://www.ibm.com/servers/eserver/zseries/zos/downloads/.

SDSF also provides functions to simplify the management of checks. See *z/OS SDSF Operation and Customization* for additional information.

Summary of Changes

z/OS V1R7.0 ISPF contains the following changes and enhancements:
- ISPF product and library changes
- ISPF Dialog Manager component changes (including DTL changes)
- ISPF PDF Component changes
- ISPF SCLM component changes
- ISPF Client/Server component changes

ISPF product and library changes

Changes to the ZENVIR variable. Characters 1 through 8 contain the product name and sequence number, in the format ISPF *x.y*, where *x.y* indicates the version number and release. Note that the *x.y* value is not necessarily the same as the operating system version. For example, a value of ″ISPF 5.7″ represents ISPF for z/OS Version 1 Release 7.0.

The ZOS390RL variable contains the level of the z/OS release running on your system.

The ZISPFOS system variable contains the level of ISPF that is running as part of the operating system release on your system. This might or might not match ZOS390RL. For this release of ISPF, the variable contains `ISPF for z/OS 01.07.00`.

ISPF Dialog Manager component changes

The DM component of ISPF includes the following new functions and enhancements:

- A new optional keyword SFIHDR on the)MODEL statement provides support for defining scroll field indicators in the model section of a panel.
- A new ISPFTTRC command supports tracing of File Tailoring service calls (FTOPEN, FTINCL, FTCLOSE, and FTERASE) and processing.
- A new ISPDPTRC command supports tracing of Panel Service calls (DISPLAY, TBDISPL, and TBQUERY) and the processing of statements within the)ABCINIT,)ABCPROC,)INIT,)REINIT, and)PROC sections of the panel.

ISPF Configuration Utility changes:

- New DEFAULT_LIBDEF_PROCESSING_OPTION keyword allows setting of the default LIBDEF processing option.
- New DEFAULT_SYSTEM_NAME keyword controls displaying of the current system name.
- New DEFAULT_USERID_DISP keyword controls displaying of the current user ID.

Dialog Tag Language (DTL) changes:

- There are no changes to Dialog Tag Language (DTL) for this release.

ISPF PDF Component changes

The ISPF PDF component contains the following new functions and enhancements:

- New ISPF Table Utility enables users to work with data in an ISPF table in full-screen mode, using the Browse and Edit functions.
- Support for large format sequential data sets (to a theoretical maximum of 16 million tracks per volume). Users can Browse, View, and Edit large format sequential data sets, as well as Move and Copy data to and from them.
- Enhanced sort processing for member lists and data set lists. Users can now specify the sort direction (A=ascending, D=descending).
- New SYSNAME command to display the current system name.
- New USERID command to display the current user ID.
- Because reported space utilization data is not meaningful for BDAM data sets, question marks (?) are now returned in these fields in option 3.4 and the DSINFO and LMDLIST calls.
- The DSINFO service has been enhanced to return two ISPF function pool variables, ZDSAPF and ZDSLNK, which give the APF and LINKLST status of a data set.
- The DSINFO service has been enhanced to return two ISPF function pool variables. ZDSTOTAX contains the number of bytes allocated to a data set. ZDSTOTUX contains the number of bytes used by a data set. These variables are long enough to cater for the theoretical maximum size of a large format sequential data set.
- The LIBDEF service has been enhanced to use the default LIBDEF processing option set in the ISPF configuration table if one is not specified.
- The LIBDEF service has been enhanced to give a return code of 4 when a STKADD processing option is requested and no current stack exists for the specified lib-type.
- An optional keyword, NOLLA, has been added to the LMMFIND, LMMSTATS, and LMPRINT services. If LLA is used to manage a PDS directory, this keyword ensures that the cached entry is not used.
- Documentation has been added for the VIRTSIZE keyword on the TBSTATS service.
- HTML and XML have been added to the list of languages that can automatically be identified by the EDIT HILITE function.
- The BROWSE DISPLAY command now allows data stored on the mainframe using a Unicode CCSID to be displayed using the terminal CCSID.
- The BROWSE FIND function now allows users to search for a UTF8, ASCII, or USASCII character string.
- 64-bit register values are now displayed for sub-task dumps.

ISPF SCLM component changes

The ISPF SCLM component contains the following new functions and enhancements:

- If a language change is done for an SCLM part via the SPROF command, the PDS directory is now updated with the user ID.
- The Delete Group function has been renamed Delete from Group to make it more clear what it does.

ISPF Client/Server component changes

The ISPF Client/Server Component enables a panel to be displayed unchanged (except for panels with graphic areas) at a workstation using the native display function of the operating system of the workstation. ISPF documents call this ″running in GUI mode.″

There are no changes to the ISPF Client/Server for this release.

ISPF migration considerations

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

Note

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

Starting with z/OS V1R2, you may notice changes in the style and structure of some content in this book—for example, headings that use uppercase for the first letter of initial words only, and procedures that have a different look and format. The changes are ongoing improvements to the consistency and retrievability of information in our books.

What's in the z/OS V1R7.0 ISPF library?

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

Title	Order Number
z/OS ISPF Dialog Developer's Guide and Reference	SC34-4821–04
z/OS ISPF Dialog Tag Language Guide and Reference	SC34-4824–04
z/OS ISPF Edit and Edit Macros	SC34-4820–04
z/OS ISPF Messages and Codes	SC34-4815–04
z/OS ISPF Planning and Customizing	GC34-4814–04
z/OS ISPF Reference Summary	SC34-4816–04
z/OS ISPF Software Configuration and Library Manager Project Manager's and Developer's Guide	SC34-4817–04
z/OS ISPF Software Configuration and Library Manager Reference	SC34-4818–04
z/OS ISPF Services Guide	SC34-4819–04
z/OS ISPF User's Guide Vol I	SC34-4822–04
z/OS ISPF User's Guide Vol II	SC34-4823–04

Chapter 1. z/OS V1R7.0 ISPF Planning

Application dialogs that were created to run under ISPF Version 1, Version 2, Version 3, or Version 4 will run under z/OS V1R7.0 ISPF with no change.

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

Hardware and Software Requirements

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

Migrating from Previous Releases to z/OS V1R7.0 ISPF

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

It is possible to migrate to z/OS V1R7.0 ISPF from all previous releases of ISPF and ISPF/PDF or from SPF.

Customization of ISPF had major changes in OS/390® Version 2 Release 8.0. When migrating from a version of OS/390 earlier than V2R8.0 you must convert your ISPF customization to the new format. See "The ISPF Configuration Table" on page 7 for more information.

If you currently have a previous version of ISPF (5685-054 or 5665-319) and ISPF/PDF (5665-402 or 5665-317) or SPF (5668-009) installed, you can have the old programs and the new programs available simultaneously for some period of time.

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

Attention: If SPF and ISPF are both installed on the system, there is a danger of destroying partitioned data sets (PDSs) that are being updated. This will happen if the same data set is updated by SPF and ISPF at the same time.

Use the following information to help you during the transition stage.

- The z/OS V1R7.0 ISPF data sets are all named ISP.SISP*xxxx* and ISP.AISP*xxxx*.

  The previous releases of Version 4 ISPF, Version 3 ISPF, and ISPF/PDF data sets were named ISP.V4RxM0.*xxxxx*, ISP.V3RxM0.*xxxxx*, and ISR.V3RxMx.*xxxx*, respectively.
- Using SMP to install z/OS V1R7.0 ISPF deletes the following licensed programs:
  - ISPF Version 3.*n* and ISPF/PDF Version 3.*n*
  - ISPF Version 4.*n*
  - OS/390 V1R3.0 ISPF
  - OS/390 V2R5.0 ISPF
  - OS/390 V2R8.0 ISPF

  - OS/390 V2R10.0 ISPF
  - z/OS V1R2.0 ISPF
  - z/OS V1R5.0 ISPF
  - z/OS V1R6.0 ISPF

  During the transition stage, maintain the previous programs on a separate set of back-up packs or in a set of libraries that does not interfere with the execution of z/OS V1R7.0 ISPF.
- During the transition stage, it is assumed that the predecessor programs reside in the system pageable link pack area. ISPF and the new programs are assumed to be running from a set of execution libraries, as described in the *z/OS Program Directory*.
- Many of the modules in ISPF have the same names as modules in the previous licensed programs. Therefore, the ISPF module libraries should not be in your system LNKLSTxx definition during the transition period.
- During the initial testing period, ISPF should be run from the SISPLOAD and SISPLPA data sets created during the SMP APPLY processing step. Use a STEPLIB DD statement in the TSO LOGON procedure to accomplish this. Include SISPSASC in the STEPLIB if you are using the ISPF client/server feature.

  If you are also running with an ISPLLIB DD statement, you must include the z/OS V1R7.0 ISPF SISPLOAD and SISPLPA data sets in the ISPLLIB concatenation and remove any Load Library data sets from a previous version or release of ISPF.
- If only the top modules of z/OS V1R7.0 ISPF (ISPICP and alias ISPSTART as well as ISRPCP and alias entries PDF and ISPF) are in the step libraries, and the other ISPF 4.2 and ISPF Version 4.1 load modules are in libraries allocated to the ISPLLIB ddname, both the old programs and the new programs can be executed concurrently under the same LOGON procedure. Concurrent execution assumes that a CLIST is used to provide the allocations of the other required ISPF libraries.
- *Do not* use the command names ISPF, PDF, or ISPSTART during the transition stage. Conflicts can occur if you invoke SPF, ISPF and ISPF/PDF Version 2, and ISPF and ISPF/PDF Version 3 modules with the same names. One solution is to rename the alias entries of the top ISPF modules in the step library to temporary names (for example, from ISPF to NEWISPF) until the transition stage is complete.
- After the transition stage, the z/OS V1R7.0 ISPF LPA-eligible load modules (data set SISPLPA) should be put in LPA, and the LPA-ineligible load modules (data set SISPLOAD) should be put in LNKLST. See “Improving ISPF Performance” on page 74 for more information. At that time, remove any DD statements for a STEPLIB or ISPLLIB definition of the ISPLOAD/ISPLPA data set from the LOGON procedure.
- Panel ISRLOGO was renamed to ISR@LOGO at z/OS Version 1 Release 1.0. If you have a modified version of ISR@PRIM you should check the panel value used in the HELP Action Bar Pull-down menu.

SAS/C Considerations

The ISPF client/server (ISPF C/S) feature includes some code compiled with the SAS/C C++ Compiler for MVS. ISPF redistributes the SAS/C transient library in the ISP.SISPSASC library. This library must be in STEPLIB or LNKLST for the modules to be loaded at execution time. Using the usual ISPLLIB allocation does not locate these modules.

If you have the SAS/C Compiler for MVS installed on your system or have the SAS/C transient library installed on your system because it was redistributed with another product, it is important that you use the correct level of the transient library. You should use the highest release and maintenance level of the transient library for all code compiled with the SAS/C compiler, unless a product requires a particular level of the transient library. ISPF C/S is intended to run with the level supplied in the SISPSASC library or higher. The ISP.SISPSAMP library contains a member (ISPFSASC) that describes the release and maintenance level of the SAS/C compiler and resident and transient libraries used for this release of ISPF.

Install the ISPF SISPSASC library (or another copy of the SAS/C transient library if you have a copy installed that is at a later release or maintenance level) in STEPLIB or LNKLST. If you have multiple copies of the SAS/C transient library be sure to use the latest level, as problems may result from using an earlier level than that for which a product is written.

Software Configuration and Library Manager (SCLM)

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

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

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

For information about converting from LMF to SCLM see “Converting From LMF To SCLM” on page 168. For detailed information about SCLM, refer to *z/OS ISPF Software Configuration and Library Manager Project Manager's and Developer's Guide*.

Maintenance Considerations

ISPF is packaged using the features of SMP/E that allow multiple NLS 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 NLS language features in one target zone and one distribution zone. This simplifies the installation of maintenance.

If you do not install your NLS 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 NLS 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 NLS features, you must use the following procedure:

1. APPLY the base PTF that is REQed by the NLS feature PTF or contains an IF REQ for an NLS feature PTF.

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

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

Load Module Search Order

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

The exception to this search order is the SISPSASC library. Modules in it are not searched for by using the ISPLLIB task library. SISPSASC must be in STEPLIB or LNKLST if you are using the ISPF C/S feature. See "SAS/C Considerations" on page 2 for more information about SISPSASC.

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

TSO Logon Procedure

Figure 1 on page 5 is a sample TSO logon procedure (logon proc) to allocate the ISPF data sets required to execute ISPF. The default names ISP.SISP*yxxx* are used for the ISPF data sets and *xxx* represents the national language designators as follows:

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

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

```
//ISPF     PROC
//ISPF     EXEC PGM=IKJEFT01,DYNAMNBR=100
//*
//STEPLIB  DD DSN=ISP.SISPLPA,DISP=SHR
//         DD DSN=ISP.SISPLOAD,DISP=SHR
//         DD DSN=ISP.SISPSASC,DISP=SHR
//*
//ISPMLIB  DD DSN=ISP.SISPMxxx,DISP=SHR
//*
//*SYSHELP DD DSN=ISP.SISPHELP,DISP=SHR
//*
//ISPPLIB  DD DSN=ISP.SISPPxxx,DISP=SHR
//*
//ISPSLIB  DD DSN=ISP.SISPSxxx,DISP=SHR
//         DD DSN=ISP.SISPSLIB,DISP=SHR
//*
//ISPTLIB  DD DSN=userid.ISPTABLE,DISP=SHR
//         DD DSN=ISP.SISPTxxx,DISP=SHR
//*
//ISPILIB  DD DSN=ISP.SISPSAMP,DISP=SHR
//*
//SYSPROC  DD DSN=ISP.SISPCLIB,DISP=SHR
//*
//SYSEXEC  DD DSN=ISP.SISPEXEC,DISP=SHR
//*
//ISPTABL  DD DSN=userid.ISPTABLE,DISP=SHR
//*
//ISPPROF  DD DSN=userid.ISPTABLE,DISP=OLD
//*
```

Figure 1. TSO Logon Procedure

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

If you are using TSO line mode support, your logon procedure must invoke the alternate entry point **PGM=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 105 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 maintenance, releases, or versions of ISPF, you must include the data sets allocated to STEPLIB that contain ISPF load modules in the ISPLLIB allocation. This prevents mixing production code with code to be tested. For more information about library search order, refer to the *z/OS ISPF Services Guide*.

The SISPSASC library is required in STEPLIB only if you are using the ISPF Client/Server feature. SISPSASC is the transient library for SAS/C C++ for MVS. This library 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 "SAS/C Considerations" on page 2 for more information.

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

Chapter 2. Customizing ISPF

The ISPF Configuration Table

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

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

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

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

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

Using the ISPF Configuration Utility Before Installation is Complete

You might want to run the ISPF Configuration Utility for a new release to set configuration options added by the new release before you actually IPL your new system for the first time. If you are installing with CBPDO you can run the utility

after the SMP/E APPLY step of the ISPF FMIDs. If you are installing with ServerPac you can run the utility after the ISPF target libraries are restored to your system.

System Requirements

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

Running the Configuration Utility with a Special Logon Procedure

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

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

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

Figure 2. ISPF Configuration Utility Logon Procedure

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

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

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

Using msys for Setup to Create ISPF Configuration Utility Keyword File

You can use the Managed System Infrastructure (msys) for Setup to create your ISPF Configuration Table keyword file and build the load module or SMP/E Usermod. For information about using msys for Setup refer to the *z/OS Managed System Infrastructure for Setup User's Guide*. This guide also provides information about receiving and installing the ISPF product package using the ISPF product definition ISPXML in data set 'CIM.SCIMXML'.

After you install the ISPF package, refer to the online help (included with the ISPF product package) for details on customizing ISPF and creating the Configuration Table.

Note: After installation, an ISPF-specific policy is available that allows you to define properties of the msys for Setup ISPF data sets. Among other parameters, this policy allows you to define the high-level qualifier to be used for the data sets.

The ISPF Configuration Utility

The ISPF Configuration Utility enables you to do the following:

- modify the configuration settings saved in the keyword files
- generate a configuration table load module or SMP/E USERMOD for use on your ISPF system

- convert existing ISPF configuration table assembler files from earlier releases of ISPF (SAMPLIB member ISRCNFIG) into the keyword file format

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

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

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

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

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

To start the ISPF Configuration Utility run the command **TSO ISPCCONF** while in ISPF. This displays the ISPF Configuration Utility main menu panel, as shown in Figure 3 on page 11.

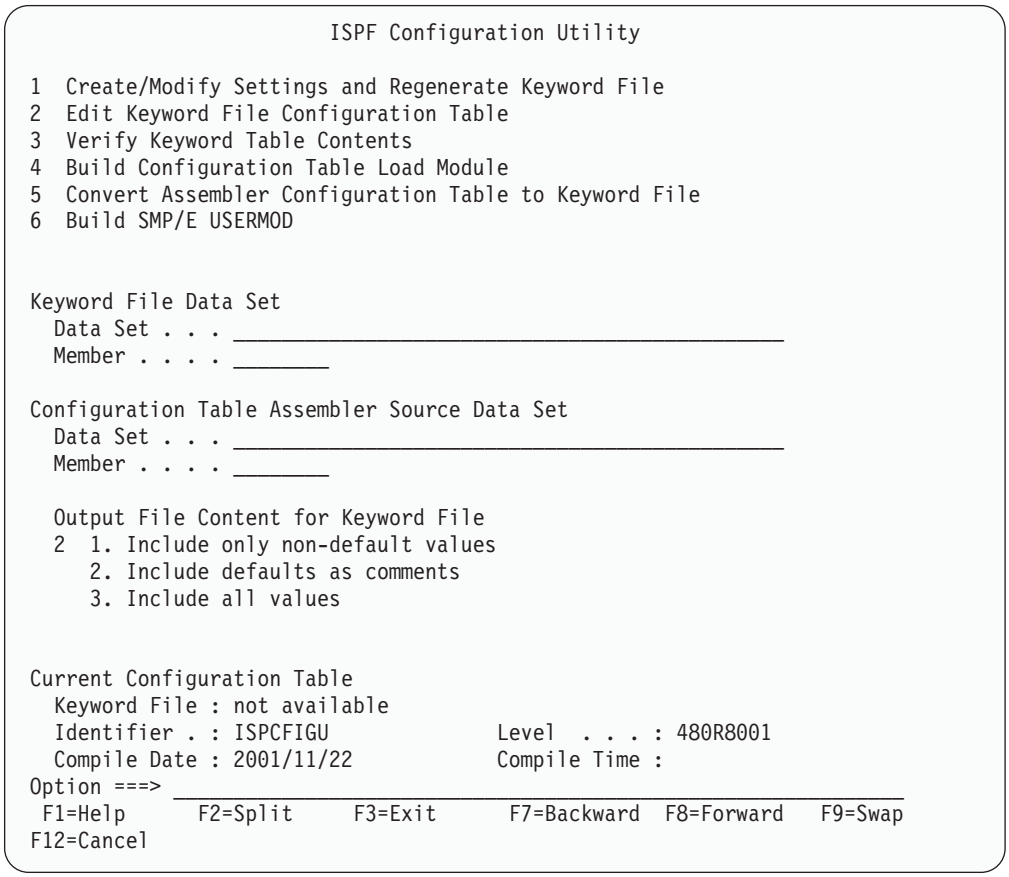


Figure 3. ISPF Configuration Utility Main Menu Panel (ISPPCONF)

The following options are available on the ISPF Configuration Utility main menu:

1. Create/Modify Settings and Regenerate Keyword File (see page 12)
2. Edit Keyword File Configuration Table (see page 28)
3. Verify Keyword Table Contents (see page 30)
4. Build Configuration Table Load Module (see page 31)
5. Convert Assembler Configuration Table to Keyword File (see page 32)
6. Build SMP/E USERMOD (see page 32).

The following 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.

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

```
                    Create/Modify ISPF Configuration          Defaults loaded

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


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


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

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

Figure 4. Create/Modify ISPF Configuration Panel (ISPPMOD)

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

- General ISPF Settings (see "General ISPF Settings Panels" on page 13)
  - 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 22)
  - 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). This panel needs to be scrolled several times to show in its entirety.

```
                  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)

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

Figure 5. Modify PDF Edit Configuration Settings Panel (ISPPMOD1)

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

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

                                                  |
  Edit Recovery Data Set Settings                 |  SCLM Warning Level
    Block Size  . . . . . . 13680                 |  2  1. None
    Primary Blocks  . . . . 40                    |     2. Warn
    Secondary Blocks  . . . 200                   |     3. Error

  ______________________________________________________________________
  New Edit Profile Settings and Overrides

  ------------------------------------------------------------------------
  Please press HELP to see important information about setting edit profile
  defaults.
  ------------------------------------------------------------------------
 F1=Help      F2=Split     F3=Exit      F7=Backward  F8=Forward   F9=Swap
F12=Cancel
```

Figure 6. Modify PDF Edit Configuration Settings Panel (ISPPMOD1)

```
                    Modify PDF Edit Configuration Settings
Command ===> ____________________________________________________________
                                                                More:   - +
  Profile Initial Macro    ________  _  Force initial macro

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

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

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

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

 ____________________________________________________________________________

  Edit Highlighting
  Language                                Enter "/" to select option
  1   1.  Automatic determination         _  HILITE ON
      2.  Assembler
      3.  PL/I                            _  Highlight DO/END logic
      4.  COBOL                           _  Highlight IF logic
      5.  Pascal
      6.  C                               _  Match parentheses
      7.  BookMaster                      /  Highlight FIND strings
      8.  REXX                            /  Highlight cursor position
      9.  ISPF Panel
      10. ISPF Skeleton
      11. JCL
      12. ISPF DTL
      13. Other (CLIST, etc.)
      14. Default (no language)
      15. PL/X
      16. IDL
      17. SuperC Listing
 F1=Help      F2=Split     F3=Exit      F7=Backward  F8=Forward   F9=Swap
F12=Cancel
```

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).

```
                    Modify Edit/View/Browse VSAM Settings      Row 1 to 2 of 6
 Command ===> ____________________________________________ Scroll ===> PAGE

   VSAM Enablement
   Enter "/" to select option
   _  VSAM Enabled for Edit
   _  VSAM Enabled for Browse
   _  VSAM Enabled for View
   _

   VSAM Commands
   VSAM Edit Command  . . DITTO VE /_____________________________________________
   VSAM Browse Command    DITTO VE /_____________________________________________
   VSAM View Command  . . DITTO VE /_____________________________________________

  ____________________________________________________________________________

  Restricted Data Sets

  Command            Data Set Name or Pattern            (E, B, V or A)
     _     ____________________________________________        _
     _     ____________________________________________        _

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

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). This panel can take more than one screen to show in its entirety.

```
                    Modify PDF Configuration Settings
Command ===> ____________________________________________________________
                                                                More:     +
PDF Exits
  Data Set Allocation Program Exit . . . . . ________
  Print Utility Program Exit . . . . . . . . ________
  Print Utility Command Exit . . . . . . . . ________
  Compress Program Exit  . . . . . . . . . . ________
  Compress Command Exit  . . . . . . . . . . ________
  Data Set List Filter Program Exit  . . . . ________
  Member List Filter Program Exit  . . . . . ________
  Data Set Name Change Program Exit  . . . . ________
  Data Set List Line Command Program Exit  . . ________
  Activity Monitoring Program Exit . . . . . ________
  Member List Line Command Program Exit  . . ________
  Member List Line Command Command Exit  . . ________

PDF Data Set Characteristics

  Outlist Utility                   SuperC Block Sizes
  Record Length  . . . 133          List Data Set . . . . . . . . 0
  Block Size . . . . . 13566        Update Data Set . . . . . . . 0
  Primary Blocks . . . 200          Profile Data Set  . . . . . . 0
  Secondary Blocks . . 100          Statements Data Set . . . . . 0
 F1=Help      F2=Split     F3=Exit      F7=Backward  F8=Forward   F9=Swap
F12=Cancel
```

Figure 10. Modify PDF Configuration Settings Panel (ISPPMOD3)

```
                    Modify PDF Configuration Settings
Command ===> ____________________________________________________________
                                                                More:   - +
                                  Listing Primary Quantity  . . 50
                                  Listing Secondary Quantity    100
                                  Update Primary Quantity . . . 15
                                  Update Secondary Quantity . . 30

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

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

  When to use COPY or COPYMOD
  2  1. Use COPY if the target block size is equal to or greater than the
        source block size, COPYMOD otherwise
     2. Use COPY if the target block size is equal to the source block size,
        COPYMOD otherwise
     3. Always use COPYMOD
 F1=Help      F2=Split     F3=Exit      F7=Backward  F8=Forward   F9=Swap
F12=Cancel
```

Figure 11. Modify PDF Configuration Settings Panel (ISPPMOD3)

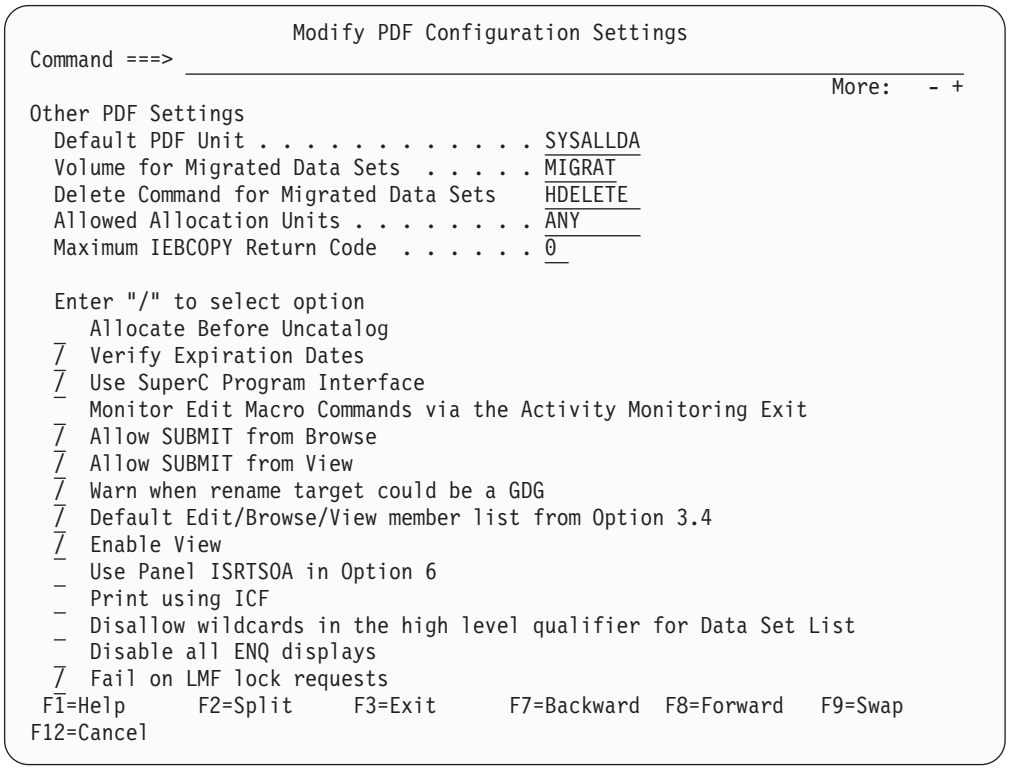


Figure 12. Modify PDF Configuration Settings Panel (ISPPMOD3)

Selecting Option 4, **ISPF Site-wide Defaults** on the Create/Modify ISPF Configuration panel (ISPPMOD) displays the Modify ISPF Sitewide Defaults panel (ISPPMOD4). This panel can take more than one screen to show in its entirety.

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

  Sitewide Defaults Version Level  . . 43000

  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
  / Show Pfkeys                        _ Reset Show Pfkeys
 F1=Help      F2=Split     F3=Exit      F7=Backward  F8=Forward   F9=Swap
F12=Cancel
```

Figure 13. Modify ISPF Sitewide Defaults Panel 1 (ISPPMOD4)

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

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

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

                                       Reset flags
  Minimum Scroll Value  0              /  Reset Scroll Values
  Maximum Scroll Value  9999
 F1=Help      F2=Split     F3=Exit      F7=Backward  F8=Forward   F9=Swap
F12=Cancel
```

Figure 14. Modify ISPF Sitewide Defaults Panel 2 (ISPPMOD4)

```
                    Modify ISPF Sitewide Defaults
Command ===> ____________________________________________________________
                                                                More:   - +

  Member list options
  Enter "/" to select option
  /  Scroll Member List                 _  Reset Scroll Member List
  _  Allow empty member list            _  Reset empty member list Options
  _  Allow empty member list
     (nomatch)
  /  Empty member list for edit only

ISPF Data Set Characteristics

  Log Data Set                          List Data Set
  Record Length  . . . :  125           Records per Block  . .  26
  Block Size . . . . . .  129

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

  ISPCTLx Data Set                      ISPWRKx Data Set
  Record Length  . . . :  80            Record Length  . . . :  256
  Block Size . . . . . .  800           Block Size . . . . . .  2560
  Primary Quantity . . .  10            Primary Quantity . . .  10
  Secondary Quantity . .  100           Secondary Quantity . .  100

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

Figure 15. Modify ISPF Sitewide Defaults Panel 3 (ISPPMOD4)

```
                    Modify ISPF Sitewide Defaults
Command ===> ____________________________________________________________
                                                                More:   - +
  LOG Data Set Disposition              LIST Data Set Disposition
  1  1. No LOG Data Set                 1  1. No LIST Data Set
     2. Print and Delete                   2. Print and Delete
     3. Delete Without Printing            3. Delete Without Printing
     4. Keep                               4. Keep
     5. Keep and Allocate a New LOG        5. Keep and Allocate a New LIST

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

  Additional Temporary Data Set Qualifier  . . _____________________

  ____________________________________________________________________


  Local PRINTDS Options  . . .  NONUM ____________________________________
_____
___
 F1=Help      F2=Split     F3=Exit      F7=Backward  F8=Forward   F9=Swap
F12=Cancel
```

Figure 16. Modify ISPF Sitewide Defaults Panel 4 (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). This panel takes many scrolled screens to show in its entirety.

```
                Modify ISPDFLTS and Other DM Settings
Command ===> ____________________________________________________________
                                                                More:     +
  ISPDFLTS Settings
                                        Enter "/" to select option
  Number of Rows for TBADD   1          _  Enable ISPF Exits
  SAS/C TCP/IP Prefix Value  DEFAULT    _  Use z/OS Unix Sockets
  SAS/C TCP/IP Data Value    DEFAULT

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

  Miscellaneous DM Settings
  Maximum Number of Split Screens . . 8
  Year 2000 Sliding Rule  . . . . . . 65
  Retrieve Command Stack Size . . . . 512
  TPUT Buffer Size  . . . . . . . . . 0
 F1=HELP      F2=SPLIT     F3=END       F4=RETURN    F5=RFIND     F6=RCHANGE
 F7=UP        F8=DOWN      F9=SWAP      F10=LEFT     F11=RIGHT    F12=RETRIEVE
```

Figure 17. Modify ISPDFLTS and Other DM Settings Panel (ISPPMOD5)

```
                Modify ISPDFLTS and Other DM Settings
Command ===> ____________________________________________________________
                                                                More:   - +
  Default Primary Panel . . . . . . . ________
  Default LIBDEF Processing Option    ________  (COND UNCOND STACK or STKADD)

 ____________________________________________________________________________


  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,       | Examples:
  MM, DD and the national language delimiter.    |   English - YY/MM/DD
                                                 |   German  - TT.MM.JJ
  Date Format (ZDATEFD)  . .  ________           |
                                                 |
  ZDATEF must use the characters YY, MM, DD      | Examples:
F1=HELP      F2=SPLIT     F3=END       F4=RETURN    F5=RFIND     F6=RCHANGE
F7=UP        F8=DOWN      F9=SWAP      F10=LEFT     F11=RIGHT    F12=RETRIEVE
```

Figure 18. Modify ISPDFLTS and Other DM Settings Panel (ISPPMOD5)

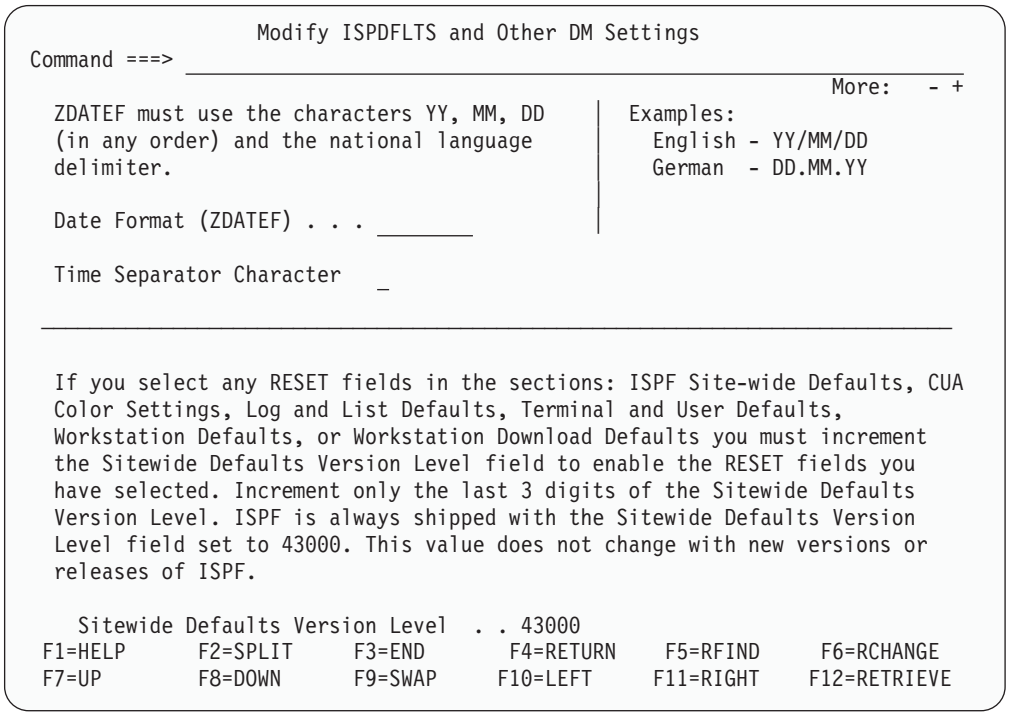


Figure 19. Modify ISPDFLTS and Other DM Settings Panel (ISPPMOD5)

The remaining screens enable you to specify the CUA Color settings for the following 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 20 shows the options for the first two panel elements.

```
                  Modify ISPDFLTS and Other DM Settings          End of data
Command ===>                                                              More:   -
  CUA Color Settings

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

    Action Bar Separator Line
    Color            Intensity       Hilite               Reset Setting
    1  1. Blue       0  0. Low       0  0. None           2  1. Yes
       2. Red           2. High          1. Blink             2. No
       3. Pink                           2. Reverse Video
       4. Green                          4. Underscore
       5. Turquoise
       6. Yellow
       7. White
 F1=HELP      F2=          F3=END       F4=DATASETS  F5=FIND      F6=CHANGE
F9=SWAP     F10=LEFT     F11=RIGHT    F12=SUBMIT
```

Figure 20. Modify ISPDFLTS and Other DM Settings Panel (ISPPMOD5)

System Profile (ISPSPROF) Settings Panels: Selecting Option 6, **Log and List Defaults** on the Create/Modify ISPF Configuration panel (ISPPMOD) displays the Modify Log/List Configuration Settings panel (ISPPMOD6). This panel can take more than one screen to show in its entirety.

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

  Sitewide Defaults Version Level  . . 43000


                            Log and List Job Cards
 Log/List Job Card1
 ________________________________________________________________________
 Log/List Job Card2
 ________________________________________________________________________
 Log/List Job Card3
 ________________________________________________________________________
 Log/List Job Card4
 F1=Help      F2=Split     F3=Exit      F7=Backward  F8=Forward   F9=Swap
F12=Cancel
```

Figure 21. Modify Log/List Configuration Settings Panel (ISPPMOD6)

```
                    Modify Log/List Configuration Settings
Command ===> ________________________________________________________________
                                                                 More:   - +
 Log/List Job Card4
 ________________________________________________________________________
 Unique Job Character  . . _


                                 Log Settings
                                              Enter "/" to select option
 DS Unique Char . . . . 1                     _  Reset Log Batch SYSOUT Class
 Lines Per Page . . . . 60                    _  Reset Log Local Printer ID
 Primary Quantity . . . 10                    _  Reset Log Local SYSOUT Class
 Secondary Quantity . . 10                    _  Message Id
 Batch SYSOUT Class . . _______________       _  Log Display Required
 Local Printer Id or    _________________     _  Log Kept
 writer-name  . . . . .
 Local SYSOUT Class . . _______________


                                List Settings
 DS Unique Char . . . . 1            Logical Record Length   121
 F1=Help      F2=Split     F3=Exit      F7=Backward  F8=Forward   F9=Swap
F12=Cancel
```

Figure 22. Modify Log/List Configuration Settings Panel (ISPPMOD6)

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


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

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

Figure 23. 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). This panel can take more than one screen to show in its entirety.

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

   Sitewide Defaults Version Level  . . 43000

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

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

Figure 24. Modify Terminal and User Settings Panel (ISPPMOD7)

```
                         Modify Terminal and User Settings
Command ===> ____________________________________________________________________
                                                                  More:   - +
                                              3. MAX
                                              4. PART
  Terminal Type
  3   1.  3277      2.  3277A     3.  3278      4.  3278A
      5.  3290A     6.  3278T     7.  3278CF    8.  3277KN
      9.  3278KN    10. 3278AR    11. 3278CY    12. 3278HN
      13. 3278HO    14. 3278IS    15. 3278L2    16. BE163
      17. BE190     18. 3278TH    19. 3278CU    20. DEU78
      21. DEU78A    22. DEU78T    23. DEU90A    24. SW116
      25. SW131     26. SW500     27. 3278GR
 ----------------------------------------------------------------------------

  Environ Defaults                           |  Retrieve Defaults
  Termtrac DD Name . . ISPSNAP               |  Retrieve Minimum Length  . . 1
  Enter "/" to select option                 |  Retrieve Cursor Position
  _  Reset Termtrac DD Name                  |  1  1. End of command
                                             |     2. Start of command
  Environ Enbldump    Environ Termtrac       |
  2  1. ON            2  1. ON               |
     2. OFF              2. OFF              |
                         3. ERROR            |
 ----------------------------------------------------------------------------
 F1=Help      F2=Split     F3=Exit      F7=Backward  F8=Forward   F9=Swap
F12=Cancel
```

Figure 25. Modify Terminal and User Settings Panel (ISPPMOD7)

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

  General Defaults                          Function Keys Defaults
  Character Set Load Module  ISP3278        Number of Keys
  Enter "/" to select option                1  1. 12  2. 24
  _  Reset Character Set Load Module        FKA Setting
                                            2  1. SHORT   2. LONG    3. OFF
                                            Enter "/" to select option
                                            /  Display Prefix

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

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

Figure 26. Modify Terminal and User Settings Panel (ISPPMOD7)

```
                    Modify Terminal and User Settings
 Command ===> ____________________________________________________________
                                                                    More:   -
   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
   (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)
   _  Enable Euro Symbol     (/=YES)
   _  Reset Enable Euro Symbol (/=YES)
  F1=Help      F2=Split     F3=Exit      F7=Backward  F8=Forward   F9=Swap
F12=Cancel
```

Figure 27. 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). This panel can take more than one screen to show in its entirety.

```
                          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. 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 28. 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 29. Workstation Defaults Panel (ISPPMOD8)

```
                              Workstation Defaults
Command ===> _____________________________________________________________________
                                                                        More:   - +
 /  GUI Display ENTER Key
 /  Save GUI Values
 /  GUI Accelerator Support
 /  GUI Download Images
    GUI Make Path for Images
 /  Continue 3270 After Loss of WS Connection

GUI Window Frame                          GUI Switch
1  1. Standard                            1  1. GUI Display
   2. Fixed                                  2. No GUI Display
   3. Dialog
                                          GUI Close Window
GUI Background Color                      1  1. Cancel
1  1. Dialog                                 2. End
   2. Standard                               3. Exit
                                             4. Return
GUI Network
1  1. TCP/IP
   2. APPC
   3. ISPDTPRF file
 F1=Help      F2=Split     F3=Exit      F7=Backward  F8=Forward   F9=Swap
F12=Cancel
```

Figure 30. Workstation Defaults Panel (ISPPMOD8)

Selecting Option 9, **Workstation Download Defaults** on the Create/Modify ISPF Configuration panel (ISPPMOD) displays the **Workstation Download Defaults** panel (ISPPMOD9). This panel can take more than one screen to show in its entirety.

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

  Sitewide Defaults Version Level  . . 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
```

```
                        Workstation Download Defaults
Command ===> ____________________________________________________________
                                                                More:   - +

 WSA Down Level Warning
 1  1. Yes            2. No

 WSA Download Directory

 ____________________________________________________________
 WSA Download FTP Address

 ____________________________________________________________

Workstation Download Option                Workstation Download Platform
1  1. Download Now                         _  1. Windows 2000/NT
   2. Do Not Download                         2. Reserved
   3. Cease Warnings                          3. AIX
                                              4. Solaris
Workstation Download Method                   5. HP UX
   1. Download Using FTP
_  2. Download Using ISPF/Client Server    Enter "/" to select option
   3. Download Manually                    /  Browse FTP Errors
                                           _  Create Directory
 F1=Help      F2=Split     F3=Exit      F7=Backward  F8=Forward   F9=Swap
F12=Cancel
```

Figure 31. 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 32 on page 29.

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 below. Enter END or EXIT
to save your changes, CANCEL to exit without saving your changes.

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

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

Figure 32. 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 33 on page 30. 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.

```
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 33. Keyword File Verification Panel (ISPPVERQ)

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 the following 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 34 is displayed. The listing shows the line in error and the reason that ISPF flagged it as incorrect.

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




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

Figure 34. Verification Failure Listing

Build Configuration Table Load Module

If you choose Option 4, **Build Configuration Table Load Module** on the ISPF Configuration Utility main menu panel, ISPF displays the Build Configuration Table Load Module panel, as shown in Figure 35.

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

While you can create the load modules with any name initially (for example, if you want to create several different configurations for your installation), ISPF only recognizes the default names of ISPCFIGU and ISPCFIGV. If you create load modules with other names and want to use them, you must rename them when they are moved into the ISPF execution data sets.

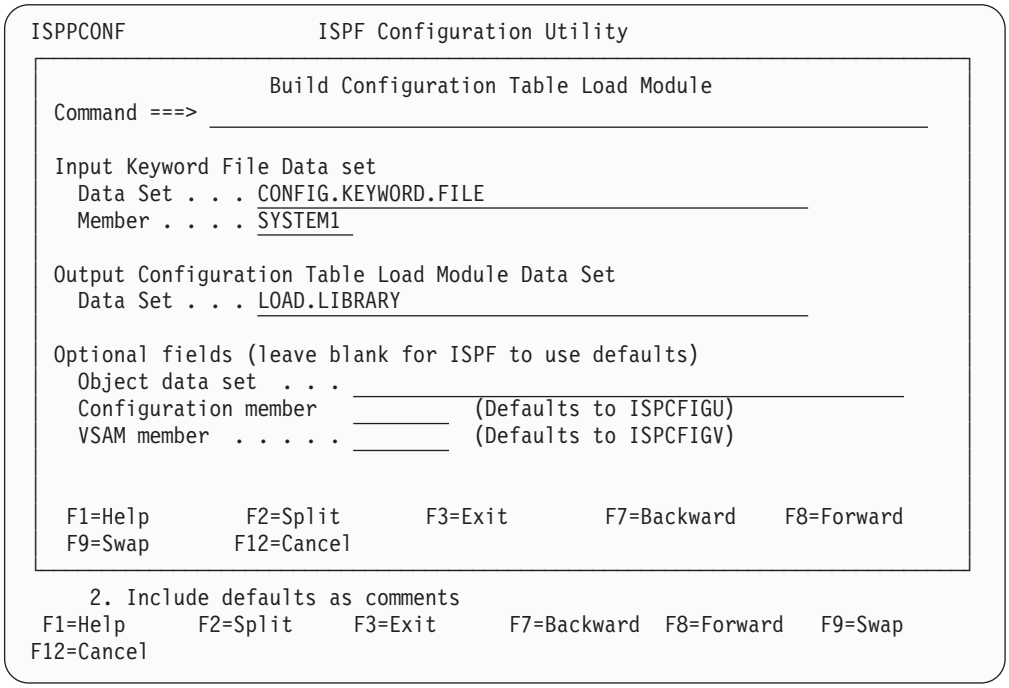


Figure 35. Build Configuration Table Load Module Panel (ISPPBLD)

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

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

Output Configuration Table Load Module Data Set
The name of the pre-existing partitioned data set into which the generated configuration table load module is to be stored.

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

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

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

Convert Assembler Configuration Table to Keyword File

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

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

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

1 Only those values that are different than the default values are included in the keyword file.

2 Those values different than the default values are included in the keyword file, and all default values are included as comment lines.

3 All values are included in the keyword file.

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

Build SMP/E USERMOD

If you choose Option 6, **Build SMP/E USERMOD** on the ISPF Configuration Utility main menu panel, ISPF displays the Build SMP/E USERMOD panel (ISPPSMP), as shown in Figure 36 on page 33.

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.

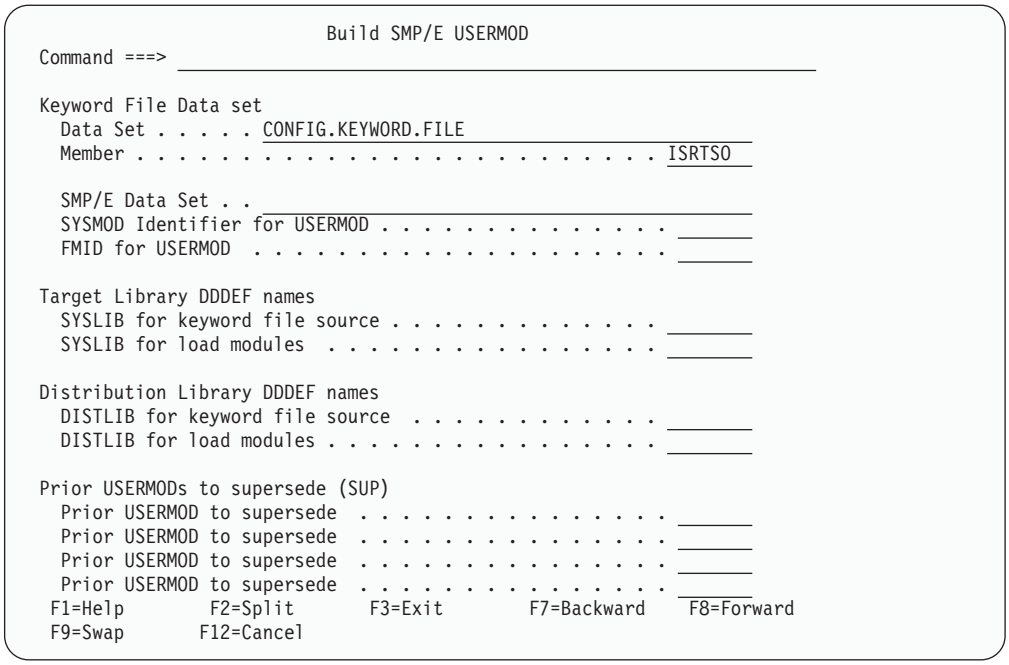


Figure 36. Build SMP/E USERMOD Panel (ISPPSMP)

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

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

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

FMID for USERMOD
The FMID for the FMID operand of the SMP/E USERMOD. The FMID for ISPF z/OS Version 1 Release 7.0 is HIF5F02.

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)
Prior USERMOD to supersede: 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 the following steps.

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

6. Place the load module target library in the normal MVS search order on your system or allocate it to ISPLLIB in your logon procedure.

Notes:

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

ISPF Configuration Table keywords and values

This section explains the keywords in the ISPF Configuration Table and their allowable and default values.

PDF Exits

The following fields specify the name of the program or command exit to be invoked at each exit point. COMMAND exits can be either CLIST or REXX. Exit names can have a maximum length of 8 characters. If both a program exit and a command exit are specified for the same exit point, the program exit is used.

DATA_SET_ALLOCATION_PROGRAM_EXIT
The program to be invoked as the data set allocation exit. This should be the name of a load module in your standard MVS search sequence.

PRINT_UTILITY_PROGRAM_EXIT
The program to be invoked as the print utility exit. This should be the name of a load module in your standard MVS search sequence.

PRINT_UTILITY_ COMMAND_ EXIT
The command to be invoked as the print utility exit. This should be the name of a member in your SYSPROC or SYSEXEC allocation.

COMPRESS_UTILITY_PROGRAM_EXIT
The program to be invoked as the compress utility exit. This should be the name of a load module in your standard MVS search sequence.

COMPRESS_UTILITY_CLIST_EXIT
The command to be invoked as the compress utility exit. This should be the name of a member in your SYSPROC or SYSEXEC allocation.

DATA_SET_LIST_FILTER_PROGRAM_EXIT
The program to be invoked as the data set list filter exit. This should be the name of a load module in your standard MVS search sequence.

MEMBER_LIST_FILTER_PROGRAM_EXIT
The program to be invoked as the member list filter exit. This should be the name of a load module in your standard MVS search sequence.

DATA_SET_NAME_CHANGE_PROGRAM_EXIT
The program to be invoked as the data set name change exit. This should be the name of a load module in your standard MVS search sequence.

DATA_SET_LIST_LINE_COMMAND_PROGRAM_EXIT
The program to be invoked as the data set list line command exit. This should be the name of a load module in your standard MVS search sequence.

ACTIVITY_MONITORING_PROGRAM_EXIT
The program to be invoked as the activity monitoring exit. This should be the name of a load module in your standard MVS search sequence.

MEMBER_LIST_LINE_COMMAND_PROGRAM_EXIT
The program to be invoked as the member list line command exit. This should be the name of a load module in your standard MVS search sequence.

MEMBER_LIST_LINE_COMMAND_COMMAND_EXIT
The command to be invoked as the member list line command exit. This should be the name of a member in your SYSPROC or SYSEXEC allocation.

Data Set Allocation Settings

PDF_DEFAULT_UNIT
The unit name used by PDF when allocating work data sets. This value is used any time PDF needs to allocate a new data set on behalf of the user with the exception of option 3.2, edit recovery and ISPF work, control, and list data sets (ISPWRKx, ISPCTLx, and ISPLSTx). These new data sets may be temporary or permanent depending on the option of PDF being used.

For improved performance it is recommended that the VIO=YES option be added to the UNITNAME macro for the unit you specify in this field, but a VIO-only unit name is not recommended. Several of the ISPF options (including the Move/Copy Utility and Outlist Utility) will not function with VIO data sets.

The default is SYSALLDA.

ALLOWED_ALLOCATION_UNITS
This field controls which unit names are eligible to a user when the user is creating a data set through option 3.2. Valid values are:

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 39).

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.

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 one of the following is done:

- 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 below). 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 **DITTO VE /**

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 below). 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 **DITTO VB /**

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 below). 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 **DITTO VB /**

Edit Site-Wide Profile Customizations

The following 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* manual. 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.

RECOVERY
Determines whether the editor maintains its recovery file to keep track of the edit session, making it possible for users to recover from system failures. Valid values are ON and OFF.

The default is OFF.

FORCE_RECOVERY
Force the specified value.

The default is NO.

RECOVERY_WARNING_MESSAGE
Determines whether the editor should warn users when they enter an edit session with RECOVERY set OFF. Valid values are WARN and NOWARN.

The default is WARN.

FORCE_RECOVERY_WARNING_MESSAGE
Force the specified value.

The default is NO.

SETUNDO
Determines whether the UNDO command will be available in Edit. Valid values are ON and OFF.

The default is ON.

FORCE_SETUNDO
Force the specified value.

The default is NO.

PACK Determines whether the editor will save data in packed or unpacked format. Valid values are ON and OFF.

The default is OFF.

FORCE_PACK
Force the specified value.

The default is NO.

IMACRO
Specifies the sitewide initial macro to be run for all users.

The default is NONE (no macro).

FORCE_IMACRO
Force the specified value.

The default is NO.

CAPS Determines whether the editor automatically rolls text to uppercase. Valid values are ON and OFF.

The default is OFF.

NOTE Determines whether the editor displays ==NOTE== lines when the Edit MODEL command is used. Valid values are ON and OFF.

The default is ON.

HEX Determines whether the edit data is displayed in hex mode.
- ON

- OFF
- VERT
- DATA

The default is OFF.

NULLS
Determines whether trailing spaces on edit data are written to the screen as nulls or blanks.
- STD
- ALL
- OFF

The default is STD.

DISPLAY_SEQUENCE_NUMBERS
Determines whether the editor will maintain sequence numbers. Valid values are ON and OFF.

The default is ON.

COBOL_NUMBERS
Determines whether the sequence numbers should be maintained in COBOL format (in columns 1-6). Valid values are ON and OFF.

The default is OFF.

STANDARD_NUMBERS
Determines whether the sequence numbers should be maintained in standard format (in columns 1-8 for variable data, the last 8 columns for fixed data). Valid values are ON and OFF.

The default is ON.

AUTONUM
Determines whether the editor automatically renumbers edit data. Valid values are ON and OFF.

The default is OFF.

AUTOLIST
Determines whether the editor writes the edit data to the ISPF LIST data set when a user ends an edit session in which data has been changed or saved. Valid values are ON and OFF.

The default is OFF.

PROFILE
Determines whether an edit profile can be deleted if it is the least recently used profile. Specify LOCK to prevent profiles from being deleted. Valid values are LOCK and UNLOCK.

The default is UNLOCK.

AUTOSAVE
Determines whether the editor automatically saves changes when the END command is entered. Valid values are ON and OFF.

The default is ON.

AUTOSAVE_PROMPT
Determines whether the editor prompts the user to have the data changes saved 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.

ISPF Site-Wide Profile Customizations

ISPF site-wide defaults are not enabled until the VERSION_LEVEL_OF_SITEWIDE_DEFAULTS field is modified.

The RESET fields below mean that ISPF resets the site-wide defaults for each user. This is done once each time the VERSION_LEVEL_OF_SITEWIDE_DEFAULTS field is incremented. Users can change the values of their fields after the incrementation has caused the reset.

VERSION_LEVEL_OF_SITEWIDE_DEFAULTS
This field indicates the modification level of the current site-wide defaults values. This field is set to 43000 initially by ISPF and will not be changed with new versions of ISPF. Each time you modify any of the sitewide defaults, increment this value by 1. For example, the first time you change any defaults you should set this field to 43001. This value is then saved in the ISPF system profile table.

When ISPF is initialized it checks the value saved in the system profile against the value in this field. If the value of this field is greater than that in the system profile, ISPF will use the values specified below. You must increment this value every time you modify a default or it will not be picked up by ISPF.

TAB_TO_POINT_AND_SHOOT
Enable tabbing to point and shoot fields. Valid values are YES and NO.

The default is NO.

RESET_TAB_TO_POINT_AND_SHOOT
Reset the value specified above.

The default is NO.

TAB_TO_ACTION_BARS
Enable tabbing to action bars. Valid values are YES and NO.

The default is YES.

RESET_TAB_TO_ACTION_BARS
Reset the value specified above.

The default is NO.

USE_SESSION_MANAGER
Value is used to set system variable ZSESS and is used to initialize system variable ZSM. Valid values are YES and NO.

The default is NO.

RESET_USE_SESSION_MANAGER
Reset the value of system variable ZSM to the value of the USE_SESSION_MANAGER keyword.

JUMP_FROM_LEADER_DOTS
Enable the ISPF jump command (for example, =2) from fields with leader dots. Valid values are YES and NO.

The default is YES.

RESET_JUMP_FROM_LEADER_DOTS
Reset the value specified above.

The default is NO.

SHOW_SPLIT_LINE
Should ISPF show the split line when a user is running in split screen mode. Valid values are YES and NO.

The default is YES.

RESET_SHOW_SPLIT_LINE
Reset the value specified above.

The default is NO.

LONG_MESSAGES_IN_POPUP
Should ISPF long messages always be shown in pop-up windows, or only when they are longer than 78 characters. Valid values are YES and NO.

The default is YES.

RESET_LONG_MESSAGES_IN_POPUP
Reset the value specified above.

The default is NO.

EDIT_PRINTDS_COMMAND
Should the user be allowed to modify the PRINTDS command generated by ISPF before its submission. Valid values are YES and NO.

The default is NO.

RESET_EDIT_PRINTDS_COMMAND
Reset the value specified above.

The default is NO.

RESTORE_TEST_TRACE_OPTIONS
Should the original TEST and TRACE options specified on ISPF invocation be restored when a user exits from Dialog Test. Dialog test will set TEST mode on. Valid values are YES and NO.

The default is YES.

RESET_RESTORE_TEST_TRACE_OPTIONS
Reset the value specified above.

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 above.

The default is NO.

LOG_DATA_SET_DISPOSITION
The default disposition for the ISPF Log data set. Valid defaults are:

1 No log data set
2 Print and delete
3 Delete without printing
4 Keep
5 Keep and allocate a new log

The default is 1, No log data set.

Note: When manually editing the configuration table the valid options are:
NONE
No log data set
J Print and delete
D Delete without printing
K Keep
R Keep and allocate a new log data set

The default is NONE, No log data set.

RESET_LOG_DATA_SET_DISPOSITION
Reset the value specified above. The default is NO.

LIST_DATA_SET_DISPOSITION
The default disposition for the ISPF List data set. Valid defaults are:
1 No list data set
2 Print and delete
3 Delete without printing
4 Keep
5 Keep and allocate a new list

The default is 1, No list data set.

Note: When manually editing the configuration table the valid options are:
NONE
No list data set
J Print and delete
D Delete without printing
K Keep
R Keep and allocate a new list data set

The default is NONE, No list data set.

RESET_LIST_DATA_SET_DISPOSITION
Reset the value specified above.

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 above. 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 above.

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 above.

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 above.

The default is NO.

SCROLL_DEFAULT
Select the default scroll value.

The default is PAGE.

SCROLL_MIN
Select the minimum scroll value allowed.

The default is 0.

SCROLL_MAX
Select the maximum scroll value allowed.

The default is 9999.

RESET_SCROLL_VALUE
Force an update of ISPSPROF from the configuration table values at ISPF initialization.

The default is NO.

DISPLAY_EMPTY_MEMBER_LIST
Controls whether an empty member list is displayed.

The default is NO.

DISPLAY_EMPTY_MEMBER_LIST_PATTERN
If the DISPLAY_EMPTY_MEMBER_LIST option is set, this field controls whether an empty list that results from a nonmatching pattern will be displayed.

The default is NO.

DISPLAY_EMPTY_MEMBER_LIST_FUNCTION
Whether empty member list options apply to non-edit functions such as View and Browse.

The default is YES.

RESET_EMPTY_MEMBER_LIST_OPTIONS
Reset the values specified in the DISPLAY_EMPTY_MEMBER_LIST fields.

The default is NO.

STATUS_AREA_DEFAULT
Select the default status area value.

The default is SESSION.

LIST_DATA_SET_RECORDS_PER_BLOCK
The number of records per block for the ISPF List data set. This value must be in the range 0 to 32760. A value of 0 will result in an ISPF list data set being allocated using a system-determined block size.

The default is 26.

LOG_DATA_SET_BLOCK_SIZE
The block size of the ISPF Log data set. This value must be in the range 0 to 32760. A value of 0 will result in an ISPF log data set being allocated using a system-determined block size.

The default is 129.

LOG_DATA_SET_RECORD_LENGTH
The record length of the ISPF Log data set.

The default is 125.

BLOCK_SIZE_FOR_TEMPORARY_CNTL_DATA_SETS
The block size for ISPF temporary control (CNTL) data sets. This value must be in the range 0 to 32760. A value of 0 will result in the data sets being allocated using a system-determined block size.

The default is 800.

RECORD_LENGTH_FOR_TEMPORARY_CNTL_DATA_SETS
The record length for ISPF temporary control (CNTL) data sets.

The default is 80.

BLOCK_SIZE_FOR_TEMPORARY_LIST_DATA_SETS
The block size for ISPF temporary list data sets. This value must be in the range 0 to 32760. A value of 0 will result in the data sets being allocated using a system-determined block size.

The default is 3146.

RECORD_LENGTH_FOR_TEMPORARY_LIST_DATA_SETS
The record length for ISPF temporary list data sets.

The default is 121.

BLOCK_SIZE_FOR_TEMPORARY_WORK_DATA_SETS
The block size for ISPF temporary work data sets. This value must be in the range 0 to 32760. A value of 0 will result in the data sets being allocated using a system-determined block size.

The default is 2560.

RECORD_LENGTH_FOR_TEMPORARY_WORK_DATA_SETS
The record length for ISPF temporary work data sets.

The default is 256.

ISPCTL0_BLOCK_SIZE
The block size for ISPCTL0 temporary control (CNTL) data set. This value must in the range 0 to 32760.

The default is 800.

ISPCTL0_RECORD_LENGTH
The record length for the ISPCTL0 temporary control (CNTL) data set.

The default is 80.

ISPCTL0_PRIMARY_QUANTITY
Primary number of blocks to be allocated for the ISPCTL0 temporary control (CNTL) data set.

The default is 10.

ISPCTL0_SECONDARY_QUANTITY
Secondary number of blocks to be allocated for the ISPCTL0 temporary control (CNTL) data set.

The default is 100.

ISPCTL_PRIMARY_QUANTITY
Primary number of blocks to be allocated for file tailoring CNTL data sets.

The default is 10.

ISPCTL_SECONDARY_QUANTITY
Secondary number of blocks to be allocated for file tailoring CNTL data sets.

The default is 100.

ISPLST_PRIMARY_QUANTITY
Primary number of blocks to be allocated for ISPF temporary list data sets.

The default is 10.

ISPLST_SECONDARY_QUANTITY
Secondary number of blocks to be allocated for ISPF temporary list data sets.

The default is 100.

ISPWRK_PRIMARY_QUANTITY
Primary number of blocks to be allocated for the file tailoring WORK data sets.

The default is 10.

ISPWRK_SECONDARY_QUANTITY
Secondary number of blocks to be allocated for the File Tailoring WORK data sets.

The default is 100.

USE_PDFCUNIT_FOR_TEMP_ISPF_DATA_SETS
Indicates whether to use the PDFCUNIT value for the units field when allocating ISPF temporary data sets. Valid values are YES and NO.

The default is NO.

ISPF_TEMPORARY_DATA_SET_QUALIFIER
An additional qualifier that will be appended to the ISPF log, list, and temporary control data set names. The qualifier will come after the ISPF assigned prefix, but before the suffix area. If Exit 16 is active, this qualifier will be part of the 26-byte prefix area passed to the exit.

The qualifier can be either of the following:

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 24 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).
```

Refer to the *z/OS MVS Initialization and Tuning Reference* for details on valid system symbols.

The default is NONE (no qualifier).

PRINTDS_DEST_OR_WRITER_OPTION
PRINTDS option, valid values are:

DEST Indicates a local printer ID is being used.

WRITER Indicates an external writer name is being used.

The default is DEST.

LOCAL_PRINTDS_OPTIONS
Parameters appended to the PRINTDS command when a local print is done. No verification of these fields is done. The maximum length of this value is 128 characters. To disable printing through PRINTDS, specify a value of DISABLE.

The default is NONUM.

USE_ALTERNATE_DIALOG_TEST_PANEL
Select alternate dialog test panels ISPYFP, ISPYFPA, or ISPYFPB. Panel ISPYFPA is formatted with the most frequently used fields at the top of the panel. Panel ISPYFPB is similar to panel ISPYFPA, but it has a selection field that allows the user to select a function: panel, command, program, or request. Unlike the panels ISPYFP or ISPYFPA, on panel ISPYFPB the panel, command, program, or request fields can all contain values. Valid options are:

1 ISPYFP panel
2 ISPYFPA panel
3 ISPYFPB panel

The default is 1.

Default CUA Color Settings

The following 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 one of the following:

Position 1 — COLOR

1 Blue
2 Red
3 Pink
4 Green

5 Turquoise
6 Yellow
7 White

Position 2 — INTENSITY
0 Low
2 High

Position 3 — HILITE
0 None
1 Blink
2 Reverse Video
4 Underscore

ISPF site-wide defaults are not enabled until the VERSION_LEVEL_OF_SITEWIDE_DEFAULTS field is modified. See "ISPF Site-Wide Profile Customizations" on page 47 for more details. If you select any RESET fields below, you *must* increment the value in the VERSION_LEVEL_OF_SITEWIDE_DEFAULTS.

The RESET fields below mean that ISPF resets the site-wide defaults for each user. This is done once each time the VERSION_LEVEL_OF_SITEWIDE_DEFAULTS field is incremented. Users can change the values of their fields after the incrementation has caused the reset.

ACTION_BAR_SELECTED_CHOICE
Default is 600 (yellow, low intensity, no highlighting).

RESET_ACTION_BAR_SELECTED_CHOICE
Reset the value specified above.

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 above.

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 above.

The default is NO.

ACTION_MESSAGE_TEXT
Default is 220 (Red, high intensity, no highlighting).

RESET_ACTION_MESSAGE_TEXT
Reset the value specified above.

The default is NO.

CAUTION_TEXT
Default is 620 (Yellow, high intensity, no highlighting).

RESET_CAUTION_TEXT
Reset the value specified above.

The default is NO.

CHOICE_ENTRY_FIELD
Default is 504 (Turquoise, low intensity, underscored)

RESET_CHOICE_ENTRY_FIELD
Reset the value specified above.

The default is NO.

COLUMN_HEADING
Default is 120 (Blue, high intensity, no highlighting).

RESET_COLUMN_HEADING
Reset the value specified above.

The default is NO.

DESCRIPTIVE_TEXT
Default is 400 (Green, low intensity, no highlighting).

RESET_DESCRIPTIVE_TEXT
Reset the value specified above.

The default is NO.

EMPHASIZED_TEXT
Default is 520 (Turquoise, high intensity, no highlighting).

RESET_EMPHASIZED_TEXT
Reset the value specified above.

The default is NO.

ERROR_EMPHASIS
Default is 622 (Yellow, high intensity, reverse video)

RESET_ERROR_EMPHASIS
Reset the value specified above.

The default is NO.

FIELD_PROMPT
Default is 400 (Green, low intensity, no highlighting).

RESET_FIELD_PROMPT
Reset the value specified above.

The default is NO.

FUNCTION_KEYS
Default is 100 (Blue, low intensity, no highlighting).

RESET_FUNCTION_KEYS
Reset the value specified above.

The default is NO.

INFORMATIONAL_MESSAGE_TEXT
Default is 720 (White, high intensity, no highlighting).

RESET_INFORMATIONAL_MESSAGE_TEXT
Reset the value specified above.

The default is NO.

LIST_ENTRY_FIELD
Default is 504 (Turquoise, low intensity, underscored)

RESET_LIST_ENTRY_FIELD
Reset the value specified above.

The default is NO.

LIST_ITEM_DESCRIPTION
Default is 400 (Green, low intensity, no highlighting).

RESET_LIST_ITEM_DESCRIPTION
Reset the value specified above.

The default is NO.

LIST_ITEM
Default is 700 (White, low intensity, no highlighting).

RESET_LIST_ITEM
Reset the value specified above.

The default is NO.

NORMAL_ENTRY_FIELD
Default is 504 (Turquoise, low intensity, underscored)

RESET_NORMAL_ENTRY_FIELD
Reset the value specified above.

The default is NO.

NORMAL_TEXT
Default is 400 (Green, low intensity, no highlighting).

RESET_NORMAL_TEXT
Reset the value specified above.

The default is NO.

PANEL_ID
Default is 100 (Blue, low intensity, no highlighting).

RESET_PANEL_ID
Reset the value specified above.

The default is NO.

PANEL_INFORMATION
Default is 400 (Green, low intensity, no highlighting).

RESET_PANEL_INFORMATION
Reset the value specified above.

The default is NO.

PANEL_TITLE
Default is 100 (Blue, low intensity, no highlighting).

RESET_PANEL_TITLE
Reset the value specified above.

The default is NO.

POINT_AND_SHOOT
Default is 520 (Turquoise, high intensity, no highlighting).

RESET_POINT_AND_SHOOT
Reset the value specified above.

The default is NO.

PULLDOWN_AVAILABLE_CHOICE
Default is 700 (White, low intensity, no highlighting).

RESET_PULLDOWN_AVAILABLE_CHOICE
Reset the value specified above.

The default is NO.

PULLDOWN_UNAVAILABLE_CHOICE
Default is 100 (Blue, low intensity, no highlighting).

RESET_PULLDOWN_UNAVAILABLE_CHOICE
Reset the value specified above.

The default is NO.

REFERENCE_PHRASE
Default is 720 (White, high intensity, no highlighting).

RESET_REFERENCE_PHRASE
Reset the value specified above.

The default is NO.

SCROLL_INFORMATION
Default is 720 (White, high intensity, no highlighting).

RESET_SCROLL_INFORMATION
Reset the value specified above.

The default is NO.

SELECTION_AVAILABLE_CHOICE
Default is 700 (White, low intensity, no highlighting).

RESET_SELECTION_AVAILABLE_CHOICE
Reset the value specified above.

The default is NO.

SELECTION_UNAVAILABLE_CHOICE
Default is 100 (Blue, low intensity, no highlighting).

RESET_SELECTION_UNAVAILABLE_CHOICE
Reset the value specified above.

The default is NO.

VARIABLE_OUTPUT_INFORMATION
Default is 500 (Turquoise, low intensity, no highlighting).

RESET_VARIABLE_OUTPUT_INFORMATION
Reset the value specified above.

The default is NO.

WARNING_MESSAGE_TEST
Default is 620 (Yellow, high intensity, no highlighting).

RESET_WARNING_MESSAGE_TEST
Reset the value specified above.

The default is NO.

WARNING_MESSAGE
Default is 220 (Red, high intensity, no highlighting).

RESET_WARNING_MESSAGE
Reset the value specified above.

The default is NO.

WORKAREA_SEPARATOR_LINE
Default is 100 (Blue, low intensity, no highlighting).

RESET_WORKAREA_SEPARATOR_LINE
Reset the value specified above.

The default is NO.

Miscellaneous Settings

MONITOR_EDIT_MACRO_COMMANDS
Should the ISPF Activity Monitoring Exit be invoked for ISREDIT commands invoked from an ISPF Edit Macro. Valid values are YES and NO.

The default is NO.

ALLOW_SUBMIT_FROM_BROWSE
Should users be allowed to issue the SUBMIT command from with a Browse session. Valid values are YES and NO.

The default is YES.

ALLOW_SUBMIT_FROM_VIEW
Should users be allowed to issue the SUBMIT command from with a View session. Valid values are YES and NO.

The default is YES.

WARN_ON_RENAME_TO_GDG_NAME
Should a warning panel be displayed when a user attempts to rename a data set to a new name that matches the naming convention of a GDG generation. The renamed data set may become a valid generation if it matches the naming convention of an existing GDG data set. If that new generation causes the LIMIT parameter value specified when the GDG was defined to be exceeded, the system will take action based on the SCRATCH/NOSCRATCH and EMPTY/NOEMPTY parameters that were specified when the GDG was defined. This action may result in one or all of the existing generations being deleted or uncataloged. Valid values are YES and NO.

The default is YES.

DEFAULT_EDIT/BROWSE/VIEW_MEMBER_LIST
Should option 3.4 (Data Set List Utility) use the enhanced member list for the Edit, Browse and View actions. Performance is improved if the traditional member list ID used, but capability is improved using the enhanced member list. Valid values are YES and NO.

The default is YES.

IS_VIEW_SUPPORTED
Should users be allowed to use the View function from option 1 or only the Browse function. Because it is based on ISPF Edit, View can impact system resource utilization. Valid values are YES and NO.

The default is YES.

USE_ALTERNATE_PANEL_ISRTSOA
Should alternate ISPF Command Shell panel ISRTSOA be used in place of

panel ISRTSO. ISRTSOA contains both an ISPF command line and a TSO command line, ISRTSO contains only one input field for both ISPF and TSO commands. Valid values are YES and NO.

The default is NO.

PRINT_USING_ICF
Indicates whether foreground print requests should be processed using an ICF printer definition. Valid values are YES and NO.

The default is NO.

DISALLOW_WILDCARDS_IN_HLQ
Indicates whether wildcards (* or %) are allowed in the high-level qualifier for data set list. Valid values are YES and NO.

The default is NO.

MAXIMUM_NUMBER_OF_SPLIT_SCREENS
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[,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[,sit2[,sit3]])
to specify 1 to 3 application IDs.

The default value is NONE (No site-wide command tables).

In addition, a special format can be used to obtain the application ID from the current system name (ISPF dialog variable ZSYSID). The special format is:

```
*, *m, or *m:n.
```

As the system name can be up to 8 characters, *m* and *n* are the start and end positions within the system name used to determine the application ID for the site command tables. The values for *m* and *n* must be in the range 1 to 8, where *m* is less than or equal to *n* and the difference in their values is no more than 3.

The default value for *m* is 1. The default value for *n* is *m*+3, to a maximum value of 8.

Compatibility Issue:
For compatibility of the ISPF configuration options with previous releases, do not specify any of the special formats (*, *m, or *m:n) as either the first user or site application ID.

SITE_COMMAND_TABLE_SEARCH_ORDER
Determines whether the site-wide command tables are searched before or after the default ISP command table. Valid values are AFTER and BEFORE.

The default is BEFORE.

Depending on this setting, the search order will be:

Search Order = BEFORE	Search Order = AFTER
1. Application	1. Application
2. USER (1 to 3)	2. USER (1 to 3)
3. SITE (1 to 3)	3. System
4. System	4. SITE (1 to 3)

YEAR_2000_SLIDING_RULE
The cutoff value used by ISPF to determine whether a 2-character year date specified to ISPF should be considered a 19xx or 20xx date. Values less than or equal to this date will be considered 20xx, values greater will be considered 19xx. Value can be an absolute number or a number preceded by a minus sign to indicate the cutoff should be the specified number of years before the current year. For example, specifying 72 indicates that any 2-character year less than or equal to 72 should be considered 20xx, anything greater should be 19xx. Specifying -40 (assuming the current year is 1999) will yield a cutoff value of 59.

The default is 65.

SHOW_ENQ_DISPLAYS
Used to indicate that users should not be able to see who has existing data set ENQs when they press the help key or when they use the ISRDDN utility.

The default is blank.

DEFAULT_SESSION_LANGUAGE

Selects the language to use as the default language for ISPF. The value chosen will be the language used by ISPF if no language is specified on invocation. Valid values are:

1. English
2. Uppercase English
3. German
4. Swiss German
5. Japanese

The default is 1 (English).

Values Formerly in ISPDFLTS

These fields represent values in the ISPF Configuration table that in previous releases of ISPF were set through ISPDFLTS.

NUMBER_OF_ROWS_FOR_TBADD

The number of rows to be used in calculating the amount of storage required when a TBADD service is invoked. The value can be an integer from 1 to 1000.

The default is 1.

RETRIEVE_COMMAND_STACK_SIZE

The size, in bytes, of the command stack that RETRIEVE command uses. ISPF uses the command stack to hold commands (stripped of leading and trailing blanks) and some ISPF internal information. A stack of 512 bytes holds approximately 20 commands with a length of 10 characters each. The value can be an integer from 312 to 4096.

The default is 512.

ENABLE_ISPF_EXITS

Indicates whether exit routines are available:

YES Indicates that installation-written routines are provided or planned, and that the ISPEXITS load module is to be loaded at ISPF initialization. If you plan to use exit routines but those routines have not been written, you can code YES for this field, causing the IBM-provided defaults module, ISPEXITS, to be loaded. Later, you can replace ISPEXITS without having to repeat this part of the installation process.

NO Indicates that exit routines are not provided or planned, and that ISPEXITS is not to be loaded, reducing startup time.

The default is NO.

SAS/C_TCPIP_DATA_VALUE

This parameter can be used to set the value of the SAS/C environment variable =TCPIP_DATA. A value of DEFAULT indicates that ISPF should set no value.

The default is DEFAULT.

SAS/C_TCPIP_PREFIX_VALUE

This parameter is used to set the value of the SAS/C environment variable =TCPIP_PREFIX. A value of DEFAULT indicates that ISPF should set no value. If a nonblank value is specified, SAS/C concatenates that string to the characters .TCPIP.DATA. For example, specifying a value of TCPIP causes SAS/C to search for the data set TCPIP.TCPIP.DATA.

The default is DEFAULT.

USE_ZOS_UNIX_SOCKETS
This parameter is used to determine whether the z/OS UNIX System Services socket implementation is to be used instead of the common TCP/IP socket implementation. Specify a value of YES for z/OS UNIX socket implementation, NO to use another common socket implementation. Valid values are YES and NO.

The default is NO.

ISPF GUI users running TCP/IP communications might want to have non-full-screen TSO data appear in an ISPF/TSO GUI window instead of being directed to their 3270 session. If this is the case, a value of YES must be specified in this field. For more information, see "Set Up ISPF GUI for TSO Line Mode Support", and the section on TSO line mode support in the *z/OS ISPF User's Guide Vol I*.

VSAM Data Set Restrictions

VSAM_RESTRICTED_EDIT_DATASET
The names of the data sets that are restricted from use in Edit. Wildcards may be used in the data set name (as in option 3.4) to specify sets of restricted data sets.

The default is NONE (all data sets allowed).

VSAM_RESTRICTED_BROWSE_DATASET
The names of the data sets that are restricted from use in Browse. Wildcards may be used in the data set name (as in option 3.4) to specify sets of restricted data sets.

The default is NONE (all data sets allowed).

VSAM_RESTRICTED_VIEW_DATASET
The names of the data sets that are restricted from use in View. Wildcards may be used in the data set name (as in option 3.4) to specify sets of restricted data sets.

The default is NONE (all data sets allowed).

ISPSPROF General Values

ISPF site-wide defaults are not enabled until the VERSION_LEVEL_OF_SITEWIDE_DEFAULTS field is modified. See "ISPF Site-Wide Profile Customizations" on page 47 for more details. If you select any RESET fields below, you *must* increment the value in the VERSION_LEVEL_OF_SITEWIDE_DEFAULTS.

The RESET fields below mean that ISPF resets the site-wide defaults for each user. This is done once each time the VERSION_LEVEL_OF_SITEWIDE_DEFAULTS field is incremented. Users can change the values of their fields after the incrementation has caused the reset.

LOG/LIST_JOB_CARD1
First job card for log/list.

The default is NONE (blank).

LOG/LIST_JOB_CARD2
Second job card for log/list.

The default is NONE (blank).

LOG/LIST_JOB_CARD3
Third job card for log/list.

The default is NONE (blank).

LOG/LIST_JOB_CARD4
Fourth job card for log/list.

The default is NONE (blank).

LOG_DATA_SET_UNIQUE_CHARACTER
This unique character is used as the default only when a first-time user profile is created.

The default is 1.

LOG_SYSOUT_CLASS
Log data set sysout class.

The default is NONE (blank).

RESET_LOG_SYSOUT_CLASS
Reset the Log data set sysout class.

The default is NO.

LOCAL_LOG_SYSOUT_CLASS
Local log data set sysout class.

The default is NONE (blank).

RESET_LOCAL_LOG_SYSOUT_CLASS
Reset the Local log data set sysout class.

The default is NO.

LOG_DISPLAY_REQUIRED
Log panel display required on termination. Valid values are YES or NO.

The default is NO.

LOG_KEPT
The log data set is to be kept. Valid values are YES or NO.

The default is NO.

LOG_LINES_PER_PAGE
The number of log lines per page.

The default is 60.

LOG_MESSAGE_ID
The log message ID. Valid values are YES or NO.

The default is NO.

LOG_LOCAL_PRINTER_ID
The local printer ID (CHAR(17)) for the log.

The default is NONE (blank).

RESET_LOG_LOCAL_PRINTER_ID
Reset the log local printer ID for the log.

The default is NO.

LOG_PAGES_PRIMARY_QUANTITY
The primary quantity of log pages.

The default is 10.

LOG_PAGES_SECONDARY_QUANTITY
The secondary quantity of log pages.

The default is 10.

LIST_DATA_SET_UNIQUE_CHARACTER
This unique character is used as the default only when a first-time user profile is created.

The default is 1.

LIST_SYSOUT_CLASS
List data set sysout class.

The default is NONE (blank).

RESET_LIST_SYSOUT_CLASS
Reset the List data set sysout class.

The default is NO.

LOCAL_LIST_SYSOUT_CLASS
Local list data set sysout class.

The default is NONE (blank).

RESET_LOCAL_LIST_SYSOUT_CLASS
Reset Local list data set sysout class.

The default is NO.

LIST_DISPLAY_REQUIRED
List panel display required on termination. Valid values are YES or NO.

The default is NO.

LIST_KEPT
The list data set is to be kept. Valid values are YES or NO.

The default is NO.

LIST_LINES_PER_PAGE
The number of list lines per page.

The default is 60.

LIST_LINE_LENGTH
The line length of the list data set.

The default is 120.

LIST_LOCAL_PRINTER_ID
The local printer ID (CHAR(17)) for the list.

The default is NONE (blank).

RESET_LIST_LOCAL_PRINTER_ID
Reset the list local printer ID for the list. Valid values are YES or NO.

The default is NO.

LIST_RECORD_FORMAT
The record format for the list.

The default is FBA.

LIST_LOGICAL_RECORD_LENGTH
The logical record length for the list.

The default is 121.

LIST_PAGES_PRIMARY_QUANTITY
The primary quantity of list pages.

The default is 100.

LIST_PAGES_SECONDARY_QUANTITY
The secondary quantity of list pages.

The default is 200.

UNIQUE_JOB_CHARACTER
The unique job character.

The default is NONE (blank).

SCREEN_FORMAT
The screen format. The screen format you choose depends on the type of terminal you are using or the type of terminal your emulator is emulating. The following 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. On VM, this is the CMS file ID. The GDDM defaults for the CMS filename, filetype, and filemode are: GDDMPRNT ADMPRINT A1.

The default is NONE.

RESET_DEVICE_NAME
Reset the device name specified.

The default is NO.

ASPECT_RATIO
Allows the user to preserve the Graphics Aspect Ratio for a ″true″ picture (0) or to preserve the positional relationship between the graphics and alphanumerics (1).

The default is zero (0).

PAD_CHARACTER
The character entered here will be used to fill input fields on a panel. It

must be different than the command delimiter and it cannot be a-z, A-Z, 0-9 or /, except N and B can be used to indicate nulls and blanks respectively.

The default is B.

DELIMITER
The character entered here will be used to separate multiple commands entered on a command line. Alphabetic and alphanumeric characters as well as = (equal sign) and . (period) are not valid.

The default is **;** (semicolon).

RETRIEVE_MINIMUM_LENGTH
The minimum number of characters ISPF should save in the retrieve stack. Valid values are 1 through 99.

The default is 1.

RETRIEVE_CURSOR_POSITION
Cursor position relative to the retrieved command. Valid values are:
1 Place cursor at the beginning of the string.
2 Place cursor at the end of the string.

The default is 1.

ENABLE_DUMP
Enable a dump for a subtask ABEND when not in ISPF TEST mode. Valid values are ON and OFF.

The default is OFF.

TERMTRAC_DD_NAME
The terminal tracing (TERMTRAC) DD name.

The default is ISPSNAP.

RESET_TERMTRAC_DD_NAME
Reset the TERMTRAC DD name 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 1 (Blue).

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 512–32767. If defined as 0, ISPF will use its own calculations to determine the TPUT buffer size. If defined within the range 512–32767, 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 specify all defaults colors except substituting blue for green, specify the number for blue (**1**) in the position for green (4th): 123**1**567.

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 7 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 above. 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* whereever the host displays the color *Blue*. The default sequence maps the colors as:

Host color	Workstation color
Blue	Dark Blue
Red	Dark Red
Pink	Light Magenta
Green	Dark Green
Turquoise	Dark Cyan
Yellow	Brown
White	Black

HOST_COLORS
Select host colors for panel fields when on the workstation. Valid values are ON (selected) or OFF (not selected).

The default is OFF (not selected).

Note: HOST_COLORS and PC_COLORS (see below) are mutually exclusive. If one is selected, DO NOT select the other as well.

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 (see below) are mutually exclusive. If one is selected, DO NOT select the other as well.

DEFAULT_MESSAGE_ID
Select to display the message identifier. The valid values are OFF or ON, the default is OFF.

DEFAULT_PANEL_ID
Select to display the panel identifier. The valid values are OFF or ON, the default is OFF.

DEFAULT_SCREEN_NAME
Select to display the screen name. The valid values are OFF or ON.

The default is OFF.

DEFAULT_SYSTEM_NAME
Select to display the system name. The valid values are OFF or ON.

The default is OFF.

DEFAULT_USERID_DISP
Select to display the user identifier. The valid values are OFF or ON.

The default is OFF.

DEFAULT_PRIMARY_PANEL
Enter a panel name for the default primary panel.

The default is ISP@MSTR.

DEFAULT_LIBDEF_PROCESSING_OPTION
Enter the default option for processing LIBDEF requests. The valid values are COND, UNCOND, STACK, or STKADD.

The default is UNCOND.

ENABLE_EURO_SYMBOL
Enable the Euro currency symbol. Valid values are YES or NO.

The default is NO.

RESET_ENABLE_EURO_SYMBOL
Reset enable the Euro currency symbol field.

The default is NO.

DATE_FORMAT_ZDATEFD
The date format using the national language convention for the day, month, year and the national language separator. If the value is DEFAULT ISPF will use the value defined in the ISPF Literal Load Module.

The default is DEFAULT.

DATE_FORMAT_ZDATEF
The date format using the characters DD for day, MM for month, and YY for year and the national language separator. The order of year month and day may change. If the value is DEFAULT, ISPF uses the value defined in the ISPF Literal Load Module.

The default is DEFAULT.

DEFAULT_TIME_SEPARATOR
The separator used in the time of day format. For example, the colon (**:**) in hh**:**mm**:**ss. If the value is D, ISPF uses the value defined in the ISPF Literal Load Module.

The default is the D.

ISPSPROF Workstation Defaults

GUI_APPC_ADDRESS
The workstation's APPC network name in fully qualified LU name format or in symbolic destination name format.

The default is NONE.

GUI_APPC_ADDRESS2
The workstation's APPC network name in fully qualified LU name format or in symbolic destination name format.

The default is NONE.

GUI_APPC_ADDRESS3
The workstation's APPC network name in fully qualified LU name format or in symbolic destination name format.

The default is NONE.

GUI_APPC_ADDRESS4
The workstation's APPC network name in fully qualified LU name format or in symbolic destination name format.

The default is NONE.

GUI_APPC_ADDRESS5
The workstation's APPC network name in fully qualified LU name format or in symbolic destination name format.

The default is NONE.

GUI_CLOSE_WINDOW
The command to be processed when a GUI window is closed. Valid values are:

1 CANCEL

2 END

3 EXIT

4 RETURN

The default value is 1 (CANCEL).

GUI_DISPLAY_ENTER_KEY
If YES the enter key button will be displayed at the bottom of GUI panels. Valid values are YES or NO.

The default is YES.

GUI_WINDOW_FRAME
Select the type of GUI window frame. Valid values are:

1 Standard a GUI window frame that can be resized and has max/min buttons

2 Fixed a GUI window frame that has max/min buttons, but cannot be resized

3 Dialog a GUI window frame that cannot be resized and does not have max/min buttons

The default is 1.

GUI_NETWORK
Select the network protocol to be used for the GUI session. Valid values are:

1 TCP/IP Transmission Control Protocol/Internet Protocol

2 APPC Advanced Program-to-Program Communication

3 ISPDTPRF file
This option tells ISPF to look for the file allocated to DD ISPDTPRF. ISPF will search this file and use the network protocol and address specified for your userid.

The default is 1.

GUI_SWITCH
A connection to the workstation will be established and subsequent panels will be displayed in GUI mode after the ENTER key is pressed. Valid values are:

1 GUI display

2 Non-GUI display

The default is 1, GUI display.

SAVE_GUI_VALUES
Save the values entered on ISPF Settings Initiate the Workstation Connection panel in the system profile. Valid values are YES or NO.

The default value is YES.

GUI_TCP_ADDRESS
The TCP/IP address of the workstation in which the GUI display or connection should be made. The TCP/IP address can be in dotted decimal format or in domain name format.

The default is NONE.

GUI_TCP_ADDRESS2
The TCP/IP address of the workstation in which the GUI display or connection should be made. The TCP/IP address can be in dotted decimal format or in domain name format.

The default is NONE.

GUI_TCP_ADDRESS3
The TCP/IP address of the workstation in which the GUI display or connection should be made. The TCP/IP address can be in dotted decimal format or in domain name format.

The default is NONE.

GUI_TCP_ADDRESS4
The TCP/IP address of the workstation in which the GUI display or connection should be made. The TCP/IP address can be in dotted decimal format or in domain name format.

The default is NONE.

GUI_TCP_ADDRESS5
The TCP/IP address of the workstation in which the GUI display or connection should be made. The TCP/IP address can be in dotted decimal format or in domain name format.

The default is NONE.

GUI_TITLE
The GUI title is the default value to be used for the title bar. If the GUI title is blank the title bar will be set to the userid from where the dialog was initiated.

The default is NONE (blank).

GUI_ACCELERATOR_SUPPORT
When running in GUI mode and Accelerator Support is YES, any accelerator keys that are defined for the pulldown menus will be appended to the pulldown choice text and the accelerator keys will be functional. Valid values are YES or NO.

The default is YES.

HOST_CODE_PAGE and HOST_CHARACTER_SET
Both the Code Page (CP) and the Character Set (CS) must be specified in order to be used. If specified they are used in translating the data from

host to workstation. If the code page and character set are specified as 0000, values from the terminal query will be used. If your terminal/emulator does not support code pages, the CODEPAGE and CHARSET parameter values on ISPSTART will be used. The default will be English (U.S.) if these ISPSTART parameters are not specified.

The default is 0000 for both.

For more information about code pages and character sets refer to the *z/OS ISPF Dialog Developer's Guide and Reference*.

CP & CS	Used by countries
0037 0697	United States, Canada, Netherlands, Portugal, Brazil, Australia, New Zealand
0273 0697	Austria, Germany
0280 0697	Italy
0277 0697	Denmark, Norway
0278 0697	Finland, Sweden
0284 0697	Spain, Latin America (Spanish)
0285 0697	United Kingdom
0297 0697	France
0500 0697	Belgium, Switzerland
0870 0959	Latin 2
0880 0960	Cyrillic
1025 1150	Cyrillic
290 1172	Japanese (Katakana)
1027 1172	Japanese (Latin)
833 1173	Korean
836 1174	Simplified Chinese
037 1175	Traditional Chinese

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

CP & CS	Used by countries
1140 0695	United States, Canada, Netherlands, Portugal, Brazil, Australia, New Zealand
1141 0695	Austria, Germany
1144 0695	Italy
1145 0695	Spain, Latin America (Spanish)
1146 0695	United Kingdom
1147 0695	France
1148 0695	Belgium, Canada, Switzerland
1149 0695	Iceland

RESET_HOST_CODE_PAGE
Reset the host code page value.

The default is NO.

RESET_HOST_CHARACTER_SET
Reset the host character set value.

The default is NO.

GUI_DEFAULT_BACKGROUND_COLOR
Allows you to specify the background color for the display of GUI panels. The valid values are: 1 (Dialog) and 2 (Standard).

The default is 1.

GUI_DOWNLOAD_IMAGES
When running in GUI mode and Download Images is YES, ISPF will attempt to download images from the host libraries defined to ddname ISPILIB. When Download Images is NO, ISPF will not download images from the host. Valid values are YES or NO.

The default is YES.

GUI_MAKEPATH_FOR_IMAGES
When running in GUI mode and Make Path for Images is YES and the image path you have specified does not exist, ISPF will attempt to create the path. Valid values are YES or NO.

The default is NO.

CONTINUE_3270_AFTER_LOSS_OF_WS_CONNECTION
If the ″3270 Mode after losing workstation connection″ setting is selected, the ISPF session continues in the 3270 emulator session rather than abend with code 989 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 2000/NT
2	Reserved
3	AIX
4	Solaris
5	HP UX

The default is NONE.

WSA_DOWNLOAD_FTP_USER
The name of the workstation agent download FTP user.

The default is NONE.

Improving ISPF Performance

To improve ISPF performance consider using the following methods, which are described in detail in this section:

- Use virtual I/O (VIO). See page 75.
- Remove or tailor edit functions. See page 75.
- Remove action bars from ISPF panels. See page 76.
- Remove scrollable areas from ISPF panels. See page 76.
- Preprocess ISPF panels. See page 77.

- Add load modules to LPALST. See page 78.
- Allocate execution data sets. See page 90.
- Disable generic searches. See page 91.
- Preallocate ISPF temporary data sets to VIO. See page 93.

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 shipped 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 "The ISPF Configuration Table" on page 7 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 "The ISPF Configuration Table" on page 7 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 "The ISPF Configuration Table" on page 7 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 "The ISPF Configuration Table" on page 7 for more information about the ISPF configuration table.

Remove Action Bars from ISPF Panels

Some of the panels that are shipped 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 99.

Remove Scrollable Areas From ISPF Panels

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

1. Invoke ISPDTLC with the MERGESAREA option by specifying:

   ```
   ISPDTLC (MERGESAREA
   ```

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

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

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

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

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

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

For more information about invoking ISPDTLC, see “Invoking the ISPDTLC Conversion Utility” on page 99.

Preprocess All ISPF Panels

Preprocessed panels are displayed much faster than if they are not preprocessed. However, any panel 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 version of the panels are not updated by PTFs from IBM. If a panel is updated by a PTF, it must be preprocessed 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*.

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

The following 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 particular 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. Refer to 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. Refer to 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.

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

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

If ISPPREP is passed a panel definition that does not meet the above 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)

After 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 (in the SISPLOAD library) 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.

No modules are required in the LPA. Table 1 on page 80 lists the load modules recommended for inclusion in the LPA.

To conserve LPA space, you can omit the following modules:

- Move some of the infrequently used modules to a system-link library. Keep the modules you most frequently use in the LPA.
- Move the ISPF NLS load modules for the languages that you are not using to a system link library. See Table 4 on page 87 for a list of the ISPF NLS load modules.
- 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. The usermods are in the SISPSAMP data set. Usermod ISPMOVE1 contains ++MOVE statements to move load modules from SISPLPA to SISPLOAD. Usermod ISPMOVE2 contains ++MOVE statements to move load modules from SISPLOAD to SISPLPA. Before using these members, you must edit them for applicability to your environment. Use only those statements that apply to the load modules that you want to move from one library to another.

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

- 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. (See Table 3 on page 81 for load module names and their aliases.) 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.

ISPF Load Module Descriptions

This section contains tables describing ISPF load modules.

- Table 1 on page 80 lists the frequently used ISPF load modules that should be in the LPA and a brief description of each.
- Table 2 on page 81 lists the base ISPF load modules that cannot be placed in the LPA and a brief description of each.
- Table 3 on page 81 lists the ISPF base load modules that can be placed in the LPA, a brief description of each, and an indication of whether the module resides below the 16 MB line.
- Table 4 on page 87 lists the ISPF NLS load modules that can be placed in the LPA, a brief description of each, and an indication of whether the module resides below the 16 MB line.

Table 1. Minimum Recommended Load Modules for the LPA

Load Module	Description
FLM$CPI	SCLM
FLMB	SCLM
FLMCPCS	SCLM
FLMDDL	SCLM
FLMIO24	SCLM
FLMP	SCLM
FLMRTLIB	SCLM
FLMS$LNK	SCLM
FLMS7C	SCLM
ISPICP	ISPF driver (alias name: ISPSTART)
ISPISM	Settings processor
ISPKEY	KEYLIST command processor
ISPKLU	KEYLIST Utility
ISPLLP	Log/List processor
ISPMAIN	ISPF controller
ISPNLxxx	where xxx designates the languages you are using
ISPOMI	MSGID processor
ISPOPF	PFSHOW processor
ISPOPI	Panel ID processor
ISPOPT	Options processor
ISPQ	ISPF C/S main
ISPQGUI	ISPF C/S processor
ISPSUBS	Common subroutines (alias name: ISPCIP)
ISPSUBX	Common subroutines
ISPTASK	Dialog driver
ISPTCM	TSO command table
ISPTUTOR	Tutorial processor (option T)
ISPYLxxx	where xxx designates the languages you are using
ISP32xxx	where xxx designates the terminal types in use
ISRBRO	Main Browse module
ISREDIT	Main Edit module
ISRNLxxx	where xxx designates the languages you are using
ISRPLEX	Edit extended highlight module
ISRPX	Edit extended highlight customization module
ISRRCL	Action bar router utility
ISRSEPRM	More SuperC
ISRSUBS	Main PDF subroutine module
ISRSUBX	Main below 16M PDF subroutine module
ISRSUPC	Superc
ISRUDA	PDF Utilities
ISRUMC	PDF Move Copy
ISR32xxx	where xxx designates the terminal types in use

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

Table 2. ISPF LPA-Ineligible Load Modules

Load Module	Residence	Description
FLMNLDES		SCLM Swiss German literals module
FLMNLDEU		SCLM German literals module
FLMNLENP		SCLM Uppercase English literals module
FLMNLENU		SCLM English literals module
FLMNLJPN		SCLM Japanese literals module
FLMPALST	Below	SCLM Architecture definition parser
FLMPBOOK	Below	SCLM BookMaster parser
FLMPC370	Below	SCLM C/370 parser
FLMPDBRM	Below	SCLM CSP DBRM parser
FLMPJOV	Below	SCLM JOVIAL parser
FLMPPAS	Below	SCLM PASCAL parser
FLMPSCRP	Below	SCLM SCRIPT parser
FLMPS1A	Below	SCLM Series/1 assembler parser
FLMTABLE	Below	SCLM Session control table
ISPVCALR	Below	Call/Return trace
ISPVSTAT	Below	CPU timing exit for ISPVCALL
ISPVVDR	Below	VDEFINE/VDELETE trace

Table 3. ISPF LPA-Eligible Load Modules

Load Module	Residence	Description
FLM$CP	Below	Alias of FLMIO24
FLM$CPI		SCLM COPY service interface
FLM$DE	Below	Alias of FLMIO24
FLM$DT	Below	Alias of FLMIO24
FLM$99	Below	Alias of FLMIO24
FLMB		SCLM Build processor
FLMBCMD		Alias of FLMDDL
FLMBD$		Alias of FLMDDL
FLMCMD		Alias of FLMS7C
FLMCPCS		SCLM Copy processor
FLMCSLNK	Below	Alias of FLMIO24
FLMCSPDB	Below	SCLM CSP DB2* Bind Translator
FLMCXCMD	Below	Alias of FLMIO24
FLMCXCPD	Below	Alias of FLMIO24
FLMCXCPM	Below	Alias of FLMIO24
FLMCXCTN	Below	Alias of FLMIO24
FLMCXGPD	Below	Alias of FLMIO24
FLMCXMLS	Below	Alias of FLMIO24
FLMCXPLS	Below	Alias of FLMIO24
FLMCXRDI	Below	Alias of FLMIO24
FLMCXSLM	Below	Alias of FLMIO24
FLMCXSSR	Below	Alias of FLMIO24
FLMCXTPT	Below	Alias of FLMIO24
FLMCXUDI	Below	Alias of FLMIO24
FLMCXXDD	Below	Alias of FLMIO24
FLMDDL		SCLM Dialog interface
FLME$CAN		Alias of FLMDDL
FLME$CRT		Alias of FLMDDL
FLME$EDT		Alias of FLMDDL
FLME$END		Alias of FLMDDL
FLME$IM		Alias of FLMDDL
FLME$MOV		Alias of FLMDDL

Table 3. ISPF LPA-Eligible Load Modules (continued)

Load Module	Residence	Description
FLME$PRO		Alias of FLMDDL
FLME$REP		Alias of FLMDDL
FLME$RET		Alias of FLMDDL
FLME$SAV		Alias of FLMDDL
FLME$SMO		Alias of FLMDDL
FLME$SRE		Alias of FLMDDL
FLMEB$		Alias of FLMDDL
FLMED$		Alias of FLMDDL
FLMIO24	Below	SCLM I/O routines running in 24-bit mode
FLMLNK	Below	Alias of FLMS$LNK
FLMLPCBL		SCLM COBOL parser
FLMLPFRT		SCLM FORTRAN parser
FLMLPGEN		SCLM ASM/REXX/CLIST/PLI/TEXT parser
FLMLSS		SCLM SYNtran interface
FLMP		SCLM Promote processor
FLMPD$		Alias of FLMDDL
FLMPTC		SCLM TSO command processor
FLMRA		SCLM Architecture report
FLMRC		SCLM Systems contents utility
FLMRTLIB	Below	SCLM PASCAL Runtime library
FLMS$LNK	Below	SCLM Link interface to services
FLMS$SRV		SCLM services interface routine
FLMS7C		SCLM Command line processor
FLMTBMAP		SCLM Build Map INFO
FLMTCCPS		SCLM CSP translator
FLMTCIDS		SCLM CSP initialized dummy data set
FLMTCLGT		SCLM CSP LISTA and GENERATE
FLMTCPC		SCLM CSP COPY
FLMTCPP		SCLM CSP PURGE
FLMTCVER		SCLM CSP VERIFY
FLMTMMI	Below	SCLM DFSUNUB0 interface program
FLMTMSI	Below	SCLM Script interface program
FLMTPRE		SCLM Input list build program
FLMTPST		SCLM Input list compiler processor
FLMTXFER		SCLM Build Translator - send files to workstation
FLMUDU$		Alias of FLMDDL
FLMUM		SCLM Migration utility
FLMVCSUP	Below	SCLM SUPERCU (support for versioning)
FLMVUS		Alias of FLMDDL
FLMXE		SCLM Export Utility
FLMXI		SCLM Import Utility
IKJDTGET		Alias of ISPDTGET
IKJDTINI		Alias of ISPDTINI
IKJDTPUT		Alias of ISPDTPUT
IKJDTTER		Alias of ISPDTTER
ISPAPAUX		DM APL auxiliary processor
ISPAPTT		DM APL terminal type dialog
ISPASUBS		DM APL subroutines
ISPCAL		Calendar and note pad
ISPCCMDE		Command Line Extender
ISPCFIG		Initialization table for the ISPF Configuration table
ISPCIP		Alias of ISPSUBS

Table 3. ISPF LPA-Eligible Load Modules (continued)

Load Module	Residence	Description
ISPDTGET		Client/Server TSO line mode support
ISPDTINI		Client/Server TSO line mode support
ISPDTPC		Client/Server TSO line mode support
ISPDTPUT		Client/Server TSO line mode support
ISPDTSSI		Client/Server TSO line mode support
ISPDTTER		Client/Server TSO line mode support
ISPDTTSK		Client/Server TSO line mode support
ISPDTWIN		Client/Server TSO line mode support
ISPENV	Below	DM Environment command
ISPEX	Below	Alias of ISPLINK
ISPEXEC	Below	Alias of ISPLINK
ISPEXITS		DM User Exit Load Module
ISPF	Below	Alias of ISRPCP
ISPGWP		GUI builder interface
ISPICP	Below	DM ISPSTART command processor
ISPISM		Settings, Option 0
ISPKEY		DM KEYLIST command processor
ISPKLU		DM KEYLIST Utility
ISPLINK	See note	DM Module dialog interface
ISPLLP		DM Log/List processor
ISPLLS		Libdef display
ISPLMSG	Below	DM Module dialog interface message
ISPLNK		Alias of ISPLINK
ISPMAIN		DM Main controller
ISPNLENP		DM Uppercase English Literals module
ISPNLENU		DM English Literals module
ISPOMI		DM Message ID option
ISPOPF		DM PFSHOW processor
ISPOPI		DM Panel ID processor
ISPOPT		DM Option processor
ISPPNXRX		Rexx panel exit system interface
ISPPREP		DM Preprocessed panel utility
ISPPUP		DM Panel update program
ISPQ		ISPF C/S functions
ISPQAPI		External interface to C/S APIs
ISPQGUI		ISPF C/S functions
ISPQRY	Below	Alias of ISPLINK
ISPREXPX		Rexx panel exit user variable interface
ISPSAM		Status area on ISR@PRIM
ISPSTART	Below	Alias of ISPICP
ISPSTRT		DM start screen processor
ISPSUBS		DM Common subroutines - Alias of ISPCIP
ISPSUBX	Below	DM Common subroutines extension
ISPTASK		DM Processor main driver
ISPTCM		DM TSO command table
ISPTTC1		DM CCSID 00037 Translate table
ISPTTC2		DM CCSID 00273 Translate table
ISPTTC3		DM CCSID 00277 Translate table
ISPTTC4		DM CCSID 00278 Translate table
ISPTTC5		DM CCSID 00280 Translate table
ISPTTC6		DM CCSID 00284 Translate table
ISPTTC7		DM CCSID 00285 Translate table
ISPTTC8		DM CCSID 00297 Translate table

Table 3. ISPF LPA-Eligible Load Modules (continued)

Load Module	Residence	Description
ISPTTC9		DM CCSID 00500 Translate table
ISPTTC10		DM CCSID 00939 Translate table
ISPTTC11		DM CCSID 00930 Translate table
ISPTTC12		DM CCSID 00933 Translate table
ISPTTC13		DM CCSID 00935 Translate table
ISPTTC14		DM CCSID 00937 Translate table
ISPTTC15		DM CCSID 00870 Translate table
ISPTTC16		DM CCSID 00880 Translate table
ISPTTC17		DM CCSID 01025 Translate table
ISPTTC18		DM CCSID 00420 Translate table (place holder)
ISPTTC19		DM CCSID 00424 Translate table (place holder)
ISPTTC20		DM CCSID 00838 Translate table (place holder)
ISPTTC21		DM CCSID 00871 Translate table (place holder)
ISPTTC22		DM CCSID 00875 Translate table (place holder)
ISPTTC23		DM CCSID 01026 Translate table (place holder)
ISPTTC24		DM CCSID 04971 translate table
ISPTTC25		DM CCSID 16804 translate table
ISPTTC26		DM CCSID 05123 translate table
ISPTTC27		DM CCSID 08482 translate table
ISPTTC28		DM CCSID 12712 translate table
ISPTTC29		DM CCSID 01123 translate table
ISPTTC30		DM CCSID 01047 translate table
ISPTTC31		DM CCSID 00924 translate table
ISPTTC32		DM CCSID 01399 translate table
ISPTTC33		DM CCSID 01390 translate table
ISPTTC34		DM CCSID 01364 translate table
ISPTTC35		DM CCSID 01371 translate table
ISPTTC36		DM CCSID 01388 translate table
ISPTTC40		DM CCSID 01140 translate table
ISPTTC41		DM CCSID 01141 translate table
ISPTTC42		DM CCSID 01142 translate table
ISPTTC43		DM CCSID 01143 translate table
ISPTTC44		DM CCSID 01144 translate table
ISPTTC45		DM CCSID 01145 translate table
ISPTTC46		DM CCSID 01146 translate table
ISPTTC47		DM CCSID 01147 translate table
ISPTTC48		DM CCSID 01148 translate table
ISPTTC49		DM CCSID 01149 translate table
ISPTTC53		DM CCSID 01153 translate table
ISPTTC54		DM CCSID 01154 translate table
ISPTTC55		DM CCSID 01155 translate table
ISPTTC58		DM CCSID 01158 translate table
ISPTTC59		DM CCSID 01159 translate table
ISPTTC60		DM CCSID 01160 translate table
ISPTTDEF		DM Translate table/load module interface
ISPTUTOR		DM Tutorial processor
ISPUCM		DM Command table utility
ISPVCALL	Below	Call/Return Trace
ISPVSTG	Below	ISPF storage trace
ISPVVD	Below	Initialize and end trace
ISPVVLI		LINK CALL trace
ISPWFT		File Transfer Interface
ISPWSCD		Processor for WSCON or WSDISCON services

Table 3. ISPF LPA-Eligible Load Modules (continued)

Load Module	Residence	Description
ISPWSD		Client/Server component download
ISPYAPPL		Dialog test
ISPYBI		DM Dialog test Breakpoint interface
ISPYDTST		DM Dialog test Command interface
ISPYFI		DM Dialog test Invoke function
ISPNLENP		DM Dialog test Uppercase English Literals module
ISPYLENU		DM Dialog test English literals
ISPYLI		DM Dialog test Browse log interface
ISPYPI		DM Dialog test Display panel
ISPYRFI		DM Dialog test Function trace interface
ISPYRVI		DM Dialog test Variables trace interface
ISPYSI		DM Dialog test Invoke SPF dialog service processor
ISPYTI		DM Dialog test Table interface routine
ISPYVI		DM Dialog test Variables interface
ISPYXDR		DM Dialog test Test driver
ISPYXMB		DM Dialog test Main breakpoints
ISP00037		Alias of ISPTTC1
ISP00273		Alias of ISPTTC2
ISP00277		Alias of ISPTTC3
ISP00278		Alias of ISPTTC4
ISP00280		Alias of ISPTTC5
ISP00284		Alias of ISPTTC6
ISP00285		Alias of ISPTTC7
ISP00297		Alias of ISPTTC8
ISP00420		Alias of ISPTTC18
ISP00424		Alias of ISPTTC19
ISP00500		Alias of ISPTTC9
ISP00838		Alias of ISPTTC20
ISP00870		Alias of ISPTTC15
ISP00871		Alias of ISPTTC21
ISP00875		Alias of ISPTTC22
ISP00880		Alias of ISPTTC16
ISP00924		Alias of ISPTTC31
ISP00930		Alias of ISPTTC11
ISP00933		Alias of ISPTTC12
ISP00935		Alias of ISPTTC13
ISP00937		Alias of ISPTTC14
ISP00939		Alias of ISPTTC10
ISP01025		Alias of ISPTTC17
ISP01026		Alias of ISPTTC23
ISP01047		Alias of ISPTTC30
ISP01123		Alias of ISPTTC29
ISP01140		Alias of ISPTTC40
ISP01141		Alias of ISPTTC41
ISP01142		Alias of ISPTTC42
ISP01143		Alias of ISPTTC43
ISP01144		Alias of ISPTTC44
ISP01145		Alias of ISPTTC45
ISP01146		Alias of ISPTTC46
ISP01147		Alias of ISPTTC47
ISP01148		Alias of ISPTTC48
ISP01149		Alias of ISPTTC49
ISP01153		Alias of ISPTTC53

Table 3. ISPF LPA-Eligible Load Modules (continued)

Load Module	Residence	Description
ISP01154		Alias of ISPTTC54
ISP01155		Alias of ISPTTC55
ISP01158		Alias of ISPTTC58
ISP01159		Alias of ISPTTC59
ISP01160		Alias of ISPTTC60
ISP01364		Alias of ISPTTC34
ISP01371		Alias of ISPTTC35
ISP01388		Alias of ISPTTC36
ISP01390		Alias of ISPTTC33
ISP01399		Alias of ISPTTC32
ISP04971		Alias of ISPTTC24
ISP05123		Alias of ISPTTC26
ISP08482		Alias of ISPTTC27
ISP12712		Alias of ISPTTC28
ISP16804		Alias of ISPTTC25
ISP3277		DM 3277 translation tables
ISP3277A		DM 3277 APL translation tables
ISP3278		DM 3278/3279/3290 translation tables
ISP3278A		DM 3278/3279/3290 APL translation tables
ISP3278K		DM 3278 KATAKANA translation tables
ISP3278T		DM 3278/3279 text translation tables
ISRALTDI		PDF Alternate dialog selection
ISRAUTOT		Autotype data set/member name retrieval
ISRBRO		PDF Browse driver
ISRBR14		Branch to REG 14
ISRCLIPB		Edit clipboard manager
ISRDDN		PDF DD Name list
ISRDSLST		REFLIST command
ISRECMBR		PDF Edit model routine for member name
ISREDIT		PDF Edit driver
ISREMSPY	Below	Show edit data during macro execution
ISREUDBG		PDF Edit UNDO debugger
ISRFLMGI	Below	SCLM GENERATE info file module
ISRFMT		PDF Format utility functions
ISRFPR		PDF Foreground processor routine
ISRFPT		PDF Foreground print routine
ISRINT		PDF workstation integration driver
ISRJB1		PDF Batch 1 processor
ISRJB2		PDF Batch 2 processor
ISRLEMX	Below	PDF Member expansion
ISRLEQDS		PDF LMF Query data set entry
ISRLESS		PDF LMF Sort setup routine
ISRNLENP		PDF Uppercase English Literals module
ISRNLENU		PDF English literals module
ISRPCP	Below	PDF command processor
ISRPLEX		PDF Edit enhanced coloring processor
ISRPLEXP		Edit plex
ISRPLKWD		PDF Edit enhanced coloring keyword lists
ISRPTC		PDF TSO command processor (option 6)
ISRPX		PDF Edit language keywords module
ISRRCL		ROUTE command driver
ISRRLSLM		SCLM retrieve stack addition routine
ISRRSLST		REFLIST functions

Table 3. ISPF LPA-Eligible Load Modules (continued)

Load Module	Residence	Description
ISRSCAN		PDF Batch member copy
ISRSCLM		PDF SCLM Invocation
ISRSEPRM		PDF Extended SuperC processor
ISRSFM		PDF SuperC search-for processor
ISRSLENP	Below	PDF Uppercase English SuperC Literals module
ISRSLENU	Below	PDF English SuperC literals module
ISRSMAC		PDF SuperC initial macro
ISRSSM		PDF SuperC compare processor
ISRSUBS		PDF common subroutines
ISRSUBX	Below	PDF common subroutines extensions
ISRSUPC	Below	PDF SuperC Extended File/Line/Word and Byte compare utility
ISRUDA		PDF Utilities Data set driver
ISRUDL		Alias of ISRUDA
ISRUHC		PDF Utilities Hard-Copy
ISRUMC		PDF Utilities Move/copy
ISRUOLP		PDF Utilities Outlist
ISRURS		PDF Utilities Reset statistics
ISRUTABL		ISPF Table Utility
ISRUVCMA		VSAM utility panel driver
ISRUVEXE		VSAM utility edit macro
ISRUVMA		VSAM utility edit macro
ISR3277		PDF 3277 Translate table
ISR3277A		PDF 3277 APL Translate table
ISR3278		PDF 3278 Translate table
ISR3278A		PDF 3278 APL Translate table
ISR3278T		PDF 3278 Text Translate table
PDF	Below	Alias of ISRPCP

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 4. ISPF NLS LPA-Eligible Load Modules

Load Module	Residence	Description
ISPAA78		Reserved for future DM translate tables
ISPBB78		Reserved for future DM translate tables
ISPCC78		Reserved for future DM translate tables
ISPDD78		Reserved for future DM translate tables
ISPEE78		Reserved for future DM translate tables
ISPFF78		Reserved for future DM translate tables
ISPGG78		Reserved for future DM translate tables
ISPHH78		Reserved for future DM translate tables
ISPJJ78		Reserved for future DM translate tables
ISPKK78		Reserved for future DM translate tables
ISRAA78		Reserved for future PDF translate tables
ISRBB78		Reserved for future PDF translate tables
ISRCC78		Reserved for future PDF translate tables
ISRDD78		Reserved for future PDF translate tables

Table 4. ISPF NLS LPA-Eligible Load Modules (continued)

Load Module	Residence	Description
ISREE78		Reserved for future PDF translate tables
ISRFF78		Reserved for future PDF translate tables
ISRGG78		Reserved for future PDF translate tables
ISRHH78		Reserved for future PDF translate tables
ISRJJ78		Reserved for future PDF translate tables
ISRKK78		Reserved for future PDF translate tables
ISPAR78		Arabic DM 3278/3279/3290 translation tables
ISRAR78		Arabic PDF 3278 translation tables
ISPB678		Belgium DM 3278/3279/3290 base translation tables
ISRB678		Belgium PDF 3278 base translation tables
ISPB978		Belgium DM 3278/3279/3290 translation tables
ISRB978		Belgium PDF 3278 translation tables
ISPPO77		Brazilian Portuguese DM 3277 translation tables
ISPPO77A		Brazilian Portuguese DM 3277 APL translation tables
ISPPO78		Brazilian Portuguese DM 3278/3279/3290 translation tables
ISPPO78A		Brazilian Portuguese DM 3278/3279/3290 APL translation tables
ISPPO78T		Brazilian Portuguese DM 3278/3279 text translation tables
ISPNLPTB		Brazilian Portuguese DM Literals module
ISPYLPTB		Brazilian Portuguese DM Dialog Test literals module
ISRPO77		Brazilian Portuguese PDF 3277 Translate table
ISRPO77A		Brazilian Portuguese PDF 3277 APL Translate table
ISRPO78		Brazilian Portuguese PDF 3278 Translate table
ISRPO78A		Brazilian Portuguese PDF 3278 APL Translate table
ISRPO78T		Brazilian Portuguese PDF 3278 TEXT Translate table
ISPNLCHS		Chinese - Simplified DM Literals Module
ISPYLCHS		Chinese - Simplified DM Dialog test literals module
ISPNLCHT		Chinese - Traditional DM Literals Module
ISPYLCHT		Chinese - Traditional DM Dialog test literals module
ISPCY78		Cyrillic DM 3278/3279/3290 translation tables
ISRCY78		Cyrillic PDF 3278 translation tables
ISPCU78		Cyrillic Ukrainian - DM 3278 Translate tables
ISRCU78		Cyrillic Ukrainian - PDF 3278 translate tables
ISPDA77		Danish DM 3277 translation tables
ISPDA77A		Danish DM 3277 APL translation tables
ISPDA78		Danish DM 3278/3279/3290 translation tables
ISPDA78A		Danish DM 3278/3279/3290 APL translation tables
ISPDA78T		Danish DM 3278/3279 text translation tables
ISPNLDAN		Danish DM Literals module
ISPYLDAN		Danish DM Dialog test literals module
ISRDA77		Danish PDF 3277 Translate table
ISRDA77A		Danish PDF 3277 APL Translate table
ISRDA78		Danish PDF 3278 Translate table
ISRDA78A		Danish PDF 3278 APL Translate table
ISRDA78T		Danish PDF 3278 TEXT Translate table
ISRGE78T		English PDF 3278 Translate table with German special character and Euro support
ISPFR77		French DM 3277 translation tables
ISPFR77A		French DM 3277 APL translation tables
ISPFR78		French DM 3278/3279/3290 translation tables
ISPFR78A		French DM 3278/3279/3290 APL translation tables

Table 4. ISPF NLS LPA-Eligible Load Modules (continued)

Load Module	Residence	Description
ISPFR78T		French DM 3278/3279 text translation tables
ISPNLFRA		French DM Literals module
ISPYLFRA		French DM Dialog test literals module
ISP3278C		Can/French DM 3278 translation tables
ISRFR77		French PDF 3277 Translate table
ISRFR77A		French PDF 3277 APL Translate table
ISRFR78		French PDF 3278 Translate table
ISRFR78A		French PDF 3278 APL Translate table
ISRFR78T		French PDF 3278 TEXT Translate table
ISR3278C		Can/French PDF 3278 translation table
ISPGE78		German DM 3278/3279/3290 translation tables
ISPGE78A		German DM 3278/3279/3290 APL translation tables
ISPNLDEU		German DM Literals module
ISPYLDEU		German DM Dialog test literals module
ISRGE78		German PDF 3278 Translate table
ISRGE78A		German PDF 3278 APL Translate table
ISRNLDEU		German PDF literals module
ISRSLDEU	Below	German PDF SuperC literals module
ISPGR78		Greek - DM 3278 Translate table
ISRGR78		Greek - PDF 3278 Translate table
ISPHN78		Hebrew (new) DM 3278/3279/3290 translation tables
ISRHN78		Hebrew (new) PDF 3278 translation tables
ISPHO78		Hebrew (old) DM 3278/3279/3290 translation tables
ISRHO78		Hebrew (old) PDF 3278 translation tables
ISPIS78		Icelandic DM 3278/3279/3290 translation tables
ISRIS78		Icelandic PDF 3278 translation tables
ISPIT77		Italian DM 3277 translation tables
ISPIT77A		Italian DM 3277 APL translation tables
ISPIT78		Italian DM 3278/3279/3290 translation tables
ISPIT78A		Italian DM 3278/3279/3290 APL translation tables
ISPNLITA		Italian DM Literals module
ISPYLITA		Italian DM Dialog test literals module
ISRIT77		Italian PDF 3277 Translate table
ISRIT77A		Italian PDF 3277 APL Translate table
ISRIT78		Italian PDF 3278 Translate table
ISRIT78A		Italian PDF 3278 APL Translate table
ISPNLJPN		Japanese DM Literals module
ISPYLJPN		Japanese DM Dialog test literals module
ISRKA78K		Japanese PDF 5550 Translate table
ISRNLJPN		Japanese PDF literals module
ISRSLJPN	Below	Japanese PDF SuperC literals module
ISPKA78K		Katakana DM Translate tables
ISP3277K		Katakana DM 3277 translation tables
ISP3278K		Katakana DM 3278 translation tables
ISR3277K		Katakana PDF 3277 Translate table
ISR3278K		Katakana PDF 3278 Translate table
ISPHA78H		Korean DM Translate tables
ISPNLKOR		Korean DM Literals module
ISPYLKOR		Korean DM Dialog test literals module
ISRHA78H		Korean PDF 5550 Translate table
ISPL178		Latin-1 DM 3278 Translate table
ISRL178		Latin-1 PDF 3278 Translate table
ISPL278		Latin-2 DM 3278/3279/3290 translation tables

Table 4. ISPF NLS LPA-Eligible Load Modules (continued)

Load Module	Residence	Description
ISRL278		Latin-2 PDF 3278 Translate table
ISPSP77		Spanish DM 3277 translation tables
ISPSP77A		Spanish DM 3277 APL translation tables
ISPSP78		Spanish DM 3278/3279/3290 translation tables
ISPSP78A		Spanish DM 3278/3279/3290 APL translation tables
ISPSP78T		Spanish DM 3278/3279 text translation tables
ISPNLESP		Spanish DM Literals module
ISPYLESP		Spanish DM Dialog test literals module
ISRSP77		Spanish PDF 3277 Translate table
ISRSP77A		Spanish PDF 3277 APL Translate table
ISRSP78		Spanish PDF 3278 Translate table
ISRSP78A		Spanish PDF 3278 APL Translate table
ISRSP78T		Spanish PDF 3278 TEXT Translate table
ISPNLDES		Swiss German DM Literals Module
ISPSW116		Swiss German DM 116-Char. Tran. Table
ISPSW131		Swiss German DM 131-Char. Tran. Table
ISPSW500		Swiss German DM CECP 192-Char Tran. Table
ISPYLDES		Swiss German DM Dialog test Literals Module
ISRNLDES		Swiss German PDF literals module
ISRSLDES	Below	Swiss German PDF SuperC literals module
ISRSW116		Swiss German PDF 116 graphic table
ISRSW131		Swiss German PDF 131 graphic table
ISRSW500		Swiss German PDF 190 graphic table
ISPTH78		Thai - DM terminal translate tables
ISRTH78		Thai - PDF terminal translate tables

Allocate Execution Data Sets

Consider the following 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.

Disabling 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 224.

Figure 37 on page 92 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: This example is nonreentrant as written. If you put nonreentrant exits in the LPA, abends can occur.

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

Figure 37. Sample Exit to Disable Generic High-Level Qualifiers (Part 1 of 2)

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

Figure 37. Sample Exit to Disable Generic High-Level Qualifiers (Part 2 of 2)

Preallocate ISPF Temporary Data Sets to VIO

ISPF uses temporary data sets to generate JCL or utility control statements or to generate listings. To preallocate these data sets to VIO, include the DD statements in Figure 38 on page 94 in the TSO LOGON procedure.

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

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

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

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


//* In the above 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.

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

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

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

//* In the above 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 value of the keyword = 8,
//* only ISPWRK1 to ISPWRK8 need to be coded.

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

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

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

//* In the above 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 value of the keyword = 8,
//* only ISPLST1 to ISPLST8 need to be coded.
```

Figure 38. DD Statements to Preallocate Data Sets

Notes:

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

4. The ISPF temporary data set default names associated with the ISPWRK*x* are SPFTEMP*x*.WORK, respectively, where *x*= value 1-9, A-W.
5. The ddnames ISPWRK*x* are used by ISPF for file tailoring services with ISPFILE allocated to a PDS. The ddnames ISPLST*x* are used for generated listings.
6. The ddname ISPCTL0 is used with the edit SUBMIT command. The ddnames ISPCTLx are used with the PDF compress (both interactive and the LMCOMP service), PDF batch (option 5) and by ISPF for processing file tailoring service FTOPEN TEMP.
7. 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.
8. 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.
9. 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*.

Allocating Optional Image ISPF Library

The data set described in Table 5 is not a required ISPF library. You must allocate it only if an application in GUI mode uses images.

Table 5. Image Data Set

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

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

If you plan to use ISPF's image support, you must allocate the image input data set to ddname ISPILIB before using the images. You can accomplish this by allocating the image input data set before you invoke ISPF. Image files in the Graphic 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 the Primary Option Menu Panel

The ISPF Primary Option Menu panel (ISR@PRIM) contains choices for all of the options used by the ISPF product. The standard panel does not include other z/OS elements. There are sample panels in the SISPPENU data set that can be used to enable a primary option menu with most z/OS elements. Three panels are supplied for this purpose:

ISR@390

This is an ISPF Primary Option Menu panel. It is identical to the ISR@PRIM panel except that it includes additional options **12** and **13**, which point to the next two panels.

ISR@390S

This secondary panel is displayed when you select Option 12 from panel ISR@390. It contains options for z/OS elements that are used by system programmers and administrators. It includes options for:

- GDDM® Print Queue Manager
- HCD I/O configuration
- DCE configuration
- APPC Administration
- WLM Work Load Manager
- FFST™ dump formatting
- Infoprint Server
- RMF
- SMP/E
- TCP/IP NPF

ISR@390U

This secondary panel is displayed when you select Option 13 from panel ISR@390. It contains options for z/OS elements that are used by most ISPF users. It includes options for:

- BookManager® Build
- BookManager Read
- BookManager Index Creation
- DFSMSrmm™/ISMF
- DFSMSdfp™/ISMF
- DFSORT™
- BDT File-to-File
- IPCS
- z/OS UNIX Browse
- z/OS UNIX Edit
- z/OS UNIX Shell
- Security Server
- TSO/E Information Center Facility
- SDSF

Using the z/OS Sample Panels

To replace the standard ISPF Primary Option Menu panel (ISR@PRIM) with the z/OS version (ISR@390), use one of the following methods:

1. Use the DEFAULT_PRIMARY_PANEL option of the ISPF Configuration Table (refer to "ISPSPROF General Values" on page 62).
2. Place the following command into a PARM keyword on the EXEC statement for program IKJEFT01 in your logon PROC or logon CLIST.

   ```
   ISPSTART PANEL(ISR@390) NEWAPPL(ISR)
   ```

   Note: If you use this method the change to the z/OS panel is only temporary. Although the ISR@390 panel is displayed after logon, if you exit ISPF,

then enter the ISPF command from the TSO/E READY prompt, the ISR@PRIM panel is displayed instead of the ISR@390 panel.

You can place this command in a different CLIST to make it easier to return to the ISR@390 panel from a TSO/E READY prompt. For example, if you create such a CLIST called ISPF390 and place it in a library concatenated to SYSPROC, you can issue the ISPF390 command to return to the ISR@390 panel from the READY prompt.

3. Rename ISR@PRIM (for example, to @SR@PRIM). Rename ISR@390 to ISR@PRIM and identify the change with an SMP/E usermod, as shown in Figure 39.

```
//JOB
//*
//* Rerun Instructions:
//*    If this job is rerun, delete the IDCAMS step
//*    before submitting it.  Otherwise, the Delete
//*    command below will delete the only target library
//*    copy of ISR@PRIM
//*
//DELETE   EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
  DELETE -
    ISP.SISPPENU(@SR@PRIM)
  SET -
    MAXCC=0
  ALTER -
    ISP.SISPPENU(ISR@PRIM) -
    NEWNAME(ISP.SISPPENU(@SR@PRIM))
/*
//SMPE    EXEC PGM=GIMSMP,REGION=0M
//SMPCSI  DD DSN=oS390r5.global.csi,DISP=SHR
//SMPCNTL DD *
  SET BDY(GLOBAL) .
    RECEIVE
      S(OS39001) .
  SET BDY(targetzone) .
    APPLY
      S(OS39001) .
/*
//SMPPTFIN DD *
++USERMOD(OS39001) REWORK(yyyyddd).
++VER(Z038) FMID(FMID) PRE(RMID_of_ISR@PRIM) .
++PNLENU(ISR@PRIM) .
/*
//         DD DSN=ISP.SISPPENU(ISR@390),DISP=SHR
//
```

Figure 39. Sample Job, ISR@PRIM to ISR@390

Notes:

1. This usermod renames ISR@PRIM to @SR@PRIM and replaces ISR@PRIM with ISR@390.
2. This usermod changes the RMID level of ISR@PRIM. This RMID will not be identified as a prerequisite or superseded level by ISPF PTFs. SMP/E APPLY processing will fail for any PTF that replaces ISR@PRIM, which will alert you that the panel has changed. When this happens, the usermod above must be restored, the PTF installed, and the usermod reapplied.

3. To find the RMID of ISR@PRIM, use the SMP/E panels to display the PNLENU entry for ISR@PRIM in your target zone, or run an SMP/E batch job with these control statements:

```
SET BOUNDARY(target) .
  LIST PNLENU(ISR@PRIM) .
```

4. Refer to the *z/OS Program Directory* for the value of the FMID.
5. For the rework date, specify the current date in the format YYYYDDD.
6. Remove the PRE operand of the ++VER statement if the RMID and FMID of ISR@PRIM are equal.
7. When this usermod is installed for the first time, there will be no existing @SR@PRIM member of SISPPENU. This will result in these messages being issued by IDCAMS in the DELETE step:

   IDC33021 ** ACTION ERROR ON ISP.ISPPENU

   IDC33301 ** @SR@PRIM NOT FOUND

   IDC05481 ** MEMBER @SR@PRIM NOT DELETED

   These messages are expected; you can ignore them.
8. If the job fails and must be rerun, rename @SR@PRIM to ISR@PRIM or remove the IDCAMS step before submitting it again. Otherwise, the target library copy of ISR@PRIM will be deleted.

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 shipped in GML, as part of the DTL panel source.

To customize one or more action bar items:

1. Update the appropriate embed file from Table 6. These embed files are distributed in the ISP.SISPGENU library.

Table 6. Embedded DTL Source Files

File name	DTL File for...
ISPDFIL1	File action bar choice
ISPFMENU	SCLM menu action bar choice
ISPDFUNC	Functions action bar choice
ISPDHUCM	Action bar choice
ISPDHYXM	Action bar choice
ISPFJOBC	Jobcard action bar choice
ISPDLANG	Compilers action bar choice
ISPDMENU	Menu action bar choice
ISPDREFE	Reflist action bar choice
ISPDREFL	Reflist action bar choice
ISPDREFM	Refmode action bar choice
ISPDSAVE	Save action bar choice
ISPFSCLM	SCLM action bar choice
ISPDTEST	Test action bar choice
ISPDUTIL	Utilities action bar choice

2. Reconvert the affected panels by specifying the appropriate list of members (Table 7 on page 99) and running the ISPDTLC conversion utility.

Table 7. DTL List of Panels

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

The list of panels for different types of modifications are shipped in ISP.SISPSAMP. ISPDTLC will convert all of the members in a list that is used as an input member name.

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 77.

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 40 on page 100 shows the first screen of the ISPDTLC invocation panel with the default data set names and with the required options selected.

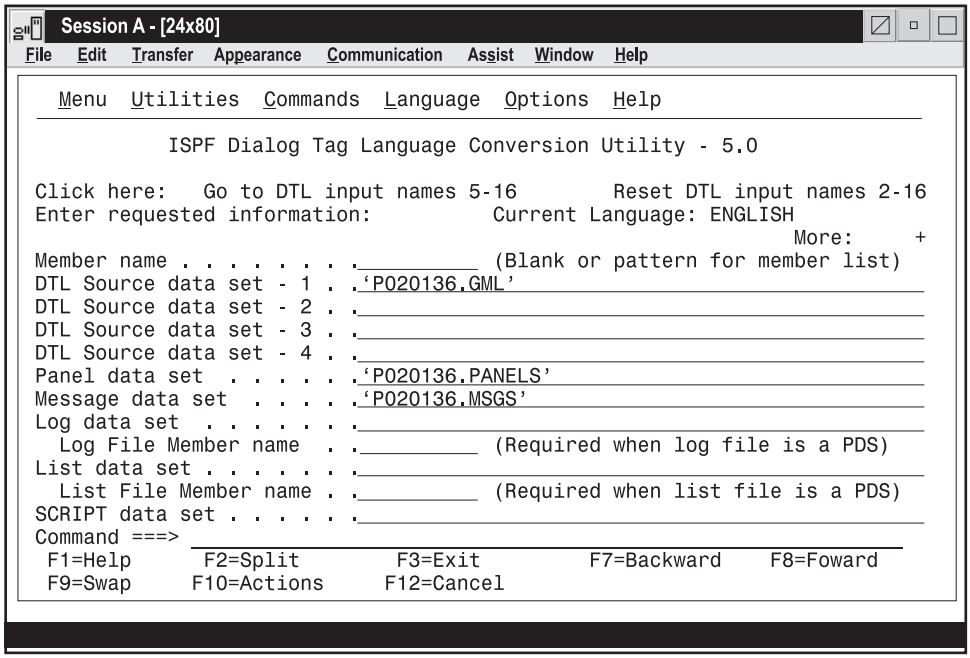


Figure 40. First ISPDTLC Screen

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

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

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

After typing in your file names, scroll to the next screen (Figure 41 on page 101).

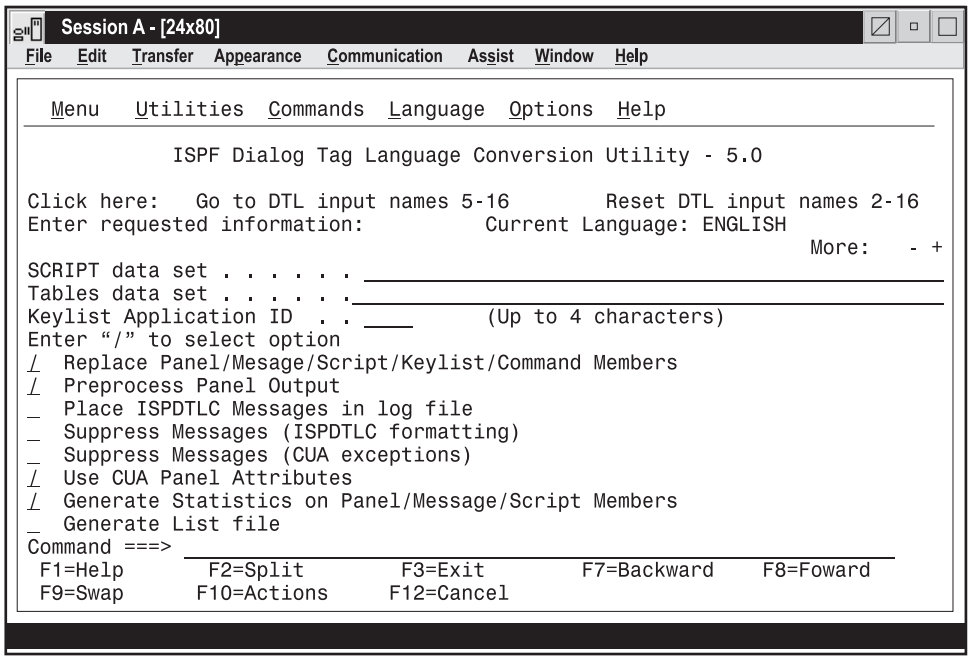


Figure 41. Second ISPDTLC Screen

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

Enter ISR as the Keylist Application ID and select the options as shown. Scroll to the next screen (Figure 42).

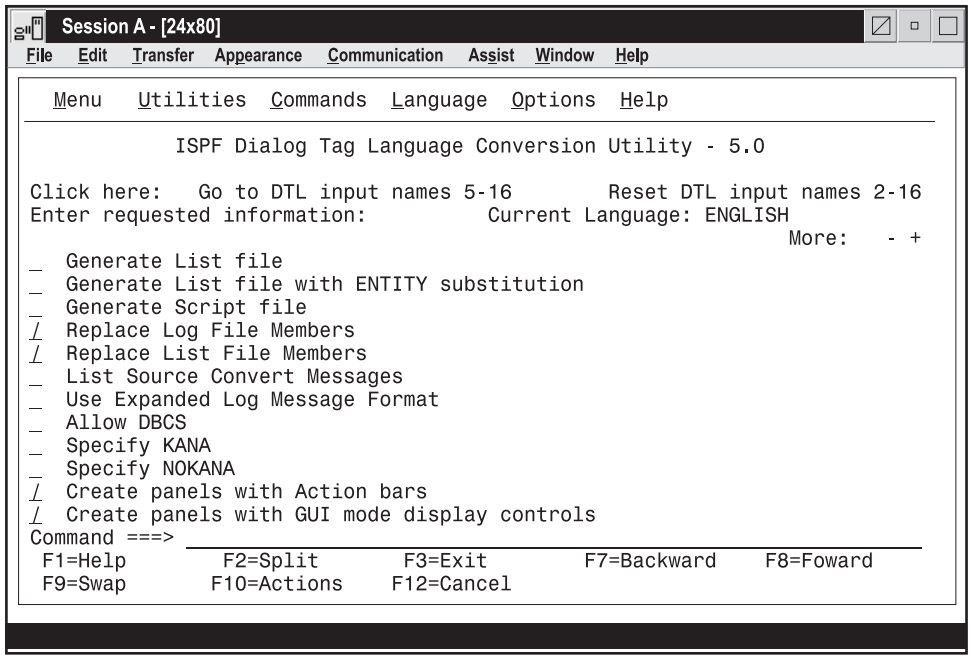


Figure 42. Third ISPDTLC Screen

Scroll to the next screen (Figure 43 on page 102).

```
 Menu  Utilities  Commands  Language  Options  Help
 ──────────────────────────────────────────────────────────────────────
          ISPF Dialog Tag Language Conversion Utility - 5.2

Click here:   Go to DTL input names 5-16        Reset DTL input names 2-16
Enter requested information:          Current Language: ENGLISH
                                                              More:   - +
/  Create panels with GUI mode display controls
/  Add ISPDTLC version / timestamp to panels and messages
_  Combine scrollable areas into panel )BODY section
_  Display converted panels
_  Display converted panels in a window
_  Bypass data set name validation (after first cycle)
/  Enable graphic character display
_  Use field names in place of Z variables
_  Align DBCS prompt text with entry field
/  Preserve leading ENTITY blanks when "space" is not specified
/  Process multiple line comment blocks
_  Display additional DTL source data set list
Command ===> ___________________________________________________________
 F1=Help      F2=Split         F3=Exit        F7=Backward    F8=Forward
 F9=Swap     F10=Actions      F12=Cancel
```

Figure 43. Fourth ISPDTLC Screen

Scroll to the last screen (Figure 44).

```
 Menu  Utilities  Commands  Language  Options  Help
 ──────────────────────────────────────────────────────────────────────
          ISPF Dialog Tag Language Conversion Utility - 5.2

Click here:   Go to DTL input names 5-16        Reset DTL input names 2-16
Enter requested information:          Current Language: ENGLISH
                                                              More:   - +
_  Display converted panels
_  Display converted panels in a window
_  Bypass data set name validation (after first cycle)
/  Enable graphic character display
_  Use field names in place of Z variables
_  Align DBCS prompt text with entry field
/  Preserve leading ENTITY blanks when "space" is not specified
/  Process multiple line comment blocks
_  Display additional DTL source data set list

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

Figure 44. Fifth ISPDTLC Screen

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

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 45 refers to the profile data set. The ISPDTLC invocation syntax specifies all of the required options for the conversion.

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

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

Figure 45. Sample JCL for Batch Conversion

The CONVACTB EXEC (written in REXX) is:

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

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

Notes:

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

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

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

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

your.gml.dataset(profile) should include:

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(profileu) 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.

Chapter 3. Customizing DM

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

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

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 "The ISPF Configuration Table" on page 7 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 support 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, do the following steps:

- The SAS/C runtime library, SISPSASC, must be in LPA or Linklist and be APF authorized. If you put the SAS/C runtime library in LPA, you must move ISPEVRG to a separate library that is in Linklist or STEPLIB. This is because ISPEVRG is not reentrant.
- The following ISPF modules and aliases must be in LPA: ISPDTSSI, ISPDTWIN, ISPDTTSK, ISPDTINI, IKJDTINI, ISPDTTER, IKJDTTER, ISPDTPUT, IKJDTPUT, ISPDTGET, IKJDTGET, ISPDTPC, ISPQ, and ISPQGUI.

- The ISPF subsystem initialization routine, ISPDTSSI, must be invoked at system IPL. To do this, add the following to your IEFSSNxx member in SYS1.PARMLIB.

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

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

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

- The following is required for TCP/IP communications:
  - z/OS Communications Service IP
  - Keyword USE_ZOS_UNIX_SOCKETS = Yes in the ISPF configuration table
  - user ID with an OMVS segment defined
  - default group with an OMVS segment defined
  - HFS home directory.

  See "Tailoring ISPF defaults" on page 132 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 8 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 8. ISPF Options and Related Module Names

Option	Module Names	TSO Commands
0	ISPISM	
1	ISRBRO	
2	ISREDIT	SUBMIT
3.1 and 3.2	ISRUDA	
3.3	ISRUMC	
3.4	ISRUDL	
3.5	ISRURS	
3.6	ISRUHC	SUBMIT, ICQCPC00
3.8	ISRUOLP	STATUS, OUTPUT, SUBMIT, ICQCPC00
3.9	ISPUCM	
3.10	ISRQCM	
3.11	ISRFMT	
3.12	ISRSSM	
3.13	ISRSEPRM	
3.14	ISRSFM	
4	ISRFPR	ASM, FORT, PLI, PLIC, LINK, TESTCOB, TESTFORT, CALL, SCRIPT, ALLOC, FREE, ICQCPC00
5	ISRJB1	SUBMIT
6	ISRPTC	All commands except those prohibited by ISPF in ISPTCM
7	ISPYXDR	
7.1	ISPYFI	
7.2	ISPYPI	
7.3	ISPYVI	
7.4	ISPYTI	
7.5	ISPYLI	
7.6	ISPYSI	
7.7.1	ISPYRFI	
7.7.2	ISPYRVI	
7.8	ISPYBI	
7.T	ISPTUTOR	
8.1	ISRLLDFI	
8.2	ISRLLUSI	
8.3	ISRLLDTI	
8.4	ISRLUACT	
8.5	ISRLALIN	
8.6	ISRLAUCL	
8.7	ISRLDSFA	
9	ISRALTDI	
10	FLMDDL	

Table 8. ISPF Options and Related Module Names (continued)

Option	Module Names	TSO Commands
C	ISPTUTOR	
T	ISPTUTOR	
Termination	ISPLLP	SUBMIT

Table 9 lists the ISPF commands with the related modules:

Table 9. ISPF Commands

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

Table 9. ISPF Commands (continued)

Command	Module Name
TUTOR	ISPTUTOR
ZKEYS	ISPOPT

Preallocation of List/Log data sets to SYSOUT

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

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

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

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

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

Notes:

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

Specifying the maximum number of split screens

ISPF can run up to 32 logical screens at one time. You can specify the maximum number of logical screens allowed for your installation by modifying the ISPF configuration table. See “The ISPF Configuration Table” on page 7 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 the following 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/HALF/MAX/PAGE
- SCROLL_MIN (minimum scroll value)
- SCROLL_MAX (maximum scroll value)
- STATUS_AREA_DEFAULT CAL/FUN/OFF/SES/UPS/USE (default is SES)

For the above 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 the following 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

The above list of defaults cannot be overridden by the end user.

See "The ISPF Configuration Table" on page 7 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 the following:

APPLID_FOR_USER_COMMAND_TABLE
The application ID for up to 3 user command tables. The default for each is NONE (no user command tables). The user command tables are searched after the command table for the current application, that is, the command table for the current APPLID, and before the site-wide and default system command tables.

APPLID_FOR_SITE_COMMAND_TABLE
The application ID for up to 3 site command tables. The default for each is NONE (no site-wide command tables). The search order for the site command tables depends on the SITE_COMMAND_TABLE_SEARCH_ORDER_SETTING.

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

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

See "The ISPF Configuration Table" on page 7 for more information about modifying the ISPF configuration table.

Application Command Table

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

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

User Command Tables

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

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

Site Command Tables

Commands in up to 3 site command tables are in effect for all applications. An option in the ISPF configuration table enables a site to specify that up to 3 site command tables are searched before or after the system command table. If the site command tables are searched first, then commands in the site command tables can override identically named commands in the system command table, and in turn can be overridden by identically named commands in the application or user command tables. If the site command tables are searched after the system command table, then their commands override no others, but can be overridden by identically named commands in either the application, user, or system command tables.

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

System Command Table

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

Program control facility modifications

If you have the Program Control Facility (PCF) licensed program (Program Number 5798-CLW) installed, update the PCF command list with the names of the new command processors (for example, ISPSTART, ISPF, and PDF) and the name of the ISPF dialog service command, ISPEXEC. The update allows invocation of these commands and eliminates an "Invalid Command" error message. If you performed this update when you installed a prior version of ISPF, you do not need to repeat it now.

Creating ISPF terminal translation tables

This section describes how to perform the following 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 the following 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 the above 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 the following 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 10) 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 10. Uppercase Character Translation Table Example

Table	Hexadecimal Code	Position
TTUPP	DC X'0001020304050607'	(X'00' to X'07')
	DC X'40090A0B0C0D0E0F'	(X'08' to X'0F')
	DC X'1011121314151617'	(X'10' to X'17')
	. . .	
	DC X'78797A7B7C7D7E7F'	(X'78' to X'7F')
	DC X'80C1C2C3C4C5C6C7'	(X'80' to X'87')
	. . .	
	DC X'E8E9EAEBECEDEEEF'	(X'E8' to X'EF')
	DC X'F0F1F2F3F4F5F6F7'	(X'F0' to X'F7')
	DC X'F8F9FAFBFCFDFEFF'	(X'F8' to X'FF')

Table 10 shows how the uppercase character translation table might be represented in the assembler module. For example, the following is true for Table 10:

- 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 11) represents display characters, in hexadecimal, as follows:

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

Table 11. Valid Terminal Output Translation Table Example

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

Generic String Master Translation Table

The positions in the generic string master translation table (TTGSM in the example shown in Table 12) 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 12. Generic String Master Translation Table Example

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

Modifying the GSM to Use the User Character Subset

This applies only if you have installed the PDF component. 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.

The following 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 173 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 pound sign (#), the dollar sign ($), and the at sign (@) (Table 13).
- TTALB, which does not include the pound sign (#), the dollar sign ($), and the at sign (@) (Table 14).

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 13. Sample Alphabetic Character Translation Table Including "#", "$", and "@".

Table	Hexadecimal Code	Position
TTALP	DC X'FFFFFFFFFFFFFFFF'	(X'00' to X'07')
	DC X'FFFFFFFFFFFFFFFF'	(X'08' to X'0F')
	DC X'FFFFFFFFFFFFFFFF'	(X'10' to X'17')
	...	
	DC X'FFFFFF0000FFFFFF'	(X'78' to X'7F')
	DC X'FF00000000000000'	(X'80' to X'87')
	...	
	DC X'0000FFFFFFFFFFFF'	(X'E8' to X'EF')
	DC X'FFFFFFFFFFFFFFFF'	(X'F0' to X'F7')
	DC X'0000FFFFFFFFFFFF'	(X'F8' to X'FF')

Table 14. Sample Alphabetic Character Translation Table Excluding "#", "$", and "@".

Table	Hexadecimal Code	Position
TTALB	DC X'FFFFFFFFFFFFFFFF'	(X'00' to X'07')
	DC X'FFFFFFFFFFFFFFFF'	(X'08' to X'0F')
	DC X'FFFFFFFFFFFFFFFF'	(X'10' to X'17')
	...	
	DC X'FFFFFFFFFFFFFFFF'	(X'50' to X'57')
	DC X'FFFFFFFFFFFFFFFF'	(X'58' to X'5F')
	...	
	DC X'FFFFFFFFFFFFFFFF'	(X'78' to X'7F')
	DC X'FF00000000000000'	(X'80' to X'87')
	DC X'0000FFFFFFFFFFFF'	(X'88' to X'8F')
	...	
	DC X'0000FFFFFFFFFFFF'	(X'E8' to X'EF')
	DC X'FFFFFFFFFFFFFFFF'	(X'F0' to X'F7')
	DC X'FFFFFFFFFFFFFFFF'	(X'F8' to X'FF')

Collating Sequence Translation Table

The collating sequence translation table (TTCOL in the example shown in Table 15 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. Refer to *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 15, 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 15 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 ‘B’, ‘a’ before ‘A’).

Table 15. Collating Sequence Translation Table Example

Table	Hexadecimal Code	Position
TTCOL	DC X'0001020304050607'	(X'00' to X'07')
	DC X'08090A0B0C0D0E0F'	(X'08' to X'0F')
	DC X'1011121314151617'	(X'10' to X'17')
	...	
	DC X'78797A7B7C7D7E7F'	(X'78' to X'7F')
	DC X'8081828384858687'	(X'80' to X'87')
	...	
	DC X'E8E9EAEBECEDEEEF'	(X'E8' to X'EF')
	DC X'F0F1F2F3F4F5F6F7'	(X'F0' to X'F7')
	DC X'F8F9FAFBFCFDFEFF'	(X'F8' to X'FF')

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

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

Specifying Terminal Types

You must tell ISPF which set of translation tables (which load module) to use. To do this, change the Dialog Tag Language source for ISPISMMN and, if you are still using the old options panels ISPOPT1 or ISPOPT1A, change the DTL source for these panels (see “Changing DTL source for ISPOPTxx Panels” on page 119). The user specifies the appropriate terminal type to refer to your set of translation tables. The load module is a member of a partitioned data set allocated to the ISPLLIB ddname, or it exists in STEPLIB, JOBLIB or LINKLIB.

Changing DTL source for ISPISMMN

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

The last “<SELFLD” tag in DTL source member ISPISMMN (see Figure 46 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 46. DTL source for Terminal Type selection - SELFLD tag

The following 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 47) and creates panel logic to determine which selection number will match a certain terminal type.

```
 Terminal Type   3    1. 3277       2. 3277A      3. 3278       4. 3278A
                      5. 3290A      6. 3278T      7. 3278CF     8. 3277KN
                      9. 3278KN    10. 3278AR    11. 3278CY    12. 3278HN
                     13. 3278HO    14. 3278IS    15. 3278L2    16. BE163
                     17. BE190     18. 3278TH    19. 3278CU    20. DEU78
                     21. DEU78A    22. DEU78T    23. DEU90A    24. SW116
                     25. SW131     26. SW500     27. 3278GR    28. 3278L1
                     29. XXXX
```

Figure 47. Example: Adding a Terminal Type to panel ISPISMMN

Modified DTL source for the English-language section of the member ISPZMMCH follows:

```
⋮
  <condexec lang=english>
    <choice selchar=1  checkvar=ztermp match=3277>3277       <:-- 01 -->
    <choice selchar=5  checkvar=ztermp match=3290A>3290A     <:-- 02 -->
    <choice selchar=9  checkvar=ztermp match=3278KN>3278KN   <:-- 03 -->
    <choice selchar=13 checkvar=ztermp match=3278HO>3278HO   <:-- 04 -->
    <choice selchar=17 checkvar=ztermp match=BE190>BE190     <:-- 05 -->
    <choice selchar=21 checkvar=ztermp match=DEU78A>DEU78A   <:-- 06 -->
    <choice selchar=25 checkvar=ztermp match=SW131>SW131     <:-- 07 -->
    <choice selchar=2  checkvar=ztermp match=3277A>3277A     <:-- 08 -->
    <choice selchar=6  checkvar=ztermp match=3278T>3278T     <:-- 09 -->
    <choice selchar=10 checkvar=ztermp match=3278AR>3278AR   <:-- 10 -->
    <choice selchar=14 checkvar=ztermp match=3278IS>3278IS   <:-- 11 -->
    <choice selchar=18 checkvar=ztermp match=3278TH>3278TH   <:-- 12 -->
    <choice selchar=22 checkvar=ztermp match=DEU78T>DEU78T   <:-- 13 -->
    <choice selchar=26 checkvar=ztermp match=SW500>SW500     <:-- 14 -->
    <choice selchar=3  checkvar=ztermp match=3278>3278       <:-- 15 -->
    <choice selchar=7  checkvar=ztermp match=3278CF>3278CF   <:-- 16 -->
    <choice selchar=11 checkvar=ztermp match=3278CY>3278CY   <:-- 17 -->
    <choice selchar=15 checkvar=ztermp match=3278L2>3278L2   <:-- 18 -->
    <choice selchar=19 checkvar=ztermp match=3278CU>3278CU   <:-- 19 -->
    <choice selchar=23 checkvar=ztermp match=DEU90A>DEU90A   <:-- 20 -->
    <choice selchar=27 checkvar=ztermp match=3278GR>3278GR   <:-- 21 -->
    <choice selchar=4  checkvar=ztermp match=3278A>3278A     <:-- 22 -->
    <choice selchar=8  checkvar=ztermp match=3277KN>3277KN   <:-- 23 -->
    <choice selchar=12 checkvar=ztermp match=3278HN>3278HN   <:-- 24 -->
    <choice selchar=16 checkvar=ztermp match=BE163>BE163     <:-- 25 -->
    <choice selchar=20 checkvar=ztermp match=DEU78>DEU78     <:-- 26 -->
    <choice selchar=24 checkvar=ztermp match=SW116>SW116     <:-- 27 -->
    <choice selchar=28 checkvar=ztermp 3278L1>3278L1         <:-- 28 -->
    <choice selchar=28 checkvar=ztermp match=XXXX>XXXX       <:-- 29 -->
  </condexec>
⋮
```

Figure 48. 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 46 on page 117 by the CHOICECOLS keyword). The above 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:

```
/* set translate load module name based on terminal type */
:
&ZCHARLM = TRANS(&ZTERM
                 3277  ,  ISP3277
                 3277A ,  ISP3277A
</source>
<condexec lang=english
<source>
                 3278  ,  ISP3278
                 3278A ,  ISP3278A
                 3290A ,  ISP3278A
                 3278T ,  ISP3278T
</source>
</condexec>
<source>
                 3278CF,  ISP3278C
                 3277KN,  ISP3277K
                 3278KN,  ISP3278K
                 3278AR,  ISPAR78
                 3278CY,  ISPCY78
                 3278HN,  ISPHN78
                 3278HO,  ISPHO78
                 3278IS,  ISPIS78
                 3278L2,  ISPL278
                 BE163,   ISPB678
                 BE190,   ISPB978
                 3278TH,  ISPTH78
                 3278CU,  ISPCU78
                 UE3278,  ISP3278
                 UE3278A, ISP3278A
                 UE3290A, ISP3278A
                 UE3278T, ISP3278T
                 DEU78,   ISPGE78
                 DEU78A,  ISPGE78A
                 DEU90A,  ISPGE78A
                 DEU78T,  ISPGE78T
                 SW116 ,  ISPSW116
                 SW131 ,  ISPSW131
                 SW500 ,  ISPSW500
                 3278L1,  ISPL178
                 3278GR,  ISPGR78
                 XXXX,    ISPOWNTT)
</source>
```

Figure 49. DTL source for Valid Terminal Types and Associated Load Module Names

In Figure 49, 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”. Refer to “Creating PDF Translation Tables” on page 170 for a discussion of PDF translation tables.

Changing DTL source for ISPOPTxx Panels

To change the Dialog Tag Language source for the ISPOPT1 and ISPOPT1A panels, use the Edit option to update the DTL source file imbed member ISPZPTTT. ISPZPTTT is used by the ISPOPT1 and ISPOPT1A panels. 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:

```
<source type=proc>
  VER (&ZTERM NB LIST
          3277,3277A,3278,3278A,3278T,3278CF,3277KN,3278KN,3290A,3278AR,
          BE163,BE190,3278CY,3278HN,3278HO,3278IS,3278L2,3278TH,3278CU,
          DEU78,DEU78A,DEU90A,SW116,SW131,SW500,3278L1,DEU78T,3278GR,
          UE3278,UE3278A,UE3290A,UE3278T,XXXX
          MSG=ISPO004)

  /* set translate load module name based on terminal type */
  &ZCHARLM = TRANS(&ZTERM
                3277  ,  ISP3277
                3277A ,  ISP3277A
                3278  ,  ISP3278
                3278A ,  ISP3278A
                3290A ,  ISP3278A
                3278T ,  ISP3278T
                3278CF,  ISP3278C
                3277KN,  ISP3277K
                3278KN,  ISP3278K
                3278AR,  ISPAR78
                3278CY,  ISPCY78
                3278HN,  ISPHN78
                3278HO,  ISPHO78
                3278IS,  ISPIS78
                3278L2,  ISPL278
                BE163,   ISPB678
                BE190,   ISPB978
                3278TH,  ISPTH78
                3278CU,  ISPCU78
                UE3278,  ISP3278
                UE3278A, ISP3278A
                UE3290A, ISP3278A
                UE3278T, ISP3278T
                DEU78,   ISPGE78
                DEU78T,  ISPGE78T
                DEU78A,  ISPGE78A
                DEU90A,  ISPGE78A
                SW116 ,  ISPSW116
                SW131 ,  ISPSW131
                SW500 ,  ISPSW500
                3278GR,  ISPGR78
                3278L1,  ISPL178
                XXXX,    ISPOWNTT)
</source>
```

Figure 50. ISPZPTTT modified to add new terminal type XXXX

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 the following EXTENDED CODE PAGE CCSIDs for the TRANS service and also with the use of the CCSID keyword on panels and messages.

Table 16. Extended CCSID1 Supported

CCSID	Character Set	Code Page	Country/Language
00037	697	37	U.S.A. Canada Netherlands Portugal Brazil Australia New Zealand
00273	697	273	Austria Germany
00277	697	277	Denmark Norway
00278	697	278	Finland Sweden
00280	697	280	Italy
00284	697	284	Spain L.A. Spanish
00285	697	285	United Kingdom
00297	697	297	France
00420	235	420	Arabic
00424	941	424	Hebrew
00500	697	500	Switzerland Belgium
00838	1176	838	Thailand
00870	959	870	Latin-2
00871	697	871	Iceland
00875	923	875	Greece
00880	960	880	Cyrillic
01025	1150	1025	Cyrillic

Table 16. Extended CCSID1 Supported (continued)

CCSID	Character Set	Code Page	Country/Language
01026	1126	1026	Turkey
01047	697	1047	Latin1
01123	1326	1123	Ukraine

Table 17. Extended CCSID1 Supported (EURO)

CCSID	Character Set	Code Page	Country/Language
00924	1353	0924	Latin9
01140	695	1140	U.S.A. Canada Netherlands Portugal Brazil Australia New Zealand
01141	695	1141	Austria Germany
01142	695	1142	Denmark Norway
01143	695	1143	Finland Sweden
01144	695	1144	Italy
01145	695	1145	Spain L.A. Spanish
01146	695	1146	United Kingdom
01147	695	1147	France
01148	695	1148	Switzerland Belgium
01149	695	1149	Iceland
01153	1375	1153	Latin2
01154	1381	1154	Cyrillic
01155	1378	1155	Turkey
01158	1388	1158	Ukraine
01160	1395	1160	Thailand
04899	1356	0803	Hebrew
04971	1371	0875	Greece
12712	1357	0424	Hebrew
16804	1461	0420	Arabic

The following Extended CCSIDs (shown in Table 18 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 18. Extended SBCS and DBCS CCSIDs Supported

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

Base Code Pages for Terminals

ISPF provides direct translation between each BASE CODE PAGE and its EXTENDED CODE PAGE for panels or messages. It also provides direct translation between extended Japanese (Latin or English) and both base Japanese (English) and base Japanese (Katakana). All translation between the single-byte EXTENDED CODE PAGEs for the double-byte languages and the CECP code pages is through CCSID 00500.

ISPF supports the base code pages (including mixed SBCS/DBCS CCSIDs for the DBCS languages) shown in Table 19.

Table 19. Base CCSIDs Supported

CCSID	Character Set	Code Page	Country/Language
00803	1147	424	Hebrew (Old)
00931	101	037	Japan (English)
04369	265	273	Germany and Austria
04371	273	275	Brazil
04373	281	277	Denmark and Norway
04374	285	278	Finland and Sweden
04376	293	280	Italy
04380	309	284	L.A. (Spanish Speaking)
04381	313	285	U.K. English
04393	1129	297	France
04934	938	838	Thailand
04966	959	870	Latin-2
04976	960	880	Cyrillic
05029	933	833	Korean
05031	936	836	Simplified Chinese

Table 19. Base CCSIDs Supported (continued)

CCSID	Character Set	Code Page	Country/Language
05033	101	037	Traditional Chinese
08229	101	037	U.S. English and Netherlands
08476	650	284	Spain
09122	332	290	Japan (Katakana)
41460	904	500	Switzerland
45556	908	500	Switzerland

ISPCCSID Translation Load Modules

ISPCCSID translation load modules translate data from one CCSID to another. There is one translation load module for each of the supported CCSIDs. The name, or alias, of each CCSID translation load module is made up of a 5-digit CCSID, prefixed with "ISP". For example, load module ISP00111 supports translation of the CCSID 00111. Each CCSID translation load module must contain at least two translation tables. These translation tables convert data between the respective CCSID and CCSID 00500. In addition each CCSID load module can contain up to 256 pairs of optional "direct" translation tables. ISPF uses the " direct" translation tables when available. Otherwise, ISPF translates the characters through CCSID 00500. Translating through CCSID 00500 can result in valid characters being lost as CCSID 00500 does not have all possible code points defined.

You can add direct "To" and "From" translation tables for direct translation to prevent possible loss of characters through CCSID 00500 for character sets other than 697 or to augment the extended code page translation tables provided by ISPF. The direct translation CCSID must be one of the CCSIDs supported by ISPF (see "Extended Code Page Translation Tables Provided by ISPF" on page 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 ISP*nnnnn*, 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 20 on page 125 and Table 21 on page 125 show examples of the "To" and "From" translation tables needed to translate characters between CCSID 00500 and CCSID 00037.

Table 20. Table for Translating from CCSID 00037 to CCSID 00500.

Table	Hexadecimal Code	Position
TO_500	DC X'4041424344454647'	(X'40' to X'47')
	DC X'4849B04B4C4D4EBB'	(X'48' to X'4F')
	DC X'5051525354555657'	(X'50' to X'57')
	DC X'58594F5B5C5D5EBA'	(X'58' to X'5F')
	. . .	
	DC X'78797A7B7C7D7E7F'	(X'78' to X'7F')
	DC X'8081828384858687'	(X'80' to X'87')
	. . .	
	DC X'E8E9EAEBECEDEEEF'	(X'E8' to X'EF')
	DC X'F0F1F2F3F4F5F6F7'	(X'F0' to X'F7')
	DC X'F8F9FAFBFCFDFE'	(X'F8' to X'FE')

Table 21. Table for Translating from CCSID 00500 to CCSID 00037.

Table	Hexadecimal Code	Position
FROM_500	DC X'4041424344454647'	(X'40' to X'47')
	DC X'4849BA4B4C4D4E5A'	(X'48' to X'4F')
	DC X'5051525354555657'	(X'50' to X'57')
	DC X'5859BB5B5C5D5EB0'	(X'58' to X'5F')
	. . .	
	DC X'78797A7B7C7D7E7F'	(X'78' to X'7F')
	DC X'8081828384858687'	(X'80' to X'87')
	. . .	
	DC X'E8E9EAEBECEDEEEF'	(X'E8' to X'EF')
	DC X'F0F1F2F3F4F5F6F7'	(X'F0' to X'F7')
	DC X'F8F9FAFBFCFDFE'	(X'F8' to X'FE')

The source for the above 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 22, which you can update. They are distributed in ISP.SISPSAMP.

Table 22. Translation Tables Provided with ISPF

Table name	CCSID	Description
ISPSTC1	00037	U.S.A, Canada, Netherlands, Portugal, Brazil, Australia, and New Zealand
ISPSTC2	00273	Austria and Germany
ISPSTC3	00277	Denmark and Norway
ISPSTC4	00278	Finland and Sweden
ISPSTC5	00280	Italy
ISPSTC6	00284	Spain and L.A. (Spanish-speaking)
ISPSTC7	00285	United Kingdom
ISPSTC8	00297	France
ISPSTC9	00500	Switzerland and Belgium
ISPSTC10	00939	Japan (Latin)
ISPSTC11	00930	Japan (Katakana)
ISPSTC12	00933	Korea
ISPSTC13	00935	Simplified Chinese
ISPSTC14	00937	Traditional Chinese

Table 22. Translation Tables Provided with ISPF (continued)

Table name	CCSID	Description
ISPSTC15	00870	Latin 2
ISPSTC16	00880	Cyrillic
ISPSTC17	01025	Cyrillic

ISPCCSID Translation Load Module Generation Macro

You can use the assembler macro, ISPCCSID, to generate custom ISPCCSID translation load modules. The macro also allows you to add "direct" translation tables to the ISPCCSID translation load modules ISPF supplies with the product. Calls to this macro must also be coded for the To_500 and From_500 tables and any "To" and "From" tables for direct translation. The load module must have either the name ISP*xxxxx* (where *xxxxx* is new CCSID) or an alias of ISP*xxxxx*. In both cases, the load module should be a CCSID defined in the Character Data Representation Architecture Registry.

Note that only the values for the hexadecimal digits X'40' through X'FE' are defined in a given translation table. These are the only code points that vary from CCSID to CCSID.

The first time you use the ISPCCSID macro, you must identify the CCSID of the ISPCCSID translation load module and provide the addresses of the "To" and "From" CCSID 00500 translation tables.

You can use the ISPCCSID macro again with the same ISPCCSID translation load module generation to identify the CCSID and translation table addresses of optional direct "To" and "From" translation tables.

The format of calls to the ISPCCSID assembler macro is:

```
ISPCCSID  CCSID=nnnnn,TO=to-address,FROM=from-address
```

The required parameters of the ISPCCSID macro are:

nnnnn
: This parameter is a 5-digit decimal (5 characters) number that specifies a CCSID. The *nnnnn* value on the first or only ISPCCSID macro definition is the CCSID associated with the ISPCCSID translation load module. The *nnnnn* value on other than the first ISPCCSID macro definition is the CCSID associated with direct "to" and "from" translation tables. If this parameter is not 5 digits, it causes an assembly error.

to-address
: On the first or only ISPCCSID macro definition, this parameter specifies the address of the translation table that converts data from the CCSID associated with the respective ISPCCSID translation load module to CCSID 00500. On subsequent ISPCCSID macro definitions within the same ISPCCSID translation load module, this parameter specifies the address of the translation table that converts data from the CCSID associated with the respective ISPCCSID translation load module to the CCSID specified on this ISPCCSID macro definition.

from-address
: On the first or only ISPCCSID macro definition, this parameter specifies the address of the translation table that converts data from CCSID 00500 to the CCSID associated with the respective ISPCCSID translation load module. On

subsequent ISPCCSID macro definitions within the same ISPCCSID translation load module, this parameter specifies the address of the translation table that converts data from the CCSID specified on this ISPCCSID macro definition to the CCSID associate with the respective ISPCCSID translation load module.

ISPCCSID Translation Load Module Definition Examples

Each ISPCCSID translation load module must be compiled separately using High Level Assembler (or a functional equivalent).

This example shows the ISPCCSID macro used with the Basic ISP00111 translation module.

```
         ISPCCSID CCSID=00111,TO=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
```

Example of User-Modifiable ISPF Translation Table

The following is the module for CCSID 00037 (ISPSTC1). The existing tables can be modified, or more pairs of direct translation tables can be added. To add direct translation tables, add a new ISPCCSID macro call for the new direct translation tables, and add the new tables. The assembler program should be renamed to ISPTTC*nn*, where *nn* is the last 1-digit or 2-digit number of the ISPSTC*nn* name. For example, ISPSTC1 should be renamed ISPTTC1, and ISPSTC14 renamed ISPTTC14.

```
*  THE FOLLOWING MACROS WILL GENERATE THE CCSID 00037 MODULE.
*
*
         ISPCCSID CCSID=00037,TO=TTC1T5H,FROM=TTC1F5H
         ISPCCSID CCSID=08229,TO=TTC1TB1,FROM=TTC1FB2
         ISPCCSID CCSID=04371,TO=TTC1TB2,FROM=TTC1FB2
*
*    TTC1T5H - CCSID 00037 TO CCSID 00500 Table
*
TTC1T5H  DS   0XL191
         DC X'4041424344454647'          (X'40' TO X'47')
         DC X'4849B04B4C4D4EBB'          (X'48' TO X'4F')
         DC X'5051525354555657'          (X'50' TO X'57')
         DC X'58594F5B5C5D5EBA'          (X'58' TO X'5F')
         DC X'6061626364656667'          (X'60' TO X'67')
         DC X'68696A6B6C6D6E6F'          (X'68' TO X'6F')
         DC X'7071727374757677'          (X'70' TO X'77')
         DC X'78797A7B7C7D7E7F'          (X'78' TO X'7F')
         DC X'8081828384858687'          (X'80' TO X'87')
         DC X'88898A8B8C8D8E8F'          (X'88' TO X'8F')
         DC X'9091929394959697'          (X'90' TO X'97')
         DC X'98999A9B9C9D9E9F'          (X'98' TO X'9F')
         DC X'A0A1A2A3A4A5A6A7'          (X'A0' TO X'A7')
```

```
         DC X'A8A9AAABACADAEAF'          (X'A8' TO X'AF')
         DC X'5FB1B2B3B4B5B6B7'          (X'B0' TO X'B7')
         DC X'B8B94A5ABCBDBEBF'          (X'B8' TO X'BF')
         DC X'C0C1C2C3C4C5C6C7'          (X'C0' TO X'C7')
         DC X'C8C9CACBCCCDCECF'          (X'C8' TO X'CF')
         DC X'D0D1D2D3D4D5D6D7'          (X'D0' TO X'D7')
         DC X'D8D9DADBDCDDDEDF'          (X'D8' TO X'DF')
         DC X'E0E1E2E3E4E5E6E7'          (X'E0' TO X'E7')
         DC X'E8E9EAEBECEDEEEF'          (X'E8' TO X'EF')
         DC X'F0F1F2F3F4F5F6F7'          (X'F0' TO X'F7')
         DC X'F8F9FAFBFCFDFE'            (X'F8' TO X'FE')
*
*    TTC1F5H - CCSID 00037 FROM CCSID 00500 Table
*
TTC1F5H  DS   0XL191
         DC X'4041424344454647'          (X'40' TO X'47')
         DC X'4849BA4B4C4D4E5A'          (X'48' TO X'4F')
         DC X'5051525354555657'          (X'50' TO X'57')
         DC X'5859BB5B5C5D5EB0'          (X'58' TO X'5F')
         DC X'6061626364656667'          (X'60' TO X'67')
         DC X'68696A6B6C6D6E6F'          (X'68' TO X'6F')
         DC X'7071727374757677'          (X'70' TO X'77')
         DC X'78797A7B7C7D7E7F'          (X'78' TO X'7F')
         DC X'8081828384858687'          (X'80' TO X'87')
         DC X'88898A8B8C8D8E8F'          (X'88' TO X'8F')
         DC X'9091929394959697'          (X'90' TO X'97')
         DC X'98999A9B9C9D9E9F'          (X'98' TO X'9F')
         DC X'A0A1A2A3A4A5A6A7'          (X'A0' TO X'A7')
         DC X'A8A9AAABACADAEAF'          (X'A8' TO X'AF')
         DC X'4AB1B2B3B4B5B6B7'          (X'B0' TO X'B7')
         DC X'B8B95F4FBCBDBEBF'          (X'B8' TO X'BF')
         DC X'C0C1C2C3C4C5C6C7'          (X'C0' TO X'C7')
         DC X'C8C9CACBCCCDCECF'          (X'C8' TO X'CF')
         DC X'D0D1D2D3D4D5D6D7'          (X'D0' TO X'D7')
         DC X'D8D9DADBDCDDDEDF'          (X'D8' TO X'DF')
         DC X'E0E1E2E3E4E5E6E7'          (X'E0' TO X'E7')
         DC X'E8E9EAEBECEDEEEF'          (X'E8' TO X'EF')
         DC X'F0F1F2F3F4F5F6F7'          (X'F0' TO X'F7')
         DC X'F8F9FAFBFCFDFE'            (X'F8' TO X'FE')
*
*    TTC1TB1 - CCSID 00037 TO CCSID 08229 Table
*
TTC1TB1  DS   0XL191
         DC X'404B4B4B4B4B4B4B'          (X'40' TO X'47')
         DC X'4B4B4A4B4C4D4E4F'          (X'48' TO X'4F')
         DC X'504B4B4B4B4B4B4B'          (X'50' TO X'57')
         DC X'4B4B5A5B5C5D5E5F'          (X'58' TO X'5F')
         DC X'60614B4B4B4B4B4B'          (X'60' TO X'67')
         DC X'4B4B6A6B6C6D6E6F'          (X'68' TO X'6F')
         DC X'4B4B4B4B4B4B4B4B'          (X'70' TO X'77')
         DC X'4B797A7B7C7D7E7F'          (X'78' TO X'7F')
         DC X'4B81828384858687'          (X'80' TO X'87')
         DC X'88894B4B4B4B4B4B'          (X'88' TO X'8F')
         DC X'4B91929394959697'          (X'90' TO X'97')
         DC X'98994B4B4B4B4B4B'          (X'98' TO X'9F')
         DC X'4BA1A2A3A4A5A6A7'          (X'A0' TO X'A7')
         DC X'A8A94B4B4B4B4B4B'          (X'A8' TO X'AF')
         DC X'4B4B4B4B4B4B4B4B'          (X'B0' TO X'B7')
         DC X'4B4B4B4B4B4B4B4B'          (X'B8' TO X'BF')
         DC X'C0C1C2C3C4C5C6C7'          (X'C0' TO X'C7')
         DC X'C8C94B4B4B4B4B4B'          (X'C8' TO X'CF')
         DC X'D0D1D2D3D4D5D6D7'          (X'D0' TO X'D7')
         DC X'D8D94B4B4B4B4B4B'          (X'D8' TO X'DF')
         DC X'E04BE2E3E4E5E6E7'          (X'E0' TO X'E7')
         DC X'E8E94B4B4B4B4B4B'          (X'E8' TO X'EF')
         DC X'F0F1F2F3F4F5F6F7'          (X'F0' TO X'F7')
         DC X'F8F94B4B4B4B4B'            (X'F8' TO X'FE')
```

```
*
*    TTC1FB1 - CCSID 00037 FROM CCSID 08229 Table
*
TTC1FB1  DS   0XL191
         DC X'4041424344454647'          (X'40' TO X'47')
         DC X'48494A4B4C4D4E4F'          (X'48' TO X'4F')
         DC X'5051525354555657'          (X'50' TO X'57')
         DC X'58595A5B5C5D5E5F'          (X'58' TO X'5F')
         DC X'6061626364656667'          (X'60' TO X'67')
         DC X'68696A6B6C6D6E6F'          (X'68' TO X'6F')
         DC X'7071727374757677'          (X'70' TO X'77')
         DC X'78797A7B7C7D7E7F'          (X'78' TO X'7F')
         DC X'8081828384858687'          (X'80' TO X'87')
         DC X'88898A8B8C8D8E8F'          (X'88' TO X'8F')
         DC X'9091929394959697'          (X'90' TO X'97')
         DC X'98999A9B9C9D9E9F'          (X'98' TO X'9F')
         DC X'A0A1A2A3A4A5A6A7'          (X'A0' TO X'A7')
         DC X'A8A9AAABACADAEAF'          (X'A8' TO X'AF')
         DC X'B0B1B2B3B4B5B6B7'          (X'B0' TO X'B7')
         DC X'B8B9BABBBCBDBEBF'          (X'B8' TO X'BF')
         DC X'C0C1C2C3C4C5C6C7'          (X'C0' TO X'C7')
         DC X'C8C9CACBCCCDCECF'          (X'C8' TO X'CF')
         DC X'D0D1D2D3D4D5D6D7'          (X'D0' TO X'D7')
         DC X'D8D9DADBDCDDDEDF'          (X'D8' TO X'DF')
         DC X'E0E1E2E3E4E5E6E7'          (X'E0' TO X'E7')
         DC X'E8E9EAEBECEDEEEF'          (X'E8' TO X'EF')
         DC X'F0F1F2F3F4F5F6F7'          (X'F0' TO X'F7')
         DC X'F8F9FAFBFCFDFE'            (X'F8' TO X'FE')
*
*    TTC1TB2 - CCSID 00037 TO CCSID 04371 Table
*
TTC1TB2  DS   0XL191
         DC X'404B4B4B4B4B794B'          (X'40' TO X'47')
         DC X'4B4B4B4B4C4D4E4B'          (X'48' TO X'4F')
         DC X'50D04B4B4B4B4B4B'          (X'50' TO X'57')
         DC X'4B4B4F5A5C5D5E4B'          (X'58' TO X'5F')
         DC X'60614B4B4B4B7C4B'          (X'60' TO X'67')
         DC X'5B4B4B6B6C6D6E6F'          (X'68' TO X'6F')
         DC X'4B4A4B4B4B4B4B4B'          (X'70' TO X'77')
         DC X'4B4B7A4B4B7D7E7F'          (X'78' TO X'7F')
         DC X'4B81828384858687'          (X'80' TO X'87')
         DC X'88894B4B4B4B4B4B'          (X'88' TO X'8F')
         DC X'4B91929394959697'          (X'90' TO X'97')
         DC X'98994B4B4B4B4B4B'          (X'98' TO X'9F')
         DC X'4BA1A2A3A4A5A6A7'          (X'A0' TO X'A7')
         DC X'A8A94B4B4B4B4B4B'          (X'A8' TO X'AF')
         DC X'5F44B4BB4B4B4B4B'          (X'B0' TO X'B7')
         DC X'4B4B4B4B4B4B4B4B'          (X'B8' TO X'BF')
         DC X'4BC1C2C3C4C5C6C7'          (X'C0' TO X'C7')
         DC X'C8C94B4B4B4B4BC0'          (X'C8' TO X'CF')
         DC X'4BD1D2D3D4D5D6D7'          (X'D0' TO X'D7')
         DC X'D8D94B4B4B4B4B4B'          (X'D8' TO X'DF')
         DC X'E04BE2E3E4E5E6E7'          (X'E0' TO X'E7')
         DC X'E8E94B4B4B4B4B7B'          (X'E8' TO X'EF')
         DC X'F0F1F2F3F4F5F6F7'          (X'F0' TO X'F7')
         DC X'F8F94B4B4B4B4B'            (X'F8' TO X'FE')
*
*    TTC1FB2 - CCSID 00037 FROM CCSID 04371 Table
*
TTC1FB2  DS   0XL191
         DC X'4041424344454647'          (X'40' TO X'47')
         DC X'4849714B4C4D4E5A'          (X'48' TO X'4F')
         DC X'5051525354555657'          (X'50' TO X'57')
         DC X'58595B685C5D5EB0'          (X'58' TO X'5F')
         DC X'6061626364656667'          (X'60' TO X'67')
         DC X'6869486B6C6D6E6F'          (X'68' TO X'6F')
         DC X'7071727374757677'          (X'70' TO X'77')
```

```
DC X'78467AEF667D7E7F'          (X'78' TO X'7F')
DC X'8081828384858687'          (X'80' TO X'87')
DC X'88898A8B8C8D8E8F'          (X'88' TO X'8F')
DC X'9091929394959697'          (X'90' TO X'97')
DC X'98999A9B9C9D9E9F'          (X'98' TO X'9F')
DC X'A0A1A2A3A4A5A6A7'          (X'A0' TO X'A7')
DC X'A8A9AAABACADAEAF'          (X'A8' TO X'AF')
DC X'B0B1B2B3B4B5B6B7'          (X'B0' TO X'B7')
DC X'B8B9BABBBCBDBEBF'          (X'B8' TO X'BF')
DC X'CFC1C2C3C4C5C6C7'          (X'C0' TO X'C7')
DC X'C8C9CACBCCCDCECF'          (X'C8' TO X'CF')
DC X'51D1D2D3D4D5D6D7'          (X'D0' TO X'D7')
DC X'D8D9DADBDCDDDEDF'          (X'D8' TO X'DF')
DC X'E0E1E2E3E4E5E6E7'          (X'E0' TO X'E7')
DC X'E8E9EAEBECEDEEEF'          (X'E8' TO X'EF')
DC X'F0F1F2F3F4F5F6F7'          (X'F0' TO X'F7')
DC X'F8F9FAFBFCFDFE'            (X'F8' TO X'FE')
END
```

Displaying square brackets used in C programs

The standard non-APL terminals that ISPF supports do not have the left and right brackets used in a C program. Therefore, the translation tables provided with ISPF are defined so that these characters are not valid and are displayed as periods.

If you have a terminal that supports these characters, you can modify the translation tables TTVAL and TTGSM described earlier. To do so, simply indicate that these characters are valid.

Note that the C/370 compiler expects the brackets at codepoints AD and BD. On an APL or TEXT terminal, if you use the ISPF-supplied terminal type 3278A, the codepoints AD and BD are displayed correctly as left and right brackets.

ISPEXEC processing

ISPEXEC is an external entry point in module ISPLINK. This is how ISPF supports the “Call ISPEXEC” interface in module dialogs. As a result of executing a CLIST that is not under ISPF, if the CLIST contains ISPEXEC dialog service statements, CLIST might try to invoke the ISPEXEC module as a command processor. The results of this change are as follows:

- If the ISPLINK module (or its alias entry points ISPLNK, ISPEXEC, ISPEX, or ISPQRY) is not invoked under ISPF, TSO issues an error message with a return code of 20.
- The ISPEXEC entry point can interfere with your installation’s setup if someone creates a CLIST called ISPEXEC, or if a CLIST specifically checks for a return code of 12 (TSO issues return code 12 if you try to invoke ISPEXEC when not under ISPF). To eliminate this problem, you make the following 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 the following SUPERZAP:

```
NAME      ISPLINK    ISPLINK
VER       00FC       0A5D       TPUT SVC
REP       00FC       0700       NOP INSTRUCTION
```

Remember that the location of the `0A5D` instruction can change from `00FC` as a result of maintenance.

ISPF-to-APL2 terminal type mappings

This section provides information about how to add or change mappings, if you need to do so. Refer to *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 the above 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, do the following:

1. Enter `apl2`
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 "The ISPF Configuration Table" on page 7 for more information.

ISPF installation-wide exits

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 23 summarizes the ISPF installation-wide exits. The exit ID is a unique number that identifies that installation-wide exit to ISPF.

Table 23. ISPF Installation-Wide Exits

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

Table 23. ISPF Installation-Wide Exits (continued)

Exit ID	Installation Exit Name	Possible Uses for the Routine
13	ISPF/PDF service end	Marks the termination of ISPF or PDF dialog services invoked through the ISPLINK or ISPEXEC interfaces.
14	SWAP logical screens	Indicates a change of the active logical screen. Together with the logical screen start and end installation-wide exits, the routine can monitor resource use for each ISPF logical screen.
15	DISPLAY service start	Provides for tailoring of panels to be displayed.
16	Log, list, and temporary data set allocation	Controls data set naming conventions for log, list, and temporary data sets.

The ISPF installation-wide exits are linked together in the ISPEXITS load module. The main entry point of ISPEXITS is ISPXDT. ISPXDT defines which of the 16 installation-wide exits you plan to use and the names of the exit routines that receive control at each user exit. To assist you in building ISPXDT, ISPF provides seven assembler macros. These macros are described in "Exit Macros" on page 134.

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 in this manual.
- You can define a data area to be used by each exit routine. These data areas can be shared by different exit routines.
- At initialization, ISPF gets the storage for the data area. It provides the length and the address of the data area in the parameter list when calling an exit routine that uses that data area. (ISPF obtains the storage from subpool 0 below the 16 MB line.) The first time ISPF calls the exit routine, the data area contains binary zeros. The information in the data area is retained between invocations of the exit routines and the storage is available for the entire ISPF session (independent of logical screen abends and restarts). It is released at ISPF termination.
- Data areas are on double-word boundaries.

How to Install the Installation-Wide Exits

To install an ISPF installation-wide exit:

1. Indicate in installation tailoring that ISPF installation-wide exits are defined. When you install ISPF, you indicate that exit routines might exist by setting the ENABLE_ISPF_EXITS keyword in the ISPF Configuration table to YES. As a result, you can replace, add, or remove exit routines by simply reassembling ISPXDT and relinking ISPEXITS. Otherwise, ISPF uses the default value of NO and does not load the ISPEXITS module. In this case, you cannot invoke any exit routines and you have to repeat part of the installation process if you want to change this value. See "The ISPF Configuration Table" on page 7 for more information.
2. Use the exit macros to assemble the exit definition table (ISPXDT). This process is discussed in detail in "Exit Macros." These macros reside in ISP.SISPMACS.
3. Link-edit the ISPEXITS load module. ISPXDT is the main entry point of this module. There is no need to include any other ISPF-supplied CSECTS in ISPEXITS. Figure 51 contains sample JCL to link-edit ISPEXITS.

```
//LKED EXEC PGM=IEWL,REGION=1024K,
// PARM='XREF,LET,LIST,RENT,SIZE=(512K,128K)'
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD UNIT=SYSDA,SPACE=(TRK,(10,5))
//SYSLMOD DD DISP=SHR,DSN=ISP.SISPLOAD
//* All exit routines and ISPXDT CSECT must be in SYSLIB,
//* but an INCLUDE SYSLIB statement is only required for
//* ISPXDT and not for the exit routines.
//SYSLIB DD DSN=ISP.LOCOBJ,DISP=SHR
//SYSLIN DD *
   ORDER ISPXDT
   ENTRY ISPXDT
   INCLUDE SYSLIB(ISPXDT)
 NAME ISPEXITS(R)
/*
//
```

Figure 51. 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.

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.

```
ISPMXED          {START|END}
```

ISPMXLST Defines the installation-wide exits where you provide exit routines.

```
ISPMXLST         [(epcode1[,epcode2]...)]
```

where:

epcode The numeric code for an user exit provided by ISPF. The list of codes must be enclosed in parentheses, and must be in ascending order. See Table 23 on page 132 for a list of the numeric codes.

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

```
ISPMXDEF         epcode
```

where:

epcode The numeric code for an exit point provided by ISPF. You must include this code as an operand for the ISPMXLST macro in order for the user exit to be defined. See Table 23 on page 132 for a list of the numeric codes.

ISPMEPT Identifies an exit routine to call at a particular user exit.

```
ISPMEPT          entryname[,data-area-name]
```

where:

entryname Identifies the entry point of the exit routine. This is usually a CSECT name.

data-area-name Identifies the name of the data area the exit routine uses. The name can be up to 8 characters long. If you list the data area on an ISPMEPT macro, you must define it using the 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.

```
ISPMXEND
```

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.

```
ISPMXDD          {START|END}
```

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

```
ISPMDAD          data-area-name,size
```

where:

data-area-name Identifies the name of the data area an exit routine uses. The name can be up to 8 characters long.

size Specifies, in bytes, the size of the data area. If the size is not a multiple of eight, ISPF rounds it up to the next multiple of eight.

How to Use the Macros to Define ISPXDT

To define ISPXDT using the exit macros:

1. Indicate the beginning of the exit entry definition section of ISPXDT by coding one ISPMXED macro with the START operand.

   Example

   ```
   ISPMXED START
   ⋮
   ```

2. Define the installation-wide exits you plan to use. Code the ISPMXLST macro, listing the numeric codes for each user exit in ascending order, enclosed within parentheses. The following example shows that exit routines will be installed at ISPF installation-wide exits 3, 7, and 14.

   Example

   ```
   ISPMXED START
   ISPMXLST (3,7,14)
   ⋮
   ```

   If no operand is specified, a dummy ISPXDT is built.

3. For each user exit that you define on the ISPMXLST macro, use the ISPMXDEF, ISPMEPT, and ISPMXEND macros to define one or more exit routines. These three macros are a set and must be coded for each user exit you identify.
   - Code the ISPMXDEF macro with the exit ID as its operand.
   - Code the ISPMEPT macro for each exit routine and, if required, its data areas.
   - Code the ISPMXEND macro to end the exit routine definitions for this user exit.

   In the following example, three exit routines are defined at user exit 7. The first two share a common data area (DAREA015) and the third uses a different data area (DAREA027). ISPF calls entry points EXPT7EP1, EXPT7EP2, and EXPT7EP3 in that order unless one of the routines returns a return code of 12 or greater. For example, if entry point EXPT7EP2 returns a return code of 12, control returns to ISPF without executing entry point EXPT7EP3. Therefore, if you define more than one exit routine at a given user exit, define them in the correct order of priority.

Example

```
ISPMXED START
ISPMXLST (3,7,14)

.
.
.

ISPMXDEF 7
ISPMEPT EXPT7EP1,DAREA015
ISPMEPT EXPT7EP2,DAREA015
ISPMEPT EXPT7EP3,DAREA027
ISPMXEND

.
.
.
```

If you use the ISPMXDEF macro to define an exit ID that is not listed on the ISPMXLST macro, ISPF issues a warning message. You can still use ISPXDT, but the user exit is defined. As a result, you can easily disable a user exit by deleting its exit ID on the ISPMXLST macro. However, for every exit point listed on the ISPMXLST macro, there must be a corresponding ISPMXDEF macro.

4. Indicate the end of the exit entry definition section of ISPXDT by coding the END operand on the ISPMXED macro.

Example

```
ISPMXED START
ISPMXLST (3,7,14)

.
.
.

ISPMXED END
```

5. Define the data areas used by the exit routines in the exit data area definition section of ISPXDT. Indicate the start of this section by using the START operand on the ISPMXDD macro.

Example

```
.
.
.

ISPMXED END
ISPMXDD START

.
.
.
```

6. Use the ISPMDAD macro to define the name and size of each data area defined on an ISPMEPT macro. Every data area referenced on an ISPMEPT macro must be defined by an ISPMDAD macro.

   The following example defines DAREA015 to be 100 bytes long. (ISPF rounds the length to 104 in this case.)

Example

```
  ⋮
  ISPMXDD START
  ISPMDAD DAREA015,100
  ⋮
```

If you use the ISPMDAD macro to define a data area that is not coded on an ISPMEPT macro, a warning message is issued during assembly and storage for the area is not obtained during execution. If none of your exit routines require a data area, you need not code an ISPMDAD macro, but you must still code ISPMXDD START and ISPMXDD END.

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

Example

```
  ⋮
  ISPMXDD START
  ISPMDAD DAREA015,100
  ⋮
  ISPMXDD END
```

Sample ISPXDT Definition

Figure 52 on page 139 shows a sample ISPXDT definition that defines exit routines for installation-wide exits 2 and 12.

```
<<ISPMXED<START
<*
<<ISPMXLST<(2,12)
<*
<<ISPMXDEF<2
<<ISPMEPT<MYEXT021,MYAREA01
<<ISPMEPT<MYEXT022,MYAREA02
<<ISPMEPT<MYEXT023,MYAREA02
<<ISPMXEND
<*
<<ISPMXDEF<12
<<ISPMEPT<MYEXT121,MYAREA03
<<ISPMEPT<MYEXT122,MYAREA01
<<ISPMEPT<MYEXT123
<<ISPMEPT<MYEXT124,MYAREA02
<<ISPMXEND
<*
<<ISPMXED<END
<*
<<ISPMXDD<START
<*
<<ISPMDAD<MYAREA01,1024
<<ISPMDAD<MYAREA02,2048
<<ISPMDAD<MYAREA03,256
<*
<<ISPMXDD<END
```

Figure 52. Sample ISPXDT Definition

At user exit 2, exit routines with entry points MYEXT021, MYEXT022, and MYEXT023 are called, in that order.

At user exit 12, exit routines with entry points MYEXT121, MYEXT122, MYEXT123, and MYEXT124 are called, in that order.

MYAREA01 (1024 bytes) is used by exit routines at MYEXT021 and MYEXT122.

MYAREA02 (2048 bytes) is used by exit routines at MYEXT022, MYEXT023, and MYEXT124.

MYAREA03 (256 bytes) is used by the exit routine at MYEXT121.

Exit routine MYEXT123 does not require a data area.

Exit Parameter List

Each of the ISPF exit routines is passed a parameter list that contains parameters common to all the installation-wide exits and parameters that are specific to the user exit. The parameters are in the order shown here, followed by exit-specific parameters in the order shown under the individual user exit description. The common parameters are:

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 the following 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.

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 139, the exit routines at this user exit receive the following 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 139, the exit routines at this user exit receive the following 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 the following:

- 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 139, the exit routines at this user exit receive the following 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 139, the exit routines at this user exit receive the following parameters. ISPF ignores any changes the exit routines make to the parameters.

Flags 4 bytes of bit flags defined as follows:

0 1 = PGM keyword specified

1 1 = CMD keyword specified

2 1 = PANEL keyword specified

3–4 Reserved

5 1 = ADDPOP keyword specified

6 1 = BARRIER keyword specified

7 1 = NEST keyword specified

8 1 = new application being invoked

9 1 = new shared pool is to be created

10 1 = PASSLIB keyword specified

11 1 = Lang (CREX) keyword specified

12 1 = APL keyword specified

13 1 = MODE(FSCR) specified

14 1 = MODE(LINE) specified

15–16 Reserved

17 1 = SUSPEND specified

18 1 = LOGO keyword specified

19–31 Reserved.

Note: If no parameters were passed, the flag bytes will be zero.

elemname
An 8-character field that contains the name of the element (PGM, CMD, or PANEL) to be selected. In case of CMD, if the element is a CLIST prefixed with %, the % symbol is removed from the elemname.

APPLID
A 4-character field that contains the application ID of the element that just terminated.

STATS
Two contiguous fullwords containing:

Total "think time" in hundredths of a second for this SELECT level only. "Think time" means all of the time intervals between the time when ISPF unlocked the keyboard for input and the time when input was received by ISPF. If there are other SELECTs nested within this one, their statistics are given at the corresponding select end exits, but not included here.

The number of times the screen was read by ISPF in this SELECT level.

logoname
An 8-character field that contains the name of the LOGO panel if it was specified.

Screen name
Screen name as set by the SCRNAME command and shown by the SCRNAME ON function.

Return Codes

Programming Interface information

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

Programming Interface information

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

Programming Interface information

In addition to the standard exit parameters described in "Exit Parameter List" on page 139, the exit routines at this user exit receive the following 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 155 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 155 and "ISPTCM Usage Notes" on page 158 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

Programming Interface information

0 Normal completion; ISPF continues processing.

4 The exit routine has changed the value of the TCM flag byte. In this case, ISPF uses the value of the flag to process the command. However, changes to the TCM flag bit indicating whether the command should be logged are ignored.

16 Authorization failure. The exit routine indicated that this command should not be attached. ISPF issues an error message indicating that the exit routine rejected the command and returns to the caller.

Other ISPF treats the error as severe and issues a message indicating that the exit routine returned an incorrect return code.

Exit 6: TSO Command End Exit

Programming Interface information

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

Programming Interface information

In addition to the standard exit parameters described in "Exit Parameter List" on page 139, the exit routines at this user exit receive the following 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 155 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

Programming Interface information

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

Programming Interface information

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

Programming Interface information

In addition to the standard exit parameters described in "Exit Parameter List" on page 139, the exit routines at this user exit receive the following 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

Programming Interface information

0 Normal completion; ISPF continues processing.

16 Authorization failure. ISPF issues a message indicating an authorization failure has occurred and LIBDEF terminates with a severe error.

Other LIBDEF treats return codes other than 0 or 16 as a severe error and returns a code of 20 to the caller.

Exit 8: RESERVE Exit

Programming Interface information

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 247 for information about ISPF’s use of ENQ and RESERVE. ISPF gives control to the exit routine before it does the RESERVE. If your serialization mechanism cannot acquire the resource, or if the exit routine returns a code greater than 0, ISPF does not attempt to get it.

Exit Parameters

Programming Interface information

In addition to the standard exit parameters described in "Exit Parameter List" on page 139, the exit routines at this user exit receive the following parameters. ISPF ignores any changes made to these parameters.

qname
An 8-character field that contains the queue name against which the RESERVE is done. (ISPF uses a qname of 'SPFEDIT '.)

rname The 44-character resource name (data set name) that is reserved.

ucbaddr
A fullword field containing the address of the unit control block (UCB) for the reserved device.

Flags 4 bytes of bit flags defined as follows:

0 1 = Exclusive reserve requested. 0 = Shared reserve requested. (ISPF always requests exclusive use of the resource at RESERVE.)

1 1 = A RESERVE or ENQ will be done on qname 'SYSIEWLP' (the linkage editor qname).

2 1 = The resource being reserved is a partitioned data set.

3 1 = The resource being reserved is a sequential data set.

4–31 Reserved.

Note: The exit routine should not use the same (qname, rname) combination (in a RESERVE macro call) that ISPF uses. If that happens, an abend occurs when ISPF attempts to do its own RESERVE.

Return Codes

Programming Interface information

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

Programming Interface information

Use this user exit to release any resources acquired at the RESERVE user exit. ISPF relies on task termination to release any resources reserved by the abending logical screen task. This user exit does not get control in the case of abnormal termination of the logical screen.

If your serialization mechanism is such that task termination will not release the reserved resources, an exit routine is provided at the logical screen end user exit to clean up any unreleased resources for that task.

The RESERVE done by ISPF is released before the exit routine is given control.

Exit Parameters

Programming Interface information

In addition to the standard exit parameters described in "Exit Parameter List" on page 139, the exit routines at this user exit receive the following 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

Programming Interface information

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

Programming Interface information

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

Programming Interface information

This user exit uses only the standard exit parameters described in "Exit Parameter List" on page 139, which includes the logical screen identifier (screenid).

Return Codes

Programming Interface information

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

Programming Interface information

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, the following 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

Programming Interface information

In addition to the standard exit parameters described in "Exit Parameter List" on page 139, the exit routines at this user exit receive the following parameters:

Flags 4 bytes of bit flags defined as follows:

0 0 = Normal termination 1 = Abnormal termination.

1-31 Reserved

Next logical screen
An 8-character field that identifies the next active ISPF logical screen. In the case of exiting ISPF, the next logical screen ID will be 0.

Return Codes

Programming Interface information

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

Programming Interface information

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

Programming Interface information

In addition to the standard exit parameters described in "Exit Parameter List" on page 139, the exit routines at this user exit receive the following 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

Programming Interface information

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

Programming Interface information

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

Programming Interface information

In addition to the standard exit parameters described in "Exit Parameter List" on page 139, the exit routines at this user exit receive the following 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

Programming Interface information

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

Programming Interface information

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

Programming Interface information

This user exit uses only the standard exit parameters described in "Exit Parameter List" on page 139, which includes the logical screen identifier (screenid). The screen ID identifies the logical screen activated as a result of the SWAP command.

Return Codes

Programming Interface information

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

Programming Interface information

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

Programming Interface information

In addition to the standard exit parameters described in "Exit Parameter List" on page 139, the exit routines at this user exit receive the following 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

Programming Interface information

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, and Temporary Data Set Allocation Exit

Programming Interface information

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 the following 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:
   uptpfx.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 following form:
   uptpfx.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

Programming Interface information

In addition to the standard exit parameters described in “Exit Parameter List” on page 139, the exit routines at this user exit receive the following 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

5–31 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

Programming Interface information

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.

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:

```
[symbol] ISPMTCM   HEADER  [,NOUSRFLG=number]
                           [,DFUSRFLG=number]
                           [,DFFLAG=flag]
                           [,DFCLRLNS=number]


                   ENTRY    ENTNAME=entry name
                           [,FLAG=flag]
                           [,USRFLG=user flag]
                           [,CLRLNS=number]


                   END
```

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.
B'..1.....'	Command requires authorization check. Set this bit on for a command processor that must be invoked as an authorized command.
B'...1....'	Command is not to be logged. Set this bit on if the TSO command buffer should not be written to the ISPLOG data set.
B'....1...'	Command is not supported by ISPF. Set this bit on for commands that cannot be invoked under ISPF.
B'.....1..'	Command is command procedure (CLIST). Set this bit on if this is the name of a CLIST member.
B'......1.'	Command is a command processor. Set this bit on if this is the name of a command processor program module.
B'.......1'	Command requires a BLDL to be issued. Set this bit on if a BLDL is to be issued to determine whether this is a command processor module or a CLIST.

USRFLG
The value of the user flag for the current entry. Specify this parameter only if NOUSRFLG is not 0. If you do not specify a value and NOUSRFLG is greater than 0, ISPF uses the default user flag (DFUSRFLG). If specified, USRFLG must be a hexadecimal string with a length equal to the value of NOUSRFLG.

CLRLNS
The minimum number of lines to clear if line mode is entered for this entry. The value should be an integer from 0 to 99. Specifying a value of 0 causes the entire physical screen to be erased. If you do not specify a value for CLRLNS, the DFCLRLNS value in the type HEADER call is used.

When the value for the number of lines to clear is nonzero, ISPF determines where to clear the screen according to the following:

1. ISPF calculates a value = (number of lines in the visible portion of the active logical screen - 1) - CLRLNS
2. The lesser of the above calculated value and the number of lines in the panel displayed on the active logical screen (but not less than 0) is the number of the line after which the screen is cleared. Thus, ISPF will clear more than the CLRLNS number of lines if it can do so without overlaying the displayed panel.

Notes:

1. If a CLRLNS value is larger than the visible portion of the active logical screen, ISPF erases the screen beginning at the top of the logical screen.
2. In split-screen mode, if line mode is entered when the top screen is active, the bottom screen will always be cleared. If the bottom screen is active, the bottom screen will be cleared (even if CLRLNS is greater than the physical screen size).
3. For DBCS devices, ISPF always erases the screen beginning at the top of the current logical screen, regardless of the CLRLNS value.
4. For 3290 devices, the entire physical screen is always cleared before going into line mode.

END Must be last macro call.

Sample ISPTCM Definition

An example of how to use the ISPMTCM macro to build ISPTCM is shown in Figure 53.

```
ISPMTCM  HEADER,DFCLRLNS=5,NOUSRFLG=2,DFFLAG=61

ISPMTCM  ENTRY,ENTNAME=ALLOCATE,CLRLNS=4

ISPMTCM  ENTNAME=MYCMDA,USRFLG=FFFF
ISPMTCM  ENTNAME=MYCMD1,CLRLNS=10,FLAG=14

ISPMTCM  END
```

Figure 53. Sample ISPTCM Definition

ISPTCM Usage Notes

- ISPTCM is a load module that contains a list of command names and their characteristics. ISPF processes each command in ISPTCM according to the FLAG field defined for its entry. If a command is not in ISPTCM, the DFFLAG parameter is used. The default value of DFFLAG is 61, which indicates to ISPF that commands not contained in ISPTCM require a function pool, an authorization check, and must be logged. Also, a BLDL should be done to locate the command. If the BLDL cannot locate the command, ISPF assumes it to be CLIST and attaches the EXEC command processor.

  You can alter DFFLAG to suit the needs of your installation. If you have not changed DFFLAG, or changed it so that it still calls for a BLDL, if a command processor is to run from the link pack area, its name *must* be in ISPTCM.
- Certain commands, such as, LOGON and ISPF are invalid under ISPF. Do not attempt to make these entries valid by changing the FLAG. The results are unpredictable.

- The USRFLG is an optional field for an entry in ISPTCM. If you define exit routines for TSO command start or TSO command end user exits, or both, you can define USRFLG according to your installation's needs. If you do, ISPF passes these flags to the exit routines. These flags do not affect ISPF execution. See "ISPF installation-wide exits" on page 132 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 the following 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 54.

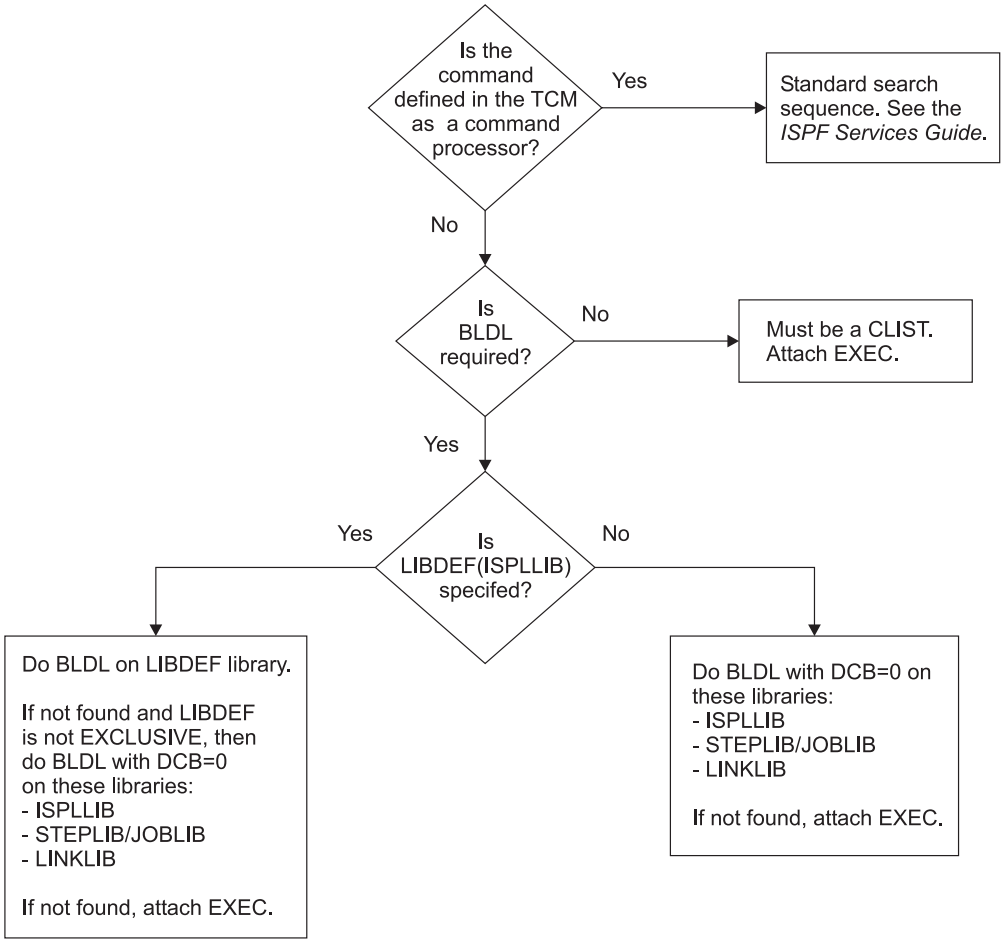


Figure 54. Search Sequence for Attaching Commands

Alternate option 7.1 panels

Panel ISPYFP, the normal panel displayed with option 7.1, requires extra scrolling to display some of the test entry fields and options. Because this can be inconvenient, ISPF provides alternate panels for ISPYFP. The alternate panels present all of the same entry fields and options in revised formats.

Panel ISPYFPA places most of the commonly used information within the first 24 panel lines. This format often eliminates the need for panel scrolling.

Panel ISPYFPB is similar to panel ISPYFPA, but it has a selection field that enables the user to select a function: panel, command, program, or request. Unlike panels ISPYFP or ISPYFPA, on panel ISPYFPB the panel, command, program, or request fields can all contain values.

All three panels are shipped as part of the product panel library. To select which panel to use, set the ISPF Configuration table keyword USE_ALTERNATE_DIALOG_TEST_PANEL to one of the following options:

1 ISPYFP panel

2 ISPYFPA panel

3 ISPYFPB panel

ISPF National Language Support (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. Refer to the *z/OS Program Directory* for installation instructions.
2. Optionally, set up a default language for your installation.

   With the shipped version of ISPF, the default session language is English. You must invoke ISPF (ISPSTART) with a language keyword to get the session in a different language. If ISPF is used at your installation primarily in a non-English language, it is recommended that you change the default language. The benefits of doing this are:
   - Not having to enter the language keyword when invoking ISPSTART
   - Improved initialization time
   - Possibly smaller LPA use
   - Initialization error messages are issued in the default language.
   - See “Changing the Session Language Default Value” for further details.
3. Set up the execution environment.

   To run ISPF in any session language, perform the following 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 page 60). For information about changing Configuration Table settings, see “The ISPF Configuration Table” on page 7.

Chapter 4. Customizing PDF

This chapter 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 on page 7. 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*.

The following topics describe how to modify the distributed release of PDF to suit the particular needs of your installation:

Edit Mode Defaults
Data Set Allocation Defaults for the Outlist Utility
Using the Hardcopy Utility with DBCS Support
SCLM Batch Considerations
Converting From LMF To SCLM
Creating PDF Translation Tables
PDF Foreground and Batch Customizing
Customizing Browse and Edit
Customizing Member List Panels
IBM Products Option
PDF Installation-Wide Exits

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 the following 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, AUTOSAVE and IMACRO in existing profiles. When a setting is forced the editor WILL CHANGE the users' profiles, so be very careful if you override the IMACRO setting. IBM recommends that you use the site-wide initial macro instead of forcing the initial macro in each user's profile.

It is helpful to understand when the ZDEFAULT profile is used and where it exists in a user's concatenations. The ZDEFAULT profile exists as a row of the edit profile table named *xxx*EDIT 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 "The ISPF Configuration Table" on page 7 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 76.

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 39 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 (*aaaa*EDRT, 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.

Note: When entering 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.

The following restrictions apply to edit recovery data sets:
- they must be allocated as sequential data sets of record format U
- they cannot be striped, or striped and compressed data sets
- they cannot be multivolume data sets
- they should not be allocated on an SMS-managed volume whose management class has the PARTIAL_RELEASE attribute

Edit recovery will not delete edit recovery data sets listed in the edit recovery table when the table has a disposition field set to 'K' associated with the recovery data set name. This is an obsolete facility for preallocating edit recovery files or enforcing naming conventions for edit recovery files. Instead, use the data set name change exit for this purpose.

Data Set Allocation Defaults for the Outlist Utility

The PDF 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 the following 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
           <action run=ISRROUTE parm=U8>
to
       <pdc unavail=zut7 acc1=alt acc2=8>Outlist
           <action run=ISRUOL type=CMD>
```

3. Use Option 3.9 to create a command table entry called ISRUOL, specifying this as the action:

```
        SELECT CMD(ISRUOL)
```

The Outlist utility invokes the TSO OUTPUT command to retrieve data from the SYSOUT queue. For the browse and print options of this utility, ISPF allocates a print data set and passes its name to the OUTPUT command by means of the PRINT parameter. The parameters in CLIST ISRUOL or the ISPF configuration table for program ISRUOLP determine the attributes of the printdata set.

You can modify the following print data set attributes to meet the needs of your installation. The default value for each attribute is shown in parentheses:

```
LRECL (133)
BLKSIZE (3059)
Primary space in tracks (200)
Secondary space in tracks (100)
```

To change the attributes for CLIST ISRUOL, find the following lines and edit them as follows:

```
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 the following 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 the following 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 the following 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. The FLMLIBS skeleton is found in your ISPF skeleton target data set ISP.SISPSLIB. FLMLIBS is the common imbed for the other SCLM skeletons used for batch submission. Data set names in member FLMLIBS need to be modified to match your installation's naming conventions.

Notes:

1. In Figure 58 on page 167, the 'ISP' data set high-level qualifier represents your ISPF data sets.
2. *xxx* corresponds to a national language as follows:

Language	**xxx**
US English	ENU (the default)
Uppercase English	ENP
Swiss German	DES
Japanese	JPN
German	DEU

```
)CM
)CM THIS DEFINES THE STEPLIB AND ISPF LIBRARIES
)CM TO BE USED DURING SCLM BATCH OPERATIONS
)CM
)CM BE SURE TO INCLUDE THE LOAD LIBRARIES CONTAINING ISPF.
//*
//*****************************************************************
//* STEPLIB LIBRARIES
//*****************************************************************
//*
//STEPLIB  DD  DSN=ISP.SISPLPA,DISP=SHR
//         DD  DSN=ISP.SISPLOAD,DISP=SHR
//         DD  DSN=ISP.SISPSASC,DISP=SHR
//*
//*****************************************************************
//* ISPF LIBRARIES
//*****************************************************************
//*
//ISPMLIB  DD  DSN=ISP.SISPMXXX,DISP=SHR ISPF MSGS
//*
//ISPSLIB  DD  DSN=ISP.SISPSXXX,DISP=SHR ISPF SKELS
//         DD  DSN=ISP.SISPSLIB,DISP=SHR ISPF SKELS
//*
//ISPPLIB  DD  DSN=ISP.SISPPXXX,DISP=SHR ISPF PANELS
//*
//ISPTLIB  DD  UNIT=&VIOUNIT;,DISP=(NEW,PASS),SPACE=(CYL,(1,1,5)),
//             DCB=(LRECL=80,BLKSIZE=19040,DSORG=PO,RECFM=FB),
//             DSN=&TABLESP                  TEMPORARY TABLE LIBRARY
//         DD  DSN=ISP.SISPTXXX,DISP=SHR ISPF TABLES
//*
//ISPTABL  DD  UNIT=&VIOUNIT;,DISP=(NEW,PASS),SPACE=(CYL,(1,1,5)),
//             DCB=(LRECL=80,BLKSIZE=19040,DSORG=PO,RECFM=FB),
//             DSN=&TABLESP                  TEMPORARY TABLE LIBRARY
//*
//ISPPROF  DD  UNIT=&VIOUNIT;,DISP=(NEW,PASS),SPACE=(CYL,(1,1,5)),
//             DCB=(LRECL=80,BLKSIZE=19040,DSORG=PO,RECFM=FB),
//             DSN=&TABLESP                  TEMPORARY TABLE LIBRARY
//*
//ISPLOG   DD  SYSOUT=*,
//             DCB=(LRECL=120,BLKSIZE=2400,DSORG=PS,RECFM=FB)
//*
//ISPCTL1  DD DISP=NEW,UNIT=VIO,SPACE=(CYL,(1,1)),
//            DCB=(LRECL=80,BLKSIZE=800,RECFM=FB)   TEMPORARY FILE
//*                                                 TAILORING DATASET
//*                                                          OW01230
//SYSTERM  DD SYSOUT=*
//*
//*---------------------------------------------------------------------
//* TEMPORARY CLIST CONTAINING COMMAND TO BE EXECUTED
//*---------------------------------------------------------------------
//SYSPROC  DD  DSN=&&&&CLIST&STEP,DISP=(OLD,DELETE)
//         DD  DSN=ISP.SISPCLIB,DISP=SHR          CLIST LIBRARY OW01230
//*
)CM
)CM 5655-042 (C) COPYRIGHT IBM CORP 1989, 1995 */
```

Figure 58. Sample FLMLIBS Skeleton

Converting From LMF To SCLM

The LMF function was removed from ISPF in z/OS V1R2.0. Users of LMF are encouraged to migrate their existing LMF-controlled libraries to SCLM. There are several differences between LMF and SCLM, but the basic library manager function is similar, so migrating the hierarchy from LMF to SCLM is straightforward. SCLM has a full set of services, so most dialogs that use LMF services can be converted easily to SCLM. SCLM also includes a complete set of Configuration Management options.

General Conversion

Before doing the conversion, you should review the *z/OS ISPF Software Configuration and Library Manager Project Manager's and Developer's Guide*, particularly Chapter 1 (Defining the Project Environment) and Chapter 4 (Converting Projects to SCLM). Chapter 1 describes the steps for setting up a project definition. Chapter 4 shows the steps for working with existing project data sets.

SCLM and LMF are similar in function, but there are some basic differences to consider when converting a project:

- In SCLM, all data sets in the hierarchy are controlled. This includes the data sets where the developers do their editing (called the *development level* in SCLM). These data sets are not controlled in the LMF environment.
- SCLM does not use a started task to update the controlled libraries. Users must have UPDATE authority to any level of the hierarchy that they must change, including the development level for editing, and any level into which they need to promote.
- SCLM uses a VSAM data set to hold its control information. Control information includes information about where members exist in the hierarchy, statistics, and dependency information.
- SCLM automates the BUILD step (compile, assemble, link-edit, and so on) and requires it to be done. This means SCLM must know what language is assigned to a part, and how the language is to be processed at BUILD time (called "SCLM Language Definition").
- Output data, such as compiled object and load modules, can be automatically generated by SCLM. Review your processes to determine how to take advantage of SCLM's synchronization of source modules and assigned outputs.

To convert from LMF to SCLM, follow the steps listed in Chapter 1 of the *z/OS ISPF Software Configuration and Library Manager Project Manager's and Developer's Guide*. This takes you through the creation of the SCLM project definition and control data sets needed for SCLM.

The steps involved in defining the project environment are:

1. Determine the project hierarchy. For an existing project the hierarchy has already been defined, but a review is recommended.
2. Identify the types of data to support. Again, this has already been done with an existing project.
3. Establish authorization codes. Authorization codes control the movement of data within the hierarchy. The purpose of this step is to assign authorization codes to the hierarchy. Authorization codes restrict the drawdown and promotion of members to certain groups within the hierarchy. The concept is

similar to LMF's concepts of authorized promoters and member access control, except that LMF uses userids and SCLM uses group names to restrict drawdown and promotion.

4. Allocate the PROJDEFS data sets. The PROJDEFS data sets are used to store the project definition data for an individual project. The project definition contains some of the same types of information as the LMF control file, such as the structure of the hierarchy.
5. Allocate the project partitioned data sets. Because you are using an existing project, most of these data sets already exist. Since all data sets are controlled in the SCLM environment, you will have to create development level data sets as part of the hierarchy for the developers to use when editing data.
6. Allocate and create the control data sets. Control data sets are used to track and control application programs within the hierarchy. There are 5 types of VSAM data sets that can be associated with a project:

   Primary Accounting
   The accounting data sets contain information about the software components in the project including statistics, dependency information and build maps (information about the last build of a member). At least one accounting data set is required for a project. The accounting data set contains some of the same types of information as the LMF control file, such as the member locking data.

   Secondary Accounting
   This is an optional backup of the information in the primary accounting data set.

   Export Accounting
   The export accounting data set contains accounting information that has been exported from the accounting data set. It is only required if the Import/Export functions of SCLM are used.

   Primary Audit Control
   The audit control data set contains audit information about changes to the software components in the project for groups that have auditing turned on. The audit data set contains the same types of information as the LMF activity log, as well as versions of members, if requested.

   Secondary Audit Control
   This is an optional backup of the information in the primary Audit Control data set.
7. Protect the project environment. This step should be reviewed because SCLM is structured slightly differently than LMF. Remember that developers must have UPDATE authority to the levels they need to change.
8. Create the project definition. The project definition is an assembler program, created using SCLM macros. Much of the definition is similar to the information that was entered in LMF's Library Controls (Option 8.1).
9. Assemble and link the project definition.

After the project has been defined, follow the steps in the *z/OS ISPF Software Configuration and Library Manager Project Manager's and Developer's Guide* to put your existing project under SCLM control. In general, the steps are:

- Prepare the hierarchy. Delete all unnecessary data from the libraries. If any groups are to be non-key, make sure they don't contain any data.
- Create alternate project definitions.
- Create architecture definitions for the project.

- Register existing members with SCLM.

Converting LMF Dialogs to SCLM

There were eight ISPF services relating to LMF. Some of these have SCLM equivalents, some do not because of the differences between SCLM and LMF. The eight services are:

LMPROM
Use the SCLM PROMOTE service.

EDIT with the LOCK parameter
Use the SCLM EDIT service.

LMMFIND with the LOCK parameter.
Use the SCLM LOCK service if locking is required. Use the DBUTIL service to find members in an SCLM project. ISPF services which use the LOCK keyword can also be modified to remove the keyword. Alternatively, you can use the ISPF Configuration option FAIL_ON_LMF_LOCK. If this option is set to NO, dialogs will continue to operate correctly without code modifications.

LMCOPY with the LOCK parameter
Use the SCLM LOCK service, then use LMCOPY to copy the member.

LMHIER
Use the SCLM NEXTGRP service. The NEXTGRP service returns the target group for a specified group. To simulate the data returned by LMHIER, loop on the NEXTGRP service until a return code greater than zero is returned. See SCLM example FLMNXTGP in ISP.SISPSAMP for an example.

LMREVIEW
There is no equivalent function in SCLM because the project control information is much different. Information about the contents of the hierarchy can be obtained using the SCLM DBUTIL service.

LMACT and LMDEACT
SCLM does not follow the concept of active and inactive projects. Updates are done by updating the SCLM project definition, then reassembling and relinking to create an updated load module. This load module is accessed when you enter SCLM (or exit and reenter, if SCLM is active when the new module is built).

Creating PDF Translation Tables

This section 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 the following translation tables:
- Valid data set name character translation table
- Invalid data set name character translation table
- Hexadecimal character translation table

- Numeric character translation table
- Alphanumeric character translation table
- Edit terminal output character translation table
- Generic string character translation table
- Generic string special character translation table
- Uppercase character translation table
- Lowercase character translation table.

The sample Assembler module includes all of the above 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)).

Translation Table for Valid Data Set Name Characters

The translation table for valid data set name characters (Table 24) 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 24. Example of Translation Table for Valid Data Set Characters

Table	Hexadecimal Code	Position
TTVDSN	DC X'FFFFFFFFFFFFFFFF'	(X'00' to X'07')
	DC X'FFFFFFFFFFFFFFFF'	(X'08' to X'0F')
	DC X'FFFFFFFFFFFFFFFF'	(X'10' to X'17')
	...	
	DC X'FFFF000000FFFFFF'	(X'78' to X'7F')
	DC X'FF00000000000000'	(X'80' to X'87')
	...	
	DC X'0000FFFFFFFFFFFF'	(X'E8' to X'EF')
	DC X'0000000000000000'	(X'F0' to X'F7')
	DC X'0000FFFFFFFFFFFF'	(X'F8' to X'FF')

Translation Table for Invalid Data Set Name Characters

The translation table for invalid data set name characters (Table 25) 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 25. Example of Translation Table for Invalid Data Set Name Characters

Table	Hexadecimal Code	Position
TTIDSN	DC X'0000000000000000'	(X'00' to X'07')
	DC X'0000000000000000'	(X'08' to X'0F')
	DC X'0000000000000000'	(X'10' to X'17')
	...	
	DC X'00007A7B7C000000'	(X'78' to X'7F')
	DC X'0081828384858687'	(X'80' to X'87')
	...	
	DC X'E8E9000000000000'	(X'E8' to X'EF')
	DC X'F1F2F3F4F5F6F700'	(X'F0' to X'F7')
	DC X'F8F9000000000000'	(X'F8' to X'FF')

Translation Table for Hexadecimal Characters

The translation table for hexadecimal characters (Table 26) specifies the valid hexadecimal characters as follows:

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

Table 26. Example of Translation Table for Hexadecimal Characters

Table	Hexadecimal Code	Position
TTHEX	DC X'FFFFFFFFFFFFFFFF'	(X'00' to X'07')
	DC X'FFFFFFFFFFFFFFFF'	(X'08' to X'0F')
	DC X'FFFFFFFFFFFFFFFF'	(X'10' to X'17')
	...	
	DC X'FFFFFFFFFFFFFFFF'	(X'78' to X'7F')
	DC X'FF0000000000000FF'	(X'80' to X'87')
	...	
	DC X'FFFFFFFFFFFFFFFF'	(X'E8' to X'EF')
	DC X'0000000000000000'	(X'F0' to X'F7')
	DC X'0000FFFFFFFFFFFF'	(X'F8' to X'FF')

Translation Table for Numeric Characters

The translation table for numeric characters (Table 27) specifies the valid numeric characters as follows:

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

Table 27. Example of Translation Table for Numeric Characters

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

Translation Table for Alphanumeric Characters

The translation table for alphanumeric characters (Table 28 specifies the valid alphanumeric characters as follows:

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

Table 28. Example of Translation Table for Alphanumeric Characters

Table	Hexadecimal Code	Position
TTALN	DC X'FFFFFFFFFFFFFFFF'	(X'00' to X'07')
	DC X'FFFFFFFFFFFFFFFF'	(X'08' to X'0F')
	DC X'FFFFFFFFFFFFFFFF'	(X'10' to X'17')
	...	
	DC X'FFFFFF0000FFFFFF'	(X'78' to X'7F')
	DC X'FF00000000000000'	(X'80' to X'87')
	...	
	DC X'0000FFFFFFFFFFFF'	(X'E8' to X'EF')
	DC X'0000000000000000'	(X'F0' to X'F7')
	DC X'0000FFFFFFFFFFFF'	(X'F8' to X'FF')

Translation Table for Edit Terminal Output Characters

The translation table for edit terminal output characters (Table 29) 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 29. Example of Translation Table for Edit Terminal Output Character

Table	Hexadecimal Code	Position
TTETO	DC X'1515151515151515'	(X'00' to X'07')
	DC X'1515151515151515'	(X'08' to X'0F')
	DC X'1515151515151515'	(X'10' to X'17')
	...	
	DC X'15797A7B7C7D7E7F'	(X'78' to X'7F')
	DC X'1581828384858687'	(X'80' to X'87')
	...	
	DC X'E8E9151515151515'	(X'E8' to X'EF')
	DC X'F0F1F2F3F4F5F6F7'	(X'F0' to X'F7')
	DC X'F8F9151515151515'	(X'F8' to X'FF')

Translation Table for Generic String Characters

The translation table for generic string characters (Table 30) 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 31 on page 174.

Table 30. Example of Translation Table for Generic String Characters

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

Translation Table for Generic String Special Characters

The translation table for generic string special characters (Table 31 on page 174) 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 30) 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 31, (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 31. Example of Translation Table for Generic String Special Characters

Table	Hexadecimal Code	Position
TTGSS	DC X'FFFFFFFFFFFFFFFF'	(X'00' to X'07')
	DC X'FFFFFFFFFFFFFFFF'	(X'08' to X'0F')
	DC X'FFFFFFFFFFFFFFFF'	(X'10' to X'17')
	...	
	DC X'FFFFFF070AFF00FF'	(X'78' to X'7F')
	DC X'FF81828384858687'	(X'80' to X'87')
	...	
	DC X'E8E9FFFFFFFFFFFF'	(X'E8' to X'EF')
	DC X'F0F1F2F3F4F5F6F7'	(X'F0' to X'F7')
	DC X'F8F9FFFFFFFFFFFF'	(X'F8' to X'FF')

Usage Notes for the GSC and GSS Tables

The generic string special (GSS) table is used to determine if a character is valid in a picture string and, if so, what offset into the generic string character (GSC) table describes its subset. The mask value from the GSC table is used to determine which characters satisfy the subset. Each character in the text being scanned is represented in the generic string master (GSM) table by a hexadecimal code that indicates which subsets that character belongs to (see "Creating ISPF terminal translation tables" on page 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 31, 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 32) translates data as follows:

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

Table 32. Example of Translation Table for Uppercase Characters

Table	Hexadecimal Code	Position
TTUPP	DC X'0001020304050607'	(X'00' to X'07')
	DC X'08090A0B0C0D0E0F'	(X'08' to X'0F')
	DC X'1011121314151617'	(X'10' to X'17')
	...	
	DC X'78797A7B7C7D7E7F'	(X'78' to X'7F')
	DC X'80C1C2C3C4C5C6C7'	(X'80' to X'87')
	...	
	DC X'E8E9EAEBECEDEEEF'	(X'E8' to X'EF')
	DC X'F0F1F2F3F4F5F6F7'	(X'F0' to X'F7')
	DC X'F8F9FAFBFCFDFEFF'	(X'F8' to X'FF')

Translation Table for Lowercase Characters

The translation table for lowercase characters (Table 33 on page 175) translates data as follows:

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

Table 33. Example of Translation Table for Lowercase Characters

Table	Hexadecimal Code	Position
TTLOW	DC X'0001020304050607'	(X'00' to X'07')
	DC X'08090A0B0C0D0E0F'	(X'08' to X'0F')
	DC X'1011121314151617'	(X'10' to X'17')
	...	
	DC X'78797A7B7C7D7E7F'	(X'78' to X'7F')
	DC X'8081828384858687'	(X'80' to X'87')
	...	
	DC X'A8A9EAEBECEDEEEF'	(X'E8' to X'EF')
	DC X'F0F1F2F3F4F5F6F7'	(X'F0' to X'F7')
	DC X'F8F9FAFBFCFDFEFF'	(X'F8' to X'FF')

Modifying the GSM to Use the User Character Subset

The generic string master (GSM) translation table (Table 34) and its related tables can be modified to add an additional character subset to be used in Edit picture string processing for the FIND and CHANGE commands. The GSM table is found in the ISPF translation tables, member ISPOWNTT in ISP.SISPSAMP.

Generic String Master Translation Table

The positions in the generic string master translation table (Table 34) 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 34. Example of Generic String Master Translation Table

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

To modify the GSM table to use a user character subset, follow these steps:

1. Choose a character to represent your subset. For example, Edit uses an @ to stand for alphabetic.
2. Modify the entry in the generic string special character (GSS) table that corresponds to the character you want to use so that it has a value of X'08'. This indicates where in the generic string character (GSC) table the mask for your character is located. The GSC does not need to be changed. It is initially set for user character sets.
3. Modify the GSM entries of those characters you want to include in your special character set so the high order bit is on.

For example:

If you want to define a character set of special attribute characters consisting of hexadecimal codes X'10' through X'17', and you want to use a ("") as the picture string identifier for them, you would do the following:

1. Modify the entry for X'4F' in the GSS table so it has a value of X'08' as shown in Table 35. Compare to Table 31 on page 174.

Table 35. Example of Translation Table for Modified Generic String Special Characters

Table	Hexadecimal Code	Position
TTGSS	DC X'FFFFFFFFFFFFFFFF'	(X'00' to X'07')
	DC X'FFFFFFFFFFFFFFFF'	(X'08' to X'0F')
	DC X'FFFFFFFFFFFFFFFF'	(X'10' to X'17')
	...	
	DC X'FFFFFF0105FFFF08'	(X'78' to X'7F')
	DC X'FFFFFFFFFFFFFFFF'	(X'80' to X'87')
	...	
	DC X'E8E9FFFFFFFFFFFF'	(X'E8' to X'EF')
	DC X'F0F1F2F3F4F5F6F7'	(X'F0' to X'F7')
	DC X'F8F9FFFFFFFFFFFF'	(X'F8' to X'FF')

2. Modify the GSM entries for hexadecimal codes X'10' through X'17' to turn the high order bit on as shown in Table 36. Compare to Table 34 on page 175.

Table 36. Example of Modified Generic String Master Translation Table

Table	Hexadecimal Code	Position
TTGSM	DC X'0101010101010101'	(X'00' to X'07')
	DC X'0101010101010101'	(X'08' to X'0F')
	DC X'8181818181818181'	(X'10' to X'17')
	...	
	DC X'0102020202020202'	(X'78' to X'7F')
	DC X'0110101010101010'	(X'80' to X'87')
	...	
	DC X'2020010101010101'	(X'E8' to X'EF')
	DC X'4040404040404040'	(X'F0' to X'F7')
	DC X'4040010101010101'	(X'F8' to X'FF')

You could locate the special attribute bytes by issuing the edit command FIND P'"'. If you do not want these bytes to be found under any other picture string, set the hexadecimal value to X'80'. These characters can be included in multiple character sets by setting the appropriate bits to on, according to the GSM table.

ISPF and PDF Terminal Translation Table Relationship

The terminal translation table relationship for ISPF and PDF is shown in Table 37.

Table 37. ISPF Terminal Translation Table Relationship

ISPF	PDF
ISP3277 ISP3277A ISP3277K ISP3278 ISP3278A ISP3278C ISP3278K ISP3278T NEW32TBL	ISR3277 ISR3277A ISR3277K ISR3278 ISR3278A ISR3278C ISR3278K ISR3278T ISR32TBL

The delivered ISPF terminal table names start with the prefix "ISP". ISPF does not require that user-defined terminal table names begin with the prefix "ISP";

however, PDF terminal names require the “ISR” prefix. PDF searches for the load module beginning with the fourth position of the actual table name and prefixes it with “ISR”.

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 the following sections.

Foreground Processing Panels and CLISTs

The Foreground processing option uses ISPF dialog management services. Figure 59 on page 178 shows the Foreground Selection Panel definition. Refer to the *z/OS ISPF Dialog Developer's Guide and Reference* for a general description of panel definition formats.

Note: In Figure 59, attribute characters have been replaced by blanks. Also, some of the ISPF-supplied CLISTs contain a specific library (such as SYS1.LINKLIB) on the program call. If the called program does not reside in that library, you might need to customize the CLIST.

```
)PANEL KEYLIST(ISRSAB,ISR)
)ATTR DEFAULT(...) FORMAT(MIX)              /* ISRFPA - ENGLISH - 5.6 */
 0B TYPE(AB)
 0D TYPE(PS)
 04 TYPE(ABSL) GE(ON)
 05 TYPE(PT)
 09 TYPE(FP)
 0A TYPE(NT)
 11 TYPE(SAC)
 22 TYPE(WASL) SKIP(ON) GE(ON)
 10 TYPE(ET)
 26 AREA(SCRL) EXTEND(ON)
 27 TYPE(CEF) PADC(USER) CKBOX(ON)
 28 TYPE(NEF) CAPS(ON) PADC(USER)
)ABC DESC('Menu') MNEM(1)
PDC DESC('Settings') UNAVAIL(ZPM1) MNEM(1) ACC(CTRL+S)
 ACTION RUN(ISRROUTE) PARM('SET')
PDC DESC('View') UNAVAIL(ZPM2) MNEM(1) ACC(CTRL+V)
 ACTION RUN(ISRROUTE) PARM('BR1')
PDC DESC('Edit') UNAVAIL(ZPM3) MNEM(1) ACC(CTRL+E)
 ACTION RUN(ISRROUTE) PARM('ED1')
PDC DESC('ISPF Command Shell') UNAVAIL(ZPM4) MNEM(6) ACC(CTRL+C)
 ACTION RUN(ISRROUTE) PARM('C1')
PDC DESC('Dialog Test...') UNAVAIL(ZPM5) MNEM(8) ACC(CTRL+T)
 ACTION RUN(ISRROUTE) PARM('DAL')
PDC DESC('Other IBM Products...') UNAVAIL(ZPM6) MNEM(1) ACC(CTRL+O)
 ACTION RUN(ISRROUTE) PARM('OIB')
PDC DESC('SCLM') UNAVAIL(ZPM7) MNEM(3) ACC(CTRL+L)
 ACTION RUN(ISRROUTE) PARM('SCL')
PDC DESC('ISPF Workplace') UNAVAIL(ZPM8) MNEM(6) ACC(CTRL+W)
 ACTION RUN(ISRROUTE) PARM('WRK')
PDC DESC('Status Area...') UNAVAIL(ZPMS) MNEM(8) ACC(CTRL+A)
 ACTION RUN(ISRROUTE) PARM('SAM')
PDC DESC('Exit') MNEM(2) PDSEP(ON) ACC(CTRL+X) ACTION RUN(EXIT)
)ABCINIT
.ZVARS=ISR@OPT
)ABC DESC('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(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')
```

Figure 59. Foreground Selection Panel Definition (ISRFPA) (Part 1 of 3)

```
PDC DESC('SuperCE') UNAVAIL(ZUT12) MNEM(3) ACC(CTRL+F3)
 ACTION RUN(ISRROUTE) PARM('U13')
PDC DESC('Search-For') UNAVAIL(ZUT13) MNEM(2) ACC(CTRL+F4)
 ACTION RUN(ISRROUTE) PARM('U14')
PDC DESC('Search-ForE') UNAVAIL(ZUT14) MNEM(4) ACC(CTRL+F5)
 ACTION RUN(ISRROUTE) PARM('U15')
)ABCINIT
.ZVARS=PDFUTIL
     &zut9 = '1'
)ABC DESC('Help') MNEM(1)
PDC DESC('General') MNEM(1) ACTION RUN(TUTOR) PARM('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(4) ACTION RUN(TUTOR) PARM('ISR43000')
PDC DESC('PL/I') MNEM(2) ACTION RUN(TUTOR) PARM('ISR45000')
PDC DESC('VS Pascal') MNEM(4) ACTION RUN(TUTOR) PARM('ISR46000')
PDC DESC('Binder/Link editor') MNEM(1) ACTION RUN(TUTOR) PARM('ISR47000')
PDC DESC('SCRIPT VS') MNEM(1) ACTION RUN(TUTOR) PARM('ISR49000')
PDC DESC('VS COBOL II debug') MNEM(1) ACTION RUN(TUTOR) PARM('ISR4AA00')
PDC DESC('OS/VS COBOL debug') MNEM(1) ACTION RUN(TUTOR) PARM('ISR4A000')
PDC DESC('FORTRAN debug') MNEM(7) ACTION RUN(TUTOR) PARM('ISR4B000')
PDC DESC('Member parts list') MNEM(1) ACTION RUN(TUTOR) PARM('ISR4C000')
PDC DESC('C/370') MNEM(3) ACTION RUN(TUTOR) PARM('ISR4D000')
PDC DESC('REXX/370') MNEM(1) ACTION RUN(TUTOR) PARM('ISR4E000')
PDC DESC('ADA/370') MNEM(2) ACTION RUN(TUTOR) PARM('ISR4F000')
PDC DESC('AD/Cycle C/370') MNEM(5) ACTION RUN(TUTOR) PARM('ISR4G000')
PDC DESC('ISPDTLC') MNEM(5) ACTION RUN(TUTOR) PARM('ISR4I000')
PDC DESC('OS/390 C/C++') MNEM(5) ACTION RUN(TUTOR) PARM('ISR4J000')
)ABCINIT
.ZVARS=FPAHELP
)BODY  CMD(ZCMD)
.. Menu. Utilities. Help.
.-------------------------------------------------------------------------
.                         .Foreground Selection Panel.
.Option ===>.Z
.SAREA38
)AREA SAREA38
.1  . Assembler             .          ..11 .*FORTRAN debug      .
.2  . COBOL                 .          ..12 . Member Parts List  .
.3  . VS FORTRAN            .          ..13 .*C/370              .
.5  . PL/I                  .          ..14 .*REXX/370           .
.6  . VS PASCAL             .          ..15 .*ADA/370            .
.7  .*Binder/Link editor    .          ..16 .*AD/Cycle C/370     .
.9  . SCRIPT/VS             .          ..18 . ISPDTLC            .
.10 .*VS COBOL II debug     .          ..19 .*OS/390 C/C++       .
.10A.*OS/VS COBOL debug     .
.. .
.  ..&multipmt                    ..*.No packed data support.            .
.  ..Z..Source Data Packed             .
)INIT
.ZVARS = '(ZCMD ZFPKEDV)'
&ZWINTTL = ' '
.HELP = ISR40000
&ZFPKEDV = ' '
  &ZFPKED = TRANS(TRUNC(&ZFPKED,1),Y,YES,*,NO) /*  DATA FORMAT CHECK  */
  &ZFPKEDV = TRANS(&ZFPKED YES,'/' NO,' ')
IF (&ZGUI = ' ')
  &MULTIPMT='Enter "/" to select option    '
ELSE
  &MULTIPMT='Check box to select option    '
.CURSOR = 'ZCMD'
```

Figure 59. Foreground Selection Panel Definition (ISRFPA) (Part 2 of 3)

```
)REINIT
REFRESH(ZFPKEDV)
)PROC
  &DSN     = ' '             /* INITIALIZE DATA SET NAME FIELD           */
  &ZORG    = ' '             /* INITIALIZE DATA SET ORGANIZATION VARIABLE */
  IF (&ZFPKEDV = ' ')
    &ZFPKED = 'NO'
  ELSE
    &ZFPKED = 'YES'
  &ZFPKED  = TRUNC(&ZFPKED,1)
  VER (&ZFPKED,NB,LIST,Y,N)              /*  Y = EXPAND PACKED DATA  */
  &ZFPKED = TRANS(TRUNC(&ZFPKED,1),Y,YES,N,NO)
  &ZFPACK = TRANS(TRUNC(&ZFPKED,1),Y,YES,N,NO)
  VPUT (ZFPACK,ZFPKED) PROFILE
&ZCMDWRK = &Z
IF (&ZCMD ¬= &Z)
  &ZCMDWRK = TRUNC(&ZCMD,'.')
  &ZTRAIL=.TRAIL
  IF (&ZCMDWRK = &Z)
    .MSG = ISRU000
&ZSEL = TRANS (TRUNC (&ZCMD,'.')
  1,'PGM(ISRFPR) PARM((ISRFP01) 1) NEWPOOL'
  2,'PGM(ISRFPR) PARM((ISRFP02) 2) NEWPOOL'
  3,'PGM(ISRFPR) PARM((ISRFP03) 3) NEWPOOL'
  5,'PGM(ISRFPR) PARM((ISRFP05) 5) NEWPOOL'
  6,'PGM(ISRFPR) PARM((ISRFP06) 6) NEWPOOL'
  7,'PGM(ISRFPR) PARM((ISRFP07) 7) NEWPOOL'
  9,'PGM(ISRFPR) PARM((ISRFP09) 9) NEWPOOL'
 10,'PGM(ISRFPR) PARM((IGZTPIN2,ISRFP10) 10) NEWPOOL'
10A,'PGM(ISRFPR) PARM((ISRFP10A) 10A) NEWPOOL'
 11,'PGM(ISRFPR) PARM((AFFFP11,DDBFP11,ISRFP11) 11) NEWPOOL'
 12,'PGM(ISRFPR) PARM((ISRFP12) 12) NEWPOOL'
 13,'PGM(ISRFPR) PARM((EDCFP13,ISRFP13) 13) NEWPOOL'
 14,'PGM(ISRFPR) PARM((FANFP14,ISRFP14) 14) NEWPOOL'
 15,'PGM(ISRALTDI) PARM(EVGFP15,,ISRFP15,*) NOCHECK'
 16,'PGM(ISRFPR) PARM((EDCFP16,ISRFP16) 16) NEWPOOL'
 18,'CMD(ISPDTLC (PANEL RETURN)) MODE(FSCR)'
 19,'PGM(ISRALTDI) PARM(CBC3PE4A,+,ISRFP19,*,ISRFP19A) NEWPOOL'
 ' ',' '
   *,'?')
)PNTS
FIELD(ZPS01001) VAR(ZCMD) VAL(1)
FIELD(ZPS01002) VAR(ZCMD) VAL(11)
FIELD(ZPS01003) VAR(ZCMD) VAL(2)
FIELD(ZPS01004) VAR(ZCMD) VAL(12)
FIELD(ZPS01005) VAR(ZCMD) VAL(3)
FIELD(ZPS01006) VAR(ZCMD) VAL(13)
FIELD(ZPS01007) VAR(ZCMD) VAL(5)
FIELD(ZPS01008) VAR(ZCMD) VAL(14)
FIELD(ZPS01009) VAR(ZCMD) VAL(6)
FIELD(ZPS01010) VAR(ZCMD) VAL(15)
FIELD(ZPS01011) VAR(ZCMD) VAL(7)
FIELD(ZPS01012) VAR(ZCMD) VAL(16)
FIELD(ZPS01013) VAR(ZCMD) VAL(9)
FIELD(ZPS01014) VAR(ZCMD) VAL(18)
FIELD(ZPS01015) VAR(ZCMD) VAL(10)
FIELD(ZPS01016) VAR(ZCMD) VAL(19)
FIELD(ZPS01017) VAR(ZCMD) VAL(10A)
)END

/* 5694-A01 (C) COPYRIGHT IBM CORP 1980, 2004 */
/* ISPDTLC Release: 5.6.  Level: PID                                  */
/* z/OS 01.06.00.  Created - Date: 19 Nov 2003, Time: 12:37           */
```

Figure 59. Foreground Selection Panel Definition (ISRFPA) (Part 3 of 3)

Each Foreground option has an associated panel and CLIST. For example, option 1 has the option panel name ISRFP01 with the corresponding CLIST name ISRFC01. Figure 60 shows the ISRFP01 option panel definition.

Note: In Figure 60, attribute characters have been replaced by blanks.

```
)PANEL KEYLIST(ISRSAB,ISR)
)ATTR DEFAULT(...) FORMAT(MIX)             /* ISRFP01 - ENGLISH - 5.2 */
 0B TYPE(AB)
 04 TYPE(ABSL) GE(ON)
 05 TYPE(PT)
 09 TYPE(FP)
 0A TYPE(NT)
 0C TYPE(NT) SKIP(ON)
 11 TYPE(SAC)
 12 TYPE(CEF) PADC(USER)
 19 TYPE(DT)
 22 TYPE(WASL) SKIP(ON) GE(ON)
 08 TYPE(CH)
 26 TYPE(NEF) CAPS(ON) PADC(USER)
 27 AREA(SCRL) EXTEND(ON)
 28 TYPE(SAC) CSRGRP(99) RADIO(ON)
)ABC DESC('Menu') MNEM(1)
PDC DESC('Settings') UNAVAIL(ZPM1) MNEM(1) ACC(CTRL+S)
 ACTION RUN(ISRROUTE) PARM('SET')
PDC DESC('View') UNAVAIL(ZPM2) MNEM(1) ACC(CTRL+V)
 ACTION RUN(ISRROUTE) PARM('BR1')
PDC DESC('Edit') UNAVAIL(ZPM3) MNEM(1) ACC(CTRL+E)
 ACTION RUN(ISRROUTE) PARM('ED1')
PDC DESC('ISPF Command Shell') UNAVAIL(ZPM4) MNEM(6) ACC(CTRL+C)
 ACTION RUN(ISRROUTE) PARM('C1')
PDC DESC('Dialog Test...') UNAVAIL(ZPM5) MNEM(8) ACC(CTRL+T)
 ACTION RUN(ISRROUTE) PARM('DAL')
PDC DESC('Other IBM Products...') UNAVAIL(ZPM6) MNEM(1) ACC(CTRL+O)
 ACTION RUN(ISRROUTE) PARM('OIB')
PDC DESC('SCLM') UNAVAIL(ZPM7) MNEM(3) ACC(CTRL+L)
 ACTION RUN(ISRROUTE) PARM('SCL')
PDC DESC('ISPF Workplace') UNAVAIL(ZPM8) MNEM(6) ACC(CTRL+W)
 ACTION RUN(ISRROUTE) PARM('WRK')
PDC DESC('Status Area...') UNAVAIL(ZPMS) MNEM(8) ACC(CTRL+A)
 ACTION RUN(ISRROUTE) PARM('SAM')
PDC DESC('Exit') MNEM(2) PDSEP(ON) ACC(CTRL+X) ACTION RUN(EXIT)
)ABCINIT
.ZVARS=ISR@OPT
)ABC DESC('RefList') MNEM(1)
PDC DESC('Current Data Set List &ZDSCURT') MNEM(1) ACC(CTRL+ALT+P)
 ACTION RUN(ISRRLIST) PARM('PL1')
PDC DESC('Current Library List &ZDSCURLT') MNEM(2) ACC(CTRL+SHIFT+P)
 ACTION RUN(ISRRLIST) PARM('LL1')
PDC DESC('List of Personal Data Set Lists') MNEM(1) PDSEP(ON) ACC(CTRL+ALT+O)
 ACTION RUN(ISRRLIST) PARM('PL2')
PDC DESC('List of Personal Library Lists') MNEM(2) ACC(CTRL+SHIFT+O)
 ACTION RUN(ISRRLIST) PARM('LL2')
```

Figure 60. Foreground Assembler H Definition (ISRFP01) (Part 1 of 5)

```
)ABCINIT
.ZVARS=REFLIST
      VGET (ZCURTB ZCURLTB) PROFILE
      IF (&ZCURTB = &Z) &ZDSCURT = &Z
      ELSE &ZDSCURT= '(&ZCURTB)'
      IF (&ZCURLTB = &Z) &ZDSCURLT = &Z
      ELSE &ZDSCURLT= '(&ZCURLTB)'
)ABC DESC('Utilities') MNEM(1)
PDC DESC('Library') UNAVAIL(ZUT1) MNEM(1) ACC(ALT+1)
 ACTION RUN(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(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-ForE') UNAVAIL(ZUT14) MNEM(4) ACC(CTRL+F5)
 ACTION RUN(ISRROUTE) PARM('U15')
)ABCINIT
.ZVARS=PDFUTIL
     &zut9 = '1'
)ABC DESC('Help') MNEM(1)
PDC DESC('Input Data Set') MNEM(1) ACTION RUN(TUTOR) PARM('ISR41001')
PDC DESC('Macro and copy libraries') MNEM(1) ACTION RUN(TUTOR) PARM('ISR41002')
PDC DESC('Object Data Set') MNEM(1) ACTION RUN(TUTOR) PARM('ISR41003')
PDC DESC('Listing Data Set') MNEM(1) ACTION RUN(TUTOR) PARM('ISR41004')
PDC DESC('Password protection') MNEM(1) ACTION RUN(TUTOR) PARM('ISR41005')
PDC DESC('Assembler selection') MNEM(1) ACTION RUN(TUTOR) PARM('ISR41006')
PDC DESC('Example') MNEM(1) ACTION RUN(TUTOR) PARM('ISR41007')
PDC DESC('Appendices') MNEM(5) ACTION RUN(TUTOR) PARM('ISR00004')
PDC DESC('Index') MNEM(3) ACTION RUN(TUTOR) PARM('ISR91000')
)ABCINIT
.ZVARS=FP1HELP
```

Figure 60. Foreground Assembler H Definition (ISRFP01) (Part 2 of 5)

```
)BODY  CMD(ZCMD)
.. Menu. RefList. Utilities. Help.
.------------------------------------------------------------------------------
.                          .Foreground Assembler.                             .
.Command ===>.Z                                                               .
.SAREA39                                                                      .
)AREA SAREA39
.ISPF Library:.                                                               .
. ..Project . . ..Z       .
. ..Group . . . ..Z       .. . ..Z       .. . ..Z       .. . ..Z       .
. ..Type  . . . ..Z       .
. ..Member  . . ..Z       ..(Blank or pattern for member selection list).
.
.Other Partitioned or Sequential Data Set:.                                   .
. ..Data Set Name  . ..Z                                                      .
.. .
.List ID . . ..Z       .   .Assembler.
.Password  . ..Z       .   .Z..1..High Level Assembler..2..Assembler H        .
.
.Assembler Options: (Options OBJECT and LIST generated automatically).        .
. ..    ===>.Z                                                          .
.
.Additional input libraries:.                                                 .
. ..    ===>.Z                                                               .
. ..    ===>.Z                                                               .
. ..    ===>.Z                                                               .
)INIT
.ZVARS = '(ZCMD PRJ1 LIB1 LIB2 LIB3 LIB4 TYP1 MEM DSN LID PSWD ZASMOPT FHASM +
           FHAL1 FHAL2 FHAL3)'
.HELP = ISR41000
.ATTR(ZASMOPT)='CSRGRP(99) RADIO(ON)'
.ATTR(PSWD)='INTENS(NON)'
  &ZUT6  = 1
  &ZUT7  = 1
  &ZUT11 = 1
  &ZUT12 = 1
  &ZUT13 = 1
  &ZUT14 = 1
  &ZMLCSR = '        '                        /*                       @M1A*/
  .HELP = ISR41A00
  &TYP1 = &ASMT                              /*ASSEMBLER TYPE VARIABLE NAME*/
  IF (&ZORG = 'PS')
    IF (&LID = ' ')      .CURSOR = LID
  IF (&DSN ¬= ' ')
    &MEM = ' '                                /*                       @M1A*/
    IF (.CURSOR = ' ')   .CURSOR = DSN
  &LID  = ' '
  IF (&ZFTEMP = '')
    &ZFPACK = &ZFPACK
    VPUT (ZFPACK) SHARED
    &ZFTEMP = '0'
  IF (&ZASMOPT ¬= 1)
    IF (&ZASMOPT ¬= 2)
      &ZASMOPT = 1
```

Figure 60. Foreground Assembler H Definition (ISRFP01) (Part 3 of 5)

```
)REINIT
REFRESH(PRJ1 LIB1 LIB2 LIB3 LIB4 TYP1 MEM DSN FHAL1 FHAL2 FHAL3)
IF (&ZNXTMSG='ISRT') .CSRPOS = &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 */
&ZCSRP = .CSRPOS                             /* AUTOTYPE */
&ZODSNLN = 0                                 /* AUTOTYPE */
&ZODSNMB = &Z                              /* AUTOTYPE */
&ZNAMES='ZCSRV ZCSRP PRJ1 LIB1 LIB2 LIB3 LIB4 TYP1 MEM '
&ZNAMES='&ZNAMES *.&ZODSNLN&ZODSNMB ZCMD'
IF (.CURSOR = DSN, FHAL1, FHAL2, FHAL3)
  &ZODSNLN = 56
  &ZODSNMB = &Z
  IF (.CURSOR = FHAL1) &ZODSNLN = 65
  IF (.CURSOR = FHAL1) &ZODSNMB = '%'
  IF (.CURSOR = FHAL2) &ZODSNLN = 65
  IF (.CURSOR = FHAL2) &ZODSNMB = '%'
  IF (.CURSOR = FHAL3) &ZODSNLN = 65
  IF (.CURSOR = FHAL3) &ZODSNMB = '%'
  &ZNAMES='ZCSRV ZCSRP * * * * * * * &ZCSRV&ZODSNLN&ZODSNMB ZCMD'
PANEXIT((ZNAMES),LOAD,ISRAUTOT)               /* AUTOTYPE */
IF (&ZNXTMSG='ISRT') EXIT                    /* AUTOTYPE */
VER(&ZASMOPT,NONBLANK)
VER(&ZASMOPT RANGE,1,2)
&ZUT6  = 0
&ZUT7  = 0
&ZUT11 = 0
&ZUT12 = 0
&ZUT13 = 0
&ZUT14 = 0
 VGET (ZRDSN) SHARED                         /* REFERENCE LIST CODE       */
 IF (&ZRDSN ¬= ' ')                          /* IF DATA SET SELECTED      */
   &DSN = &ZRDSN                             /*    PUT DSN VARIABLE       */
   &ZRDSN = ' '                              /*     INTO PANEL            */
   &ZRVOL = ' '                              /*     INTO PANEL            */
   VPUT (ZRDSN ZRVOL) SHARED                 /*                           */
   .CURSOR = DSN
   .MSG = ISRDS003                           /*    MSG PENDING            */
```

Figure 60. Foreground Assembler H Definition (ISRFP01) (Part 4 of 5)

```
VGET (DSALSEL) SHARED                          /*                          */
 IF (&DSALSEL ¬= ' ')                           /* IF LIBRARY SELECTED       */
   VGET (DSA1,DSA2,DSA3,DSA4,DSA5,DSA6,DSA7) SHARED
   &PRJ1 = &DSA1                               /*    PUT LIBRARY VARIABLES  */
   &LIB1 = &DSA2                               /*      INTO PANEL           */
   &LIB2 = &DSA3                               /*         .                 */
   &LIB3 = &DSA4                               /*         .                 */
   &LIB4 = &DSA5                               /*         .                 */
   &TYP1 = &DSA6                               /*         .                 */
   &MEM  = &DSA7                               /*         .                 */
   &DSN = ' '                                  /*    BLANK OUT DSN          */
   &DSALSEL = ' '                         /*    CLEAR LIBRARY SELECTION */
   VPUT (DSALSEL) SHARED                       /*                           */
   .CURSOR = MEM
   .MSG = ISRDS003                             /*    MSG PENDING            */
                                               /*                           */
 IF (&ZCMD ¬= ' ')  .MSG = ISPZ001             /* INVALID COMMAND           */
 IF (&DSN = ' ')                               /* IF NOT OTHER DATA SET,    */
   VER (&PRJ1,NB)                              /* VERIFY LIBRARY FIELDS ARE */
   VER (&LIB1,NB)                              /* INPUT                     */
   VER (&TYP1,NB)
 IF (&DSN ¬= ' ')                              /*                       @M1A*/
   VER(&DSN DSNAMEFM)
 IF (&LID ¬= '*')  VER (&LID,NAME)             /* LIST ID MUST BE VALID NAME */
 VER (&PSWD,INCLUDE,ALPHA,NUM,MSG=ISRC609)     /* Check password     @V9A*/
 IF (&ZASMOPT = 1)
   &FHALEV = 'HLASM'
 IF (&ZASMOPT = 2)
   &FHALEV = 'HASM'
 IF (&FHAL1 ¬= ' ')                            /*                       @M2A*/
   VER (&FHAL1,DSNAMEPQ)                       /* VERIFY ADDITIONAL LIB1@DSNQ*/
 IF (&FHAL2 ¬= ' ')                            /*                       @M2A*/
   VER (&FHAL2,DSNAMEPQ)                       /* VERIFY ADDITIONAL LIB2@DSNQ*/
 IF (&FHAL3 ¬= ' ')                            /*                       @L2A*/
   VER (&FHAL3,DSNAMEPQ)                       /* VERIFY ADDITIONAL LIB2@DSNQ*/
 &ASMT = &TYP1                                 /* SAVE ASSEMBLER TYPE       */
 &ZSEL = 'CMD(%ISRFC01)'                       /* EXECUTE ASSEMBLE CLIST    */
                                               /* Begin @L3A                */
 &ZSYSDS1 = &FHAL1                             /* Fill fields for ISRJFSYS  */
 &ZSYSDS2 = &FHAL2                             /* to use as input.          */
 &ZSYSDS3 = &FHAL3                             /* ZSYSDS? is a qualified    */
 &ZSYSCUR1 = 'FHAL1'                           /*     dataset.              */
 &ZSYSCUR2 = 'FHAL2'                           /* ZSYSCUR? is were the cursor*/
 &ZSYSCUR3 = 'FHAL3'                           /*     is placed on a error. */
 VPUT (ZSYSDS1 ZSYSDS2 ZSYSDS3 ZSYSCUR1 ZSYSCUR2 ZSYSCUR3) SHARED
                                               /* End   @L3A                */
 VPUT (PRJ1,LIB1,LIB2,LIB3,LIB4,ASMT,FHASM) PROFILE        /* OY14824*/
 VPUT (FHAL1,FHAL2,FHAL3,FHALEV,ZASMOPT) PROFILE
 VPUT (PRJ1,LIB1,LIB2,LIB3,LIB4,ASMT,FHASM,DSN,LID) SHARED /*    @L2C*/
 &ZFBROWS = 'ISRBROB ' /*BROWSE LISTING   , IF BLANK NO AUTO BROWSE  */
 &ZFPRINT = 'ISRFPPRT' /*PRINT PANEL NAME, IF BLANK NO AUTO PRINT PNL*/
 VPUT (ZFBROWS,ZFPRINT,FHAL1,FHAL2,FHAL3,FHALEV) SHARED    /* OW10516*/
)END
/* 5694-A01 (C) COPYRIGHT IBM CORP 1980, 2001                          */
/* ISPDTLC Release: 5.2.  Level: PID                                   */
/* z/OS 01.02.00.  Created - Date: 10 Mar 2001, Time: 14:25            */
/* OW21977 - 960813 - OS/390 R2 ROLLUP APAR - OW19891, OW20382         */
/* OW10516 - 950113 - Add High Level Assembler.  GT4045 - MOS          */
```

Figure 60. Foreground Assembler H Definition (ISRFP01) (Part 5 of 5)

Table 38 on page 186 lists the names of the ISPF-supplied panels and CLISTs for the Foreground processing option.

Table 38. ISPF-Supplied Panels and CLISTs for Foreground Processing Option

Option	Description	Panel ID	CLIST ID
-	FOREGROUND SELECTION MENU	ISRFPA	--
1	ASSEMBLER	ISRFP01	ISRFC01
2	VS COBOL II	ISRFP02	ISRFC02
3	FORTRAN COMPILE	ISRFP03	ISRFC03
5	PLI OPTIMIZER COMPILE	ISRFP05	ISRFC05
6	VS PASCAL COMPILE	ISRFP06	ISRFC06
7	BINDER/LINK EDIT	ISRFP07	ISRFC07
9	SCRIPT/VS	ISRFP09	ISRFC09
10	VS COBOL II DEBUG	--	--
10A	COBOL INTERACTIVE DEBUG	ISRFP10A	ISRFC10A
11	FORTRAN INTERACTIVE DEBUG	ISRFP11	ISRFC11
12	MEMBER PARTS LIST	ISRFP12	ISRFC12
13	C/370 COMPILE	--	--
14	REXX/370 COMPILE	--	--
15	ADA/370 COMPILE	--	--
16	AD/CYCLE C/370 COMPILE	--	--
17	AD/CYCLE C/370 COBOL/370	--	--
18	ISPDTLC	ISPCP01	--
19	OS/390 C/C++	--	--

Figure 61 shows the overall flow of control for foreground processing.

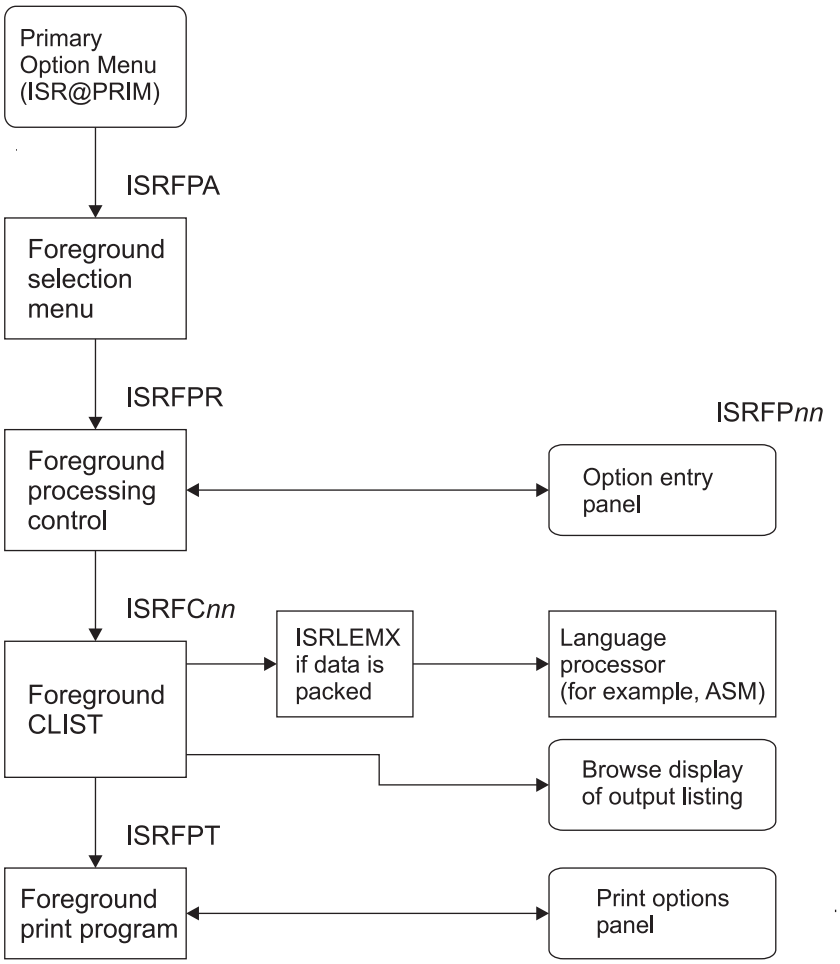


Figure 61. Foreground Processing Flow

ISRFPA is the Foreground Selection Panel. Each option on this menu translates to the following 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 188. 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

The following 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 above 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

The following field is optional:

PSWD OS password

The ISRFPR program verifies and processes these variables and then stores the following 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.

The following sections describe additional variables.

Variables that Control Automatic Browse and Print

The process of automatically invoking the Browse function for a generated list data set, and subsequently displaying the Foreground Print Options panel is completely external to the ISRFPR program. The ZFBROWS and ZFPRINT variables control these functions. These variables reside in the shared variable pool. Select the distributed Foreground option on the ISRFPA Foreground Selection panel to store ZFBROWS and ZFPRINT in the shared variable pool. ZFBROWS is set to "ISRBROB" (use any nonblank name) and ZFPRINT is set to "ISRFPPRT" (the print panel name).

Use the BROWSE service to invoke the browse function from each CLIST. Use the SELECT service to invoke the print function from each CLIST. SELECT service invokes the ISRFPT program (load module). ISRFPT is distributed with PDF.

ZFPRINT is the variable that is set to the name of the panel to set the print options panel name the dialog developer calls.

The ZFLID variable in ISRFPPRT (Foreground Print Options panel) should be set to the data set name you are going to print. Use the VPUT service to put both the ZFPRINT and ZFLID variables in the shared variable pool. If you do not want to invoke automatic browse or print, set the corresponding variables to blank in either the individual option panels or CLISTs. If you want to set the variables from a panel, remember to use VPUT to store the variables in the shared variable pool so the CLISTs can access them.

When the language processor sends a return code greater than 12 to the ISPF-supplied CLISTs, CLIST bypasses automatic browse and display of the print panel. The exceptions to this are the PL/I Checkout compiler and the COBOL and FORTRAN Interactive Debug programs. The CLISTs for these programs do not display browse or print if the return code is a system return code (for example, “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 (ISRFP07) after processing the FORTRAN panel (ISRFP03), set ZNEXTPN to ISRFP07 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 188.

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, you should be aware of the following items:

- Additional variables such as LANG and ZORG are used to pass information to log messages, error messages, and panels.
- The following 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 below.

1. Add the new option (for example, '10') to the ISPF Primary Option Menu, panel ISR@PRIM. In the translated value for option 10, use the PANEL keyword to specify the name of the selection panel to be displayed next. For example:

```
)PROC
  &ZSEL = TRANS( TRUNC(&ZCMD,'.')
                . . .

               10,'PANEL(XYZ)'
```

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

```
)PROC
  &ZSEL = TRANS( TRUNC(&ZCMD,'.')
                . . .

                2,'PGM(ISRFPR) PARM((FORNEW) 2)'
```

3. Proceed as specified in "Steps to Add a New Foreground Option to the Foreground Selection Panel," starting at step 2.

Steps to Add a New Foreground Option to the Foreground Selection Panel: The steps required to add a new option to the PDF Foreground Selection Panel are listed below.

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:

```
)PROC
  &ZSEL = TRANS( TRUNC(&ZCMD,'.')
                . . .
                99,'PGM(ISRFPR) PARM((FORNEW) 99)'
                . . .
```

2. Add new option panel FORNEW to the panel library. Use one of the distributed option panels (for example, ISRFP01) as a model.
3. Develop a corresponding CLIST (referenced from panel FORNEW by the ZSEL variable). Use one of the distributed CLISTs (for example, ISRFC01) as a model. Add the CLIST to a library accessible to ddname SYSPROC.

Batch Processing Panels, CLISTs, and Skeletons

The Batch option uses ISPF dialog management services. Figure 62 and Figure 63 on page 196 show the Batch Selection and Batch JCL generation panel formats. Refer to the *z/OS ISPF Dialog Developer's Guide and Reference* for a general description of panel definition formats.

Note: In Figure 62 and Figure 63 on page 196, attribute characters have been replaced by blanks.

```
)PANEL KEYLIST(ISRSAB,ISR)
)ATTR DEFAULT(...) FORMAT(MIX)              /* ISRJPA - ENGLISH - 5.6 */
 0B TYPE(AB)
 0D TYPE(PS)
 04 TYPE(ABSL) GE(ON)
 05 TYPE(PT)
 09 TYPE(FP)
 0A TYPE(NT)
 0C TYPE(NT) SKIP(ON)
 11 TYPE(SAC)
 22 TYPE(WASL) SKIP(ON) GE(ON)
 08 TYPE(CH)
 10 TYPE(ET)
 26 AREA(SCRL) EXTEND(ON)
 27 TYPE(CEF) PADC(USER) CKBOX(ON)
 28 TYPE(NEF) CAPS(ON) PADC(USER)
)ABC DESC('Menu') MNEM(1)
PDC DESC('Settings') UNAVAIL(ZPM1) MNEM(1) ACC(CTRL+S)
 ACTION RUN(ISRROUTE) PARM('SET')
PDC DESC('View') UNAVAIL(ZPM2) MNEM(1) ACC(CTRL+V)
 ACTION RUN(ISRROUTE) PARM('BR1')
PDC DESC('Edit') UNAVAIL(ZPM3) MNEM(1) ACC(CTRL+E)
 ACTION RUN(ISRROUTE) PARM('ED1')
PDC DESC('ISPF Command Shell') UNAVAIL(ZPM4) MNEM(6) ACC(CTRL+C)
 ACTION RUN(ISRROUTE) PARM('C1')
PDC DESC('Dialog Test...') UNAVAIL(ZPM5) MNEM(8) ACC(CTRL+T)
 ACTION RUN(ISRROUTE) PARM('DAL')
PDC DESC('Other IBM Products...') UNAVAIL(ZPM6) MNEM(1) ACC(CTRL+O)
 ACTION RUN(ISRROUTE) PARM('OIB')
PDC DESC('SCLM') UNAVAIL(ZPM7) MNEM(3) ACC(CTRL+L)
 ACTION RUN(ISRROUTE) PARM('SCL')
PDC DESC('ISPF Workplace') UNAVAIL(ZPM8) MNEM(6) ACC(CTRL+W)
 ACTION RUN(ISRROUTE) PARM('WRK')
PDC DESC('Status Area...') UNAVAIL(ZPMS) MNEM(8) ACC(CTRL+A)
 ACTION RUN(ISRROUTE) PARM('SAM')
PDC DESC('Exit') MNEM(2) PDSEP(ON) ACC(CTRL+X) ACTION RUN(EXIT)
)ABCINIT
.ZVARS=ISR@OPT
)ABC DESC('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')
```

Figure 62. Batch Selection Panel Definition (ISRJPA) (Part 1 of 4)

```
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-ForE') UNAVAIL(ZUT14) MNEM(4) ACC(CTRL+F5)
 ACTION RUN(ISRROUTE) PARM('U15')
)ABCINIT
.ZVARS=PDFUTIL
     &zut9 = '1'
)ABC DESC('Help') MNEM(1)
PDC DESC('General') MNEM(1) ACTION RUN(TUTOR) PARM('ISR50010')
PDC DESC('Assembler') MNEM(1) ACTION RUN(TUTOR) PARM('ISR51000')
PDC DESC('COBOL') MNEM(1) ACTION RUN(TUTOR) PARM('ISR52000')
PDC DESC('VS Fortran') MNEM(4) ACTION RUN(TUTOR) PARM('ISR53000')
PDC DESC('PL/I') MNEM(2) ACTION RUN(TUTOR) PARM('ISR55000')
PDC DESC('VS Pascal') MNEM(4) ACTION RUN(TUTOR) PARM('ISR56000')
PDC DESC('Binder/Link editor') MNEM(1) ACTION RUN(TUTOR) PARM('ISR57000')
PDC DESC('VS COBOL II debug') MNEM(1) ACTION RUN(TUTOR) PARM('ISR5A000')
PDC DESC('Member parts list') MNEM(1) ACTION RUN(TUTOR) PARM('ISR5C000')
PDC DESC('C/370') MNEM(3) ACTION RUN(TUTOR) PARM('ISR5D000')
PDC DESC('REXX/370') MNEM(1) ACTION RUN(TUTOR) PARM('ISR5E000')
PDC DESC('ADA/370') MNEM(2) ACTION RUN(TUTOR) PARM('ISR5F000')
PDC DESC('AD/Cycle C/370') MNEM(5) ACTION RUN(TUTOR) PARM('ISR5G000')
PDC DESC('ISPDTLC') MNEM(5) ACTION RUN(TUTOR) PARM('ISR5I000')
PDC DESC('OS/390 C/C++') MNEM(1) ACTION RUN(TUTOR) PARM('ISR5J000')
PDC DESC('Appendices') MNEM(4) ACTION RUN(TUTOR) PARM('ISR00004')
)ABCINIT
.ZVARS=JPAHELP
)BODY  CMD(ZCMD)
.. Menu. Utilities. Help.
.------------------------------------------------------------------------------
.                          .Batch Selection Panel.                            .
.Option ===>.Z                                                                .
.SAREA38                                                                      .
)AREA SAREA38
.1 .Assembler           ...7 .*Binder/Link editor ...15.*ADA/370           ..
.2 .COBOL               ...10.*VS COBOL II debug  ...16.*AD/Cycle C/370    ..
.3 .VS FORTRAN          ...12. Member Parts List  ...18. ISPDTLC           ..
.5 .PLI                 ...13.*C/370              ...19.*OS/390 C/C++      ..
.6 .VS PASCAL           ...14.*REXX/370           .
.. .
.  ..&multipmt                          ..*.No packed data support.            .
.  ..Z..Source data online              .
.  ..Z..Source data packed              .
.
.Job Statement Information:.Verify before proceeding.                          .
.                                                                              .
.===>.Z                                                                       .
.===>.Z                                                                       .
.===>.Z                                                                       .
.===>.Z                                                                       .
```

Figure 62. Batch Selection Panel Definition (ISRJPA) (Part 2 of 4)

```
)INIT
.ZVARS = '(ZCMD ZDSCKOV ZBPKEDV BJC1 BJC2 BJC3 BJC4)'
&ZWINTTL = ' '
.HELP = ISR50000
&ZDSCKOV = ' '
&ZBPKEDV = ' '
&ZUT6  = 1
&ZUT7  = 1
&ZUT11 = 1
&ZUT12 = 1
&ZUT13 = 1
&ZUT14 = 1
IF (&ZDSCKO = ' ')
  &ZDSCKO = Y
&ZDSCKO = TRANS(TRUNC(&ZDSCKO,1),N,NO,*,YES)
&ZDSCKOV = TRANS(&ZDSCKO YES,'/' NO,' ')
IF (&ZBPKED = &Z)
  &ZBPKED = N
&ZBPKED = TRANS(TRUNC(&ZBPKED,1),Y,YES,*,NO)
&ZBPKEDV = TRANS(&ZBPKED YES,'/' NO,' ')
IF (&ZGUI = ' ')
  &MULTIPMT='Enter "/" to select option    '
ELSE
  &MULTIPMT='Check box to select option    '
.CURSOR = 'ZCMD'
)REINIT
REFRESH(ZDSCKOV ZBPKEDV)
&ZUT6  = 1
&ZUT7  = 1
&ZUT11 = 1
&ZUT12 = 1
&ZUT13 = 1
&ZUT14 = 1
)PROC
&ZUT6  = 0
&ZUT7  = 0
&ZUT11 = 0
&ZUT12 = 0
&ZUT13 = 0
&ZUT14 = 0
IF (&ZDSCKOV = ' ')
  &ZDSCKO = 'NO'
ELSE
  &ZDSCKO = 'YES'
&ZDSCKO = TRUNC(&ZDSCKO,1)      /*DATA SET (DS) CHECK:             */
&ZDSCHK = TRANS(TRUNC(&ZDSCKO,1),Y,Y,N,N)
&ZDSCKO = TRANS(TRUNC(&ZDSCKO,1),Y,YES,N,NO)
IF (&ZBPKEDV = ' ')
  &ZBPKED = 'NO'
ELSE
  &ZBPKED = 'YES'
```

Figure 62. Batch Selection Panel Definition (ISRJPA) (Part 3 of 4)

```
&ZBPKED  = TRUNC(&ZBPKED,1)
&ZBPKED = TRANS(TRUNC(&ZBPKED,1),Y,YES,N,NO)
&ZBPACK = TRANS(TRUNC(&ZBPKED,1),Y,YES,N,NO)
VER (&ZDSCKO,NB,LIST,YES,NO)           /* Y= VERIFY DSN;N= NO VERIFICATION*/
VER (&ZBPKED,NB,LIST,YES,NO)           /*  Y = EXPAND PACKED DATA  */
&DSN  = ' '                            /* INITIALIZE DATA SET NAME FIELD  */
VPUT (ZDSCHK,ZDSCKO,DSN) SHARED              /* PLACE IN SHARED POOL FOR ISRJB2 */
&RTNPNL = ISRJPB
VPUT (BJC1,BJC2,BJC3,BJC4,ZBPACK,ZBPKED) PROFILE
&ZCMDWRK = &Z
IF (&ZCMD ¬= &Z)
  &ZCMDWRK = TRUNC(&ZCMD,'.')
  &ZTRAIL=.TRAIL
  IF (&ZCMDWRK = &Z)
    .MSG = ISRU000
&ZSEL = TRANS (TRUNC (&ZCMD,'.')
  1,'PGM(ISRJB2) PARM((ISRJP01) 1) NEWPOOL'
  2,'PGM(ISRJB2) PARM((ISRJP02) 2) NEWPOOL'
  3,'PGM(ISRJB2) PARM((ISRJP03) 3) NEWPOOL'
  5,'PGM(ISRJB2) PARM((ISRJP05) 5) NEWPOOL'
  6,'PGM(ISRJB2) PARM((ISRJP06) 6) NEWPOOL'
  7,'PGM(ISRJB2) PARM((ISRJP07) 7) NEWPOOL'
 10,'PGM(ISRJB2) PARM((ISRJP10) 10) NEWPOOL'
 12,'PGM(ISRJB2) PARM((ISRJP12) 12) NEWPOOL'
 13,'PGM(ISRJB2) PARM((EDCJP13,ISRJP13) 13) NEWPOOL'
 14,'PGM(ISRJB2) PARM((FANJP14,ISRJP14) 14) NEWPOOL'
 15,'PGM(ISRALTDI) PARM(EVGJP15,,ISRJP15,*) NOCHECK'
 16,'PGM(ISRJB2) PARM((EDCJP16,ISRJP16) 16) NEWPOOL'
 18,'CMD(ISPDTLC (PANEL SUBMIT RETURN)) MODE(FSCR)'
 19,'PGM(ISRALTDI) PARM(CBC3PE5A,+,ISRJP19,*,ISRJP19A) NOCHECK'
 ' ',' '
   *,'?')
)PNTS
FIELD(ZPS01001) VAR(ZCMD) VAL(1)
FIELD(ZPS01002) VAR(ZCMD) VAL(7)
FIELD(ZPS01003) VAR(ZCMD) VAL(15)
FIELD(ZPS01004) VAR(ZCMD) VAL(2)
FIELD(ZPS01005) VAR(ZCMD) VAL(10)
FIELD(ZPS01006) VAR(ZCMD) VAL(16)
FIELD(ZPS01007) VAR(ZCMD) VAL(3)
FIELD(ZPS01008) VAR(ZCMD) VAL(12)
FIELD(ZPS01009) VAR(ZCMD) VAL(18)
FIELD(ZPS01010) VAR(ZCMD) VAL(5)
FIELD(ZPS01011) VAR(ZCMD) VAL(13)
FIELD(ZPS01012) VAR(ZCMD) VAL(19)
FIELD(ZPS01013) VAR(ZCMD) VAL(6)
FIELD(ZPS01014) VAR(ZCMD) VAL(14)
)END
/* 5647-A01 (C) COPYRIGHT IBM CORP 1980, 1999 */
/* ISPDTLC Release: 5.6.  Level: PID                                  */
/* z/OS 01.06.00.  Created - Date: 19 Nov 2003, Time: 12:37           */
```

Figure 62. Batch Selection Panel Definition (ISRJPA) (Part 4 of 4)

```
)PANEL KEYLIST(ISRSAB,ISR)
)ATTR DEFAULT(...) FORMAT(MIX)              /* ISRJPB - ENGLISH - 5.6 */
 0B TYPE(AB)
 0D TYPE(PS)
 04 TYPE(ABSL) GE(ON)
 05 TYPE(PT)
 09 TYPE(FP)
 0A TYPE(NT)
 0C TYPE(NT) SKIP(ON)
 11 TYPE(SAC)
 22 TYPE(WASL) SKIP(ON) GE(ON)
 08 TYPE(CH)
 10 TYPE(ET)
 26 AREA(SCRL) EXTEND(ON)
 27 TYPE(CEF) PADC(USER) CKBOX(ON)
 28 TYPE(NEF) CAPS(ON) PADC(USER)
)ABC DESC('Menu') MNEM(1)
PDC DESC('Settings') UNAVAIL(ZPM1) MNEM(1) ACC(CTRL+S)
 ACTION RUN(ISRROUTE) PARM('SET')
PDC DESC('View') UNAVAIL(ZPM2) MNEM(1) ACC(CTRL+V)
 ACTION RUN(ISRROUTE) PARM('BR1')
PDC DESC('Edit') UNAVAIL(ZPM3) MNEM(1) ACC(CTRL+E)
 ACTION RUN(ISRROUTE) PARM('ED1')
PDC DESC('ISPF Command Shell') UNAVAIL(ZPM4) MNEM(6) ACC(CTRL+C)
 ACTION RUN(ISRROUTE) PARM('C1')
PDC DESC('Dialog Test...') UNAVAIL(ZPM5) MNEM(8) ACC(CTRL+T)
 ACTION RUN(ISRROUTE) PARM('DAL')
PDC DESC('Other IBM Products...') UNAVAIL(ZPM6) MNEM(1) ACC(CTRL+O)
 ACTION RUN(ISRROUTE) PARM('OIB')
PDC DESC('SCLM') UNAVAIL(ZPM7) MNEM(3) ACC(CTRL+L)
 ACTION RUN(ISRROUTE) PARM('SCL')
PDC DESC('ISPF Workplace') UNAVAIL(ZPM8) MNEM(6) ACC(CTRL+W)
 ACTION RUN(ISRROUTE) PARM('WRK')
PDC DESC('Status Area...') UNAVAIL(ZPMS) MNEM(8) ACC(CTRL+A)
 ACTION RUN(ISRROUTE) PARM('SAM')
PDC DESC('Exit') MNEM(2) PDSEP(ON) ACC(CTRL+X) ACTION RUN(EXIT)
)ABCINIT
.ZVARS=ISR@OPT
)ABC DESC('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(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')
```

Figure 63. Batch JCL Generation Panel Definition (ISRJPB) (Part 1 of 4)

```
PDC DESC('SuperC') UNAVAIL(ZUT11) MNEM(1) PDSEP(ON) ACC(CTRL+F2)
 ACTION RUN(ISRROUTE) PARM('U12')
PDC DESC('SuperCE') UNAVAIL(ZUT12) MNEM(3) ACC(CTRL+F3)
 ACTION RUN(ISRROUTE) PARM('U13')
PDC DESC('Search-For') UNAVAIL(ZUT13) MNEM(2) ACC(CTRL+F4)
 ACTION RUN(ISRROUTE) PARM('U14')
PDC DESC('Search-ForE') UNAVAIL(ZUT14) MNEM(4) ACC(CTRL+F5)
 ACTION RUN(ISRROUTE) PARM('U15')
)ABCINIT
.ZVARS=PDFUTIL
     &zut9 = '1'
)ABC DESC('Help') MNEM(1)
PDC DESC('General') MNEM(1) ACTION RUN(TUTOR) PARM('ISR50000')
PDC DESC('Assembler') MNEM(1) ACTION RUN(TUTOR) PARM('ISR51000')
PDC DESC('COBOL') MNEM(1) ACTION RUN(TUTOR) PARM('ISR52000')
PDC DESC('VS Fortran') MNEM(4) ACTION RUN(TUTOR) PARM('ISR53000')
PDC DESC('PL/I') MNEM(2) ACTION RUN(TUTOR) PARM('ISR55000')
PDC DESC('VS Pascal') MNEM(4) ACTION RUN(TUTOR) PARM('ISR56000')
PDC DESC('Binder/Link editor') MNEM(1) ACTION RUN(TUTOR) PARM('ISR57000')
PDC DESC('VS COBOL II debug') MNEM(1) ACTION RUN(TUTOR) PARM('ISR5A000')
PDC DESC('Member parts list') MNEM(1) ACTION RUN(TUTOR) PARM('ISR5C000')
PDC DESC('C/370') MNEM(3) ACTION RUN(TUTOR) PARM('ISR5D000')
PDC DESC('REXX/370') MNEM(1) ACTION RUN(TUTOR) PARM('ISR5E000')
PDC DESC('ADA/370') MNEM(2) ACTION RUN(TUTOR) PARM('ISR5F000')
PDC DESC('AD/Cycle C/370') MNEM(5) ACTION RUN(TUTOR) PARM('ISR5G000')
PDC DESC('ISPDTLC') MNEM(5) ACTION RUN(TUTOR) PARM('ISR5I000')
PDC DESC('OS/390 C/C++') MNEM(1) ACTION RUN(TUTOR) PARM('ISR5J000')
PDC DESC('Appendices') MNEM(4) ACTION RUN(TUTOR) PARM('ISR00004')
)ABCINIT
.ZVARS=JPBHELP
)BODY  CMD(ZCMD)
.. Menu. Utilities. Help.
.------------------------------------------------------------------------------
.                          .Batch Selection Panel.                            .
.Option ===>.Z                                                                .
.SAREA38                                                                      .
)AREA SAREA38
.1 .Assembler           ...7 .*Binder/Link editor ...15.*ADA/370           ..
.2 .COBOL               ...10.*VS COBOL II debug  ...16.*AD/Cycle C/370    ..
.3 .VS FORTRAN          ...12. Member Parts List  ...18. ISPDTLC           ..
.5 .PLI                 ...13.*C/370              ...19.*OS/390 C/C++      ..
.6 .VS PASCAL           ...14.*REXX/370           .
.              ..
.              ..*.No packed data support.                                     .
.Instructions:.                                                               .
. ..Enter option to continue generating JCL,.CANCEL.command to exit without.   .
. ..submitting job or.END.command to &ZBMSG                                   .
. ... .
. ..&multipmt                          .
. ..Z..Source data online               .
. ..Z..Source data packed               .
.
.Job Statement Information:.                                                  .
.                                                                             .
.===>.Z                                                                      .
.===>.Z                                                                      .
.===>.Z                                                                      .
.===>.Z                                                                      .
```

Figure 63. Batch JCL Generation Panel Definition (ISRJPB) (Part 2 of 4)

```
)INIT
.ZVARS = '(ZCMD ZDSC ZBPK BJC1 BJC2 BJC3 BJC4)'
&ZWINTTL = ' '
.HELP = ISR50000
&ZDSC = ' '
&ZBPK = ' '
&ZUT6  = 1
&ZUT7  = 1
&ZUT11 = 1
&ZUT12 = 1
&ZUT13 = 1
&ZUT14 = 1
IF (&ZDSCKO = ' ')
  &ZDSCKO = Y
&ZDSCKO = TRANS(TRUNC(&ZDSCKO,1),N,NO,*,YES)
&ZDSC = TRANS(&ZDSCKO NO,' ' YES,'/')
IF (&ZBPKED = &Z)
  &ZBPKED = N
&ZBPKED = TRANS(TRUNC(&ZBPKED,1),Y,YES,*,NO)
&ZBPK = TRANS(&ZBPKED NO,' ' YES,'/')
IF (&ZJOBSTEP = 'YES')
  &ZBMSG = 'submit job.'
IF (&ZJOBSTEP = 'NO ')
  &ZBMSG = 'exit without submitting job.'
IF (&ZGUI = ' ')
  &MULTIPMT='Enter "/" to select option    '
ELSE
  &MULTIPMT='Check box to select option    '
.CURSOR = 'ZCMD'
)REINIT
REFRESH(ZDSC ZBPK)
&zut6 = 1
&zut7 = 1
&zut11 = 1
&zut12 = 1
&zut13 = 1
&zut14 = 1
)PROC
&zut6 = 0
&zut7 = 0
&zut11 = 0
&zut12 = 0
&zut13 = 0
&zut14 = 0
&ZDSCKO = TRANS(&ZDSC ' ','NO' *,'YES')
&ZBPKED = TRANS(&ZBPK ' ','NO' *,'YES')
&ZDSCKO = TRUNC(&ZDSCKO,1)      /*DATA SET (DS) CHECK:             */
VER (&ZDSCKO,NB,LIST,Y,N)       /* Y= VERIFY DSN;N= NO VERIFICATION*/
&ZDSCHK = TRANS(TRUNC(&ZDSCKO,1),Y,Y,N,N)
&ZBPKED  = TRUNC(&ZBPKED,1)
VER (&ZBPKED,NB,LIST,Y,N)               /*  Y = EXPAND PACKED DATA  */
&ZBPKED = TRANS(TRUNC(&ZBPKED,1),Y,YES,N,NO)
&ZBPACK = TRANS(TRUNC(&ZBPKED,1),Y,YES,N,NO)
&DSN  = ' '                     /* INITIALIZE DATA SET NAME FIELD  */
VPUT (ZDSCHK,ZDSCKO,DSN) SHARED /* PLACE IN SHARED POOL FOR ISRJB2 */
VPUT (ZBPACK,ZBPKED) PROFILE
&ZSEL = TRANS (&ZCMD
                C,C
              CAN,C
           CANCEL,C
                *,'*' )
```

Figure 63. Batch JCL Generation Panel Definition (ISRJPB) (Part 3 of 4)

```
if (&ZSEL = 'C') goto ENDD
&ZCMDWRK = &Z
IF (&ZCMD ¬= &Z)
  &ZCMDWRK = TRUNC(&ZCMD,'.')
  &ZTRAIL=.TRAIL
  IF (&ZCMDWRK = &Z)
    .MSG = ISRU000
&ZSEL = TRANS (TRUNC (&ZCMD,'.')
  1,'PGM(ISRJB2) PARM((ISRJP01) 1) NEWPOOL'
  2,'PGM(ISRJB2) PARM((ISRJP02) 2) NEWPOOL'
  3,'PGM(ISRJB2) PARM((ISRJP03) 3) NEWPOOL'
  5,'PGM(ISRJB2) PARM((ISRJP05) 5) NEWPOOL'
  6,'PGM(ISRJB2) PARM((ISRJP06) 6) NEWPOOL'
  7,'PGM(ISRJB2) PARM((ISRJP07) 7) NEWPOOL'
 10,'PGM(ISRJB2) PARM((ISRJP10) 10) NEWPOOL'
 12,'PGM(ISRJB2) PARM((ISRJP12) 12) NEWPOOL'
 13,'PGM(ISRJB2) PARM((EDCJP13,ISRJP13) 13) NEWPOOL'
 14,'PGM(ISRJB2) PARM((FANJP14,ISRJP14) 14) NEWPOOL'
 15,'PGM(ISRALTDI) PARM(EVGJP15,,ISRJP15,*) NOCHECK'
 16,'PGM(ISRJB2) PARM((EDCJP16,ISRJP16) 16) NEWPOOL'
 18,'CMD(ISPDTLC (PANEL SUBMIT RETURN)) MODE(FSCR)'
 19,'PGM(ISRALTDI) PARM(CBC3PE5A,+,ISRJP19,*,ISRJP19A) NOCHECK'
 ' ',' '
   *,'?')
ENDD:
)PNTS
FIELD(ZPS01001) VAR(ZCMD) VAL(1)
FIELD(ZPS01002) VAR(ZCMD) VAL(7)
FIELD(ZPS01003) VAR(ZCMD) VAL(15)
FIELD(ZPS01004) VAR(ZCMD) VAL(2)
FIELD(ZPS01005) VAR(ZCMD) VAL(10)
FIELD(ZPS01006) VAR(ZCMD) VAL(16)
FIELD(ZPS01007) VAR(ZCMD) VAL(3)
FIELD(ZPS01008) VAR(ZCMD) VAL(12)
FIELD(ZPS01009) VAR(ZCMD) VAL(18)
FIELD(ZPS01010) VAR(ZCMD) VAL(5)
FIELD(ZPS01011) VAR(ZCMD) VAL(13)
FIELD(ZPS01012) VAR(ZCMD) VAL(19)
FIELD(ZPS01013) VAR(ZCMD) VAL(6)
FIELD(ZPS01014) VAR(ZCMD) VAL(14)
)END
/* 5647-A01 (C) COPYRIGHT IBM CORP 1980, 2000 */
/* ISPDTLC Release: 5.6.  Level: PID                                  */
/* z/OS 01.06.00.  Created - Date: 19 Nov 2003, Time: 12:37           */
```

Figure 63. Batch JCL Generation Panel Definition (ISRJPB) (Part 4 of 4)

The Batch option includes eight suboptions. Each Batch option has an associated panel, CLIST, and skeleton. For option 1, for example, the option panel name is ISRJP01, the CLIST name is ISRJC01, and the skeleton name is ISRJS01. Figure 64 on page 200 shows the Batch Assembler H panel definition.

Note: In Figure 64 on page 200, attribute characters have been replaced by blanks.

```
)PANEL KEYLIST(ISRSAB,ISR)
)ATTR DEFAULT(...) FORMAT(MIX)             /* ISRJP01 - ENGLISH - 5.2 */
 0B TYPE(AB)
 04 TYPE(ABSL) GE(ON)
 05 TYPE(PT)
 09 TYPE(FP)
 0A TYPE(NT)
 0C TYPE(NT) SKIP(ON)
 11 TYPE(SAC)
 12 TYPE(CEF) PADC(USER)
 19 TYPE(DT)
 22 TYPE(WASL) SKIP(ON) GE(ON)
 08 TYPE(CH)
 26 TYPE(NEF) CAPS(ON) PADC(USER)
 27 AREA(SCRL) EXTEND(ON)
 28 TYPE(SAC) CSRGRP(99) RADIO(ON)
)ABC DESC('Menu') MNEM(1)
PDC DESC('Settings') UNAVAIL(ZPM1) MNEM(1) ACC(CTRL+S)
 ACTION RUN(ISRROUTE) PARM('SET')
PDC DESC('View') UNAVAIL(ZPM2) MNEM(1) ACC(CTRL+V)
 ACTION RUN(ISRROUTE) PARM('BR1')
PDC DESC('Edit') UNAVAIL(ZPM3) MNEM(1) ACC(CTRL+E)
 ACTION RUN(ISRROUTE) PARM('ED1')
PDC DESC('ISPF Command Shell') UNAVAIL(ZPM4) MNEM(6) ACC(CTRL+C)
 ACTION RUN(ISRROUTE) PARM('C1')
PDC DESC('Dialog Test...') UNAVAIL(ZPM5) MNEM(8) ACC(CTRL+T)
 ACTION RUN(ISRROUTE) PARM('DAL')
PDC DESC('Other IBM Products...') UNAVAIL(ZPM6) MNEM(1) ACC(CTRL+O)
 ACTION RUN(ISRROUTE) PARM('OIB')
PDC DESC('SCLM') UNAVAIL(ZPM7) MNEM(3) ACC(CTRL+L)
 ACTION RUN(ISRROUTE) PARM('SCL')
PDC DESC('ISPF Workplace') UNAVAIL(ZPM8) MNEM(6) ACC(CTRL+W)
 ACTION RUN(ISRROUTE) PARM('WRK')
PDC DESC('Status Area...') UNAVAIL(ZPMS) MNEM(8) ACC(CTRL+A)
 ACTION RUN(ISRROUTE) PARM('SAM')
PDC DESC('Exit') MNEM(2) PDSEP(ON) ACC(CTRL+X) ACTION RUN(EXIT)
)ABCINIT
.ZVARS=ISR@OPT
)ABC DESC('RefList') MNEM(1)
PDC DESC('Current Data Set List &ZDSCURT') MNEM(1) ACC(CTRL+ALT+P)
 ACTION RUN(ISRRLIST) PARM('PL1')
PDC DESC('Current Library List &ZDSCURLT') MNEM(2) ACC(CTRL+SHIFT+P)
 ACTION RUN(ISRRLIST) PARM('LL1')
PDC DESC('List of Personal Data Set Lists') MNEM(1) PDSEP(ON) ACC(CTRL+ALT+O)
 ACTION RUN(ISRRLIST) PARM('PL2')
PDC DESC('List of Personal Library Lists') MNEM(2) ACC(CTRL+SHIFT+O)
 ACTION RUN(ISRRLIST) PARM('LL2')
```

Figure 64. Batch Assembler H Definition (ISRJP01) (Part 1 of 5)

```
)ABCINIT
.ZVARS=REFLIST
      VGET (ZCURTB ZCURLTB) PROFILE
      IF (&ZCURTB = &Z) &ZDSCURT = &Z
      ELSE &ZDSCURT= '(&ZCURTB)'
      IF (&ZCURLTB = &Z) &ZDSCURLT = &Z
      ELSE &ZDSCURLT= '(&ZCURLTB)'
)ABC DESC('Utilities') MNEM(1)
PDC DESC('Library') UNAVAIL(ZUT1) MNEM(1) ACC(ALT+1)
 ACTION RUN(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(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-ForE') UNAVAIL(ZUT14) MNEM(4) ACC(CTRL+F5)
 ACTION RUN(ISRROUTE) PARM('U15')
)ABCINIT
.ZVARS=PDFUTIL
     &zut9 = '1'
)ABC DESC('Help') MNEM(1)
PDC DESC('Input Data Set') MNEM(1) ACTION RUN(TUTOR) PARM('ISR51001')
PDC DESC('SYSLIB Data Sets') MNEM(1) ACTION RUN(TUTOR) PARM('ISR51002')
PDC DESC('Object Data Set') MNEM(1) ACTION RUN(TUTOR) PARM('ISR51003')
PDC DESC('Listing') MNEM(1) ACTION RUN(TUTOR) PARM('ISR51004')
PDC DESC('Password protection') MNEM(1) ACTION RUN(TUTOR) PARM('ISR51007')
PDC DESC('Assembler selection') MNEM(1) ACTION RUN(TUTOR) PARM('ISR51008')
PDC DESC('Appendices') MNEM(4) ACTION RUN(TUTOR) PARM('ISR00004')
PDC DESC('Index') MNEM(2) ACTION RUN(TUTOR) PARM('ISR91000')
)ABCINIT
.ZVARS=JP1HELP
```

Figure 64. Batch Assembler H Definition (ISRJP01) (Part 2 of 5)

```
)BODY  CMD(ZCMD)
.. Menu. RefList. Utilities. Help.
.------------------------------------------------------------------------------
.                            .Batch Assembler.                                .
.Command ===>.Z                                                               .
.SAREA39                                                                      .
)AREA SAREA39
.ISPF Library:.                                                               .
. ..Project . . ..Z       .
. ..Group . . . ..Z       .. . ..Z       .. . ..Z       .. . ..Z       .
. ..Type  . . . ..Z       .
. ..Member  . . ..Z       ..(Blank or pattern for member selection list).
.
.Other Partitioned or Sequential Data Set:.                                   .
. ..Data Set Name  . ..Z                                                      .
.. .
                                                    .Assembler.
.List ID  . . . . ..Z       ..(Blank for hardcopy). .Z..1..High Level Assembler.
.SYSOUT class . . ..Z              ..(For hardcopy).   .2..Assembler H        .
.
.Assembler options:.                                                          .
...Term  . . ..Z     ..(TERM or NOTERM).
...Other . . ..Z                                                            .
.
.Additional input libraries:.                                                 .
. ..    ===>.Z                                                               .
. ..    ===>.Z                                                               .
. ..    ===>.Z                                                               .
)INIT
.ZVARS = '(ZCMD PRJ1 LIB1 LIB2 LIB3 LIB4 TYP1 MEM DSN LID ZASMOPT BCLA +
          BHASMT BHASM BHAL1 BHAL2 BHAL3)'
.HELP = ISR51000
.ATTR(ZASMOPT)='CSRGRP(99) RADIO(ON)'
&ZUT6  = 1
&ZUT7  = 1
&ZUT11 = 1
&ZUT12 = 1
&ZUT13 = 1
&ZUT14 = 1
&ZMLCSR = '        '                        /*                          @M1A*/
&TYP1 = &ASMT                               /*ASSEMBLER TYPE VARIABLE NAME*/
IF (&DSN ¬= ' ')
  &MEM = ' '                                /*                          @M1A*/
  IF (.CURSOR = ' ')   .CURSOR = DSN
  IF (&ZASMOPT ¬= 1)
    IF (&ZASMOPT ¬= 2)
      &ZASMOPT = 1
IF (&BASMT = ' ')
&BASMT = NOTERM                             /* DEFAULT TO "NOTERM"        */
IF (&ZBTEMP = '')
  &ZBPACK = &ZBPACK
  VPUT (ZBPACK) SHARED
  &ZBTEMP = '0'
IF (&BHALEV = &Z)                                                 /* @OW19891*/
  &BHALEV = 'HLASM'                                               /* @OW19891*/
VGET (BHAL1,BHAL2,BHAL3) PROFILE            /*                      OW22979*/
)REINIT
REFRESH(PRJ1 LIB1 LIB2 LIB3 LIB4 TYP1 MEM DSN BHAL1 BHAL2 BHAL3)
```

Figure 64. Batch Assembler H Definition (ISRJP01) (Part 3 of 5)

```
IF (&ZNXTMSG='ISRT') .CSRPOS = &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*/
IF (.MSG = ISRDS003)                       /*                        @M1A*/
  REFRESH (PRJ1,LIB1,LIB2,LIB3,LIB4,TYP1,MEM,DSN)
)PROC
&ZCSRV = .CURSOR                           /* AUTOTYPE */
&ZCSRP = .CSRPOS                           /* AUTOTYPE */
&ZODSNLN = 0                               /* AUTOTYPE */
&ZODSNMB = &Z                              /* AUTOTYPE */
&ZNAMES='ZCSRV ZCSRP PRJ1 LIB1 LIB2 LIB3 LIB4 TYP1 MEM '
&ZNAMES='&ZNAMES *.&ZODSNLN&ZODSNMB ZCMD'
IF (.CURSOR = DSN, BHAL1, BHAL2, BHAL3)
  &ZODSNLN = 56
  &ZODSNMB = &Z
  IF (.CURSOR = BHAL1) &ZODSNLN = 65
  IF (.CURSOR = BHAL1) &ZODSNMB = '%'
  IF (.CURSOR = BHAL2) &ZODSNLN = 65
  IF (.CURSOR = BHAL2) &ZODSNMB = '%'
  IF (.CURSOR = BHAL3) &ZODSNLN = 65
  IF (.CURSOR = BHAL3) &ZODSNMB = '%'
  &ZNAMES='ZCSRV ZCSRP * * * * * * * &ZCSRV&ZODSNLN&ZODSNMB ZCMD'
PANEXIT((ZNAMES),LOAD,ISRAUTOT)             /* AUTOTYPE */
IF (&ZNXTMSG='ISRT') EXIT                  /* AUTOTYPE */
VER(&ZASMOPT,NONBLANK)
VER(&ZASMOPT RANGE,1,2)
&ZUT6  = 0
&ZUT7  = 0
&ZUT11 = 0
&ZUT12 = 0
&ZUT13 = 0
&ZUT14 = 0
VGET (ZRDSN) SHARED                        /* REFERENCE LIST CODE        */
IF (&ZRDSN ¬= ' ')                         /* IF DATA SET SELECTED       */
  &DSN = &ZRDSN                            /*    PUT DSN VARIABLE        */
  &ZRDSN = ' '                             /*     INTO PANEL             */
  &ZRVOL = ' '                             /*     INTO PANEL             */
VPUT (ZRDSN ZRVOL) SHARED                  /*                            */
  .CURSOR = DSN
  .MSG = ISRDS003                          /*    MSG PENDING             */
VGET (DSALSEL) SHARED                      /*                            */
IF (&DSALSEL ¬= ' ')                       /* IF LIBRARY SELECTED        */
   VGET (DSA1,DSA2,DSA3,DSA4,DSA5,DSA6,DSA7) SHARED
  &PRJ1 = &DSA1                            /*    PUT LIBRARY VARIABLES   */
  &LIB1 = &DSA2                            /*      INTO PANEL            */
  &LIB2 = &DSA3                            /*         .                  */
```

Figure 64. Batch Assembler H Definition (ISRJP01) (Part 4 of 5)

```
  &LIB3 = &DSA4                          /*         .                 */
  &LIB4 = &DSA5                          /*         .                 */
  &TYP1 = &DSA6                          /*         .                 */
  &MEM  = &DSA7                          /*         .                 */
  &DSN = ' '                             /*    BLANK OUT DSN          */
  &DSALSEL = ' '                         /*    CLEAR LIBRARY SELECTION */
  VPUT (DSALSEL) SHARED                  /*                           */
  .CURSOR = MEM
  .MSG = ISRDS003                        /*    MSG PENDING            */
                                         /*                           */
IF (&ZCMD ¬= ' ') .MSG = ISPZ001         /* INVALID COMMAND           */
VER (&LID,NAME)                          /* LIST ID MUST BE VALID NAME */
IF (&ZDSCHK = 'N' )                      /* IF NO SPF CHECK OF DATASET */
  IF (&DSN = ' ' )                       /*  AND SPF LIBRARY SPECIFIED */
    VER (&PRJ1,NB,NAME)                  /*     REQUIRED FIELD        */
    VER (&LIB1,NB,NAME)                  /*     REQUIRED FIELD        */
    VER (&TYP1,NB,NAME)                  /*     REQUIRED FIELD        */
  IF (&DSN ¬= ' ')                       /*                        @M1A*/
    VER(&DSN DSNAMEPQ)
ELSE
  IF (&DSN ¬= ' ')
    VER(&DSN DSNAMEFM)
IF (&ZASMOPT = 1)
  &BHALEV = 'HLASM'
IF (&ZASMOPT = 2)
  &BHALEV = 'HASM'
VER (&BHALEV,NB,LIST,HASM,HLASM)         /* Assembler level     OW10516*/
VER (&BHASMT,NB,LIST,TERM,NOTERM)        /* TERM OR NOTERM REQUIRED    */
IF (&LID = ' ' )                         /* IF HARDCOPY DESIRED SPECIFY*/
  VER (&BCLA,NONBLANK)                   /*  VALID LOCAL SYSOUT CLASS  */
IF (&BHAL1 ¬= ' ')                       /* IF LIB #1 SPECIFIED    @M2A*/
  VER (&BHAL1,DSNAMEPQ)                  /* VERIFY ADDITIONAL LIB1@DSNQ*/
IF (&BHAL2 ¬= ' ')                       /* IF LIB #2 SPECIFIED    @M2A*/
  VER (&BHAL2,DSNAMEPQ)                  /* VERIFY ADDITIONAL LIB2@DSNQ*/
IF (&BHAL3 ¬= ' ')                       /* IF LIB #2 SPECIFIED    @L2A*/
  VER (&BHAL3,DSNAMEPQ)                  /* VERIFY ADDITIONAL LIB2@DSNQ*/
&ASMT = &TYP1                            /* SAVE ASSEMBLER TYPE        */
&ZSEL = 'CMD(%ISRJC01)'                  /* EXECUTE ASSEMBLE CLIST     */
IF (&LID ¬= ' ' )                        /* IF &BCLA NOT REQD          */
  IF (&BCLA = ' ')                       /*  AND NOT SET               */
    &BCLA = *                            /*  DEFAULT TO MSGCLASS       */
                                         /* Begin @L3A                 */
&ZSYSDS1 = &BHAL1                        /* Fill fields for ISRJFSYS   */
&ZSYSDS2 = &BHAL2                        /* to use as input.           */
&ZSYSDS3 = &BHAL3                        /* ZSYSDS? is a qualified     */
&ZSYSCUR1 = 'BHAL1'                      /*     dataset.               */
&ZSYSCUR2 = 'BHAL2'                      /* ZSYSCUR? is were the cursor*/
&ZSYSCUR3 = 'BHAL3'                      /*     is placed on a error.  */
VPUT (ZSYSDS1 ZSYSDS2 ZSYSDS3 ZSYSCUR1 ZSYSCUR2 ZSYSCUR3) SHARED
                                         /* End   @L3A                 */
VPUT (PRJ1,LIB1,LIB2,LIB3,LIB4,ASMT,BCLA,BHASMT,BHASM) PROFILE
VPUT (BHAL1,BHAL2,BHAL3,BHALEV,ZASMOPT) PROFILE
VPUT (PRJ1,LIB1,LIB2,LIB3,LIB4,ASMT,BCLA,BHASMT,BHASM) SHARED
VPUT (DSN,LID,BHAL1,BHAL2,BHAL3,BHALEV) SHARED            /* OW10516*/
)END
/* 5694-A01 (C) COPYRIGHT IBM CORP 1980, 2001                        */
/* ISPDTLC Release: 5.2.  Level: PID                                 */
/* z/OS 01.02.00.  Created - Date: 10 Mar 2001, Time: 14:25          */
```

Figure 64. Batch Assembler H Definition (ISRJP01) (Part 5 of 5)

The names of the ISPF-supplied panels, CLISTs, and skeletons for the Batch processing option are shown in the Table 39 on page 205.

Table 39. ISPF-Supplied Panels, CLISTs, and Skeletons for Batch Option

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

The overall flow of control for Batch processing is shown in Figure 65.

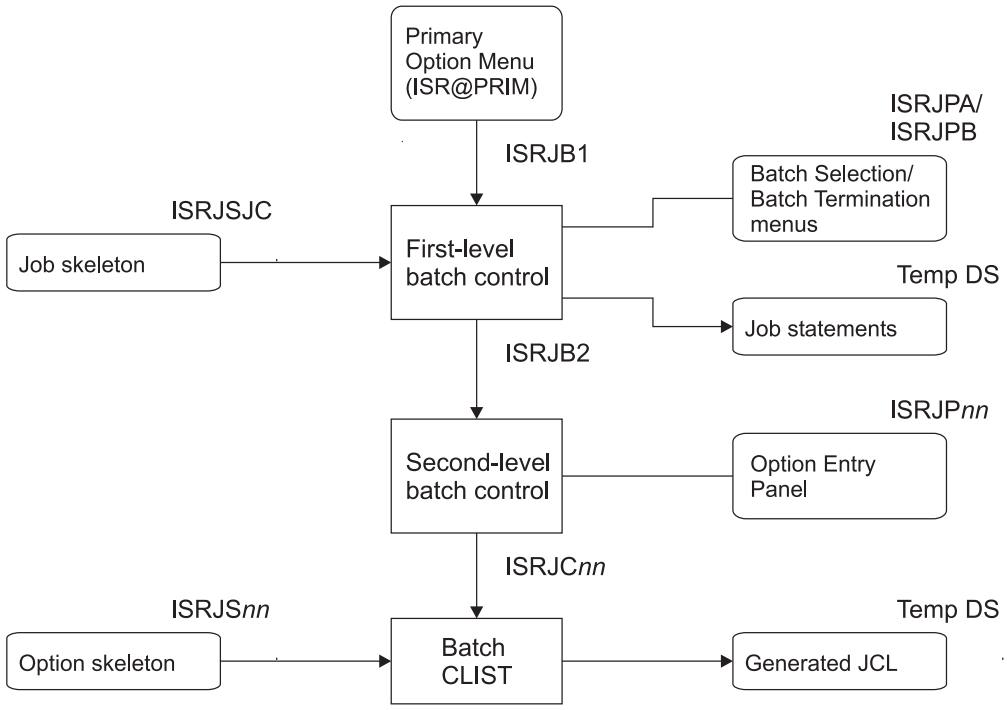


Figure 65. Batch Processing Flow

Two PDF programs control Batch processing: ISRJB1 and ISRJB2. Program ISRJB1 receives control directly from the ISPF Primary Option Menu, ISR@PRIM, through specification of the following 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 177. 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 the following 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, 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 188.

Generated JCL

The job control language (JCL) statements generated by the Batch processing option are accumulated in a temporary data set named "userid.SPFTEMPn.CNTL" or "prefix.userid.SPFTEMPn.CNTL", where *n* is the screen number. This data set contains the job cards generated by ISRJB1 and job steps generated by the option CLISTs. In all cases, file tailoring services are used to generate the JCL. If VIO is used, the generated JCL is in a VIO data set allocated to DDname ISPCTL1.

ISRJB1 generates up to four job statement lines from information that you supply in the Batch Selection panel, ISRJPA. A job statement line is not generated if the corresponding field on the panel is blank. An attempt is made to provide unique jobnames by using the following 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. Refer to the "Batch (Option 5)" chapter 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: The above 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 below.

1. Add the new option (for example, ’99’) to the ISPF Primary Option Menu, panel name ISR@PRIM. In the translated value for option 99, use the PGM keyword to specify that program ISRJB1 is to receive control, and use the PARM keyword to pass the name of the Batch Selection panel. Specify the NOCHECK option to allow ISRJB1 to select subsequent panels. Set the ZTRAIL variable to the value remaining from the truncation function. (ZTRAIL is already set in panel ISR@PRIM, but you must set it yourself if you invoke ISRJB1 from some other selection panel.) For example:

```
)PROC
&ZSEL; = TRANS( TRUNC(&ZCMD,'.'
   . . .

  99,'PGM(ISRJB1) PARM(ZNEW99A) NOCHECK '
   )
&ZTRAIL; = .TRAIL
```

2. Add a new Batch Selection panel (ZNEW99A in this example) to the panel library. Use panel ISRJPA as a model. For each option on the panel, use the PGM keyword to specify that program ISRJB2 is to receive control, and use the PARM keyword to pass the name of the option panel.
3. In the panel definition for ZNEW99A, set the variable RTNPNL to ZNEW99B (as an example) to specify the corresponding Batch Selection panel with Job Step Generated. Add new panel ZNEW99B to the panel library, using panel ISRJPB as a model.
4. Develop new option panels, CLISTs, and skeletons as described in "Steps to Add a New Batch Option to the PDF Batch Selection Panel."

Steps to Add a New Batch Option to the PDF Batch Selection Panel: The steps required to add a new option to the Batch Selection Panel and the panel showing Job Step Generated are listed below.

1. Add the new option (for example, '99') to panels (ISRJPA and ISRJPB). Use the PGM keyword to specify that program ISRJB2 is to receive control and use the PARM keyword to pass the name of the new option panel. For example:

```
)PROC
&ZSEL = TRANS(&ZCMD
  . . .

  99,'PGM(ISRJB2) PARM(ZNEW99 99)'
```

2. Add new option panel ZNEW99 to the panel library. Use one of the distributed option panels (for example, ISRJP01) as a model.
3. Develop a corresponding CLIST (referenced from panel ZNEW99 by the ZSEL variable) and skeleton (referenced from the CLIST by the file tailoring FTINCL service). Use the distributed CLISTs and skeletons (for example, ISRJC01 and ISRJS01) as models. Add the CLIST to a library accessible to ddname SYSPROC, and add the skeleton to the skeleton library.

You can also develop a new option that displays a lower-level selection panel, from which user selections invoke the Batch CLISTs.

For example:

```
)PROC
&ZSEL = TRANS (&ZCMD
   1,'PGM(ISRJB2) PARM(ISRJP01)'
   2,'PGM(ISPJB2) PARM(ISRJP02)'
   . . .

   9,'CMD(ZNEWCMD &ZCMD)'
```

In this example, the CLIST ZNEWCMD has been selected, and the parameter ZCMD (the option entered on the panel) has been passed to the CLIST. ZNEWCMD can then use the SELECT service to display a lower-level selection panel similar to panel ISRJPA, but without job statement information. When the options have been processed, the CLIST should end and return control to ISRJB1 to close the temporary data set and submit the job to TSO. If JCL was generated the invoked dialog (CLIST or PANEL) must end with a return code of 0. If no JCL was

generated the dialog must either end with a return code greater than 0 or set dialog variable ZADARC to a Y in either the ISPF SHARED or PROFILE pool.

Note: If the Source Data Online field, variable ZDSCKO, is set to YES, the dialog variables required for batch option panels are the same as those for foreground processing. That is, variables PRJ1, LIB1, LIB2, LIB3, LIB4, TYP1, MEM, DSN, and ZSEL must be defined in either the)INIT,)BODY, or)PROC section of the panel displayed by ISRJB2.

If the Source Data Online, variable ZDSCKO, is set to NO, these variables are not required. However, if a data set name is not supplied to ISRJB2 in these variables, the output variables ZDSQMEM, ZDSQ, ZDSQ2, ZDSQ3, ZDSQ4, ZDS, ZDS2, ZDS3, ZDS4, ZMEM, and ZORG are blank when control is passed to the CLIST referenced in variable ZSEL in the option panel.

Batch Processing Options Considerations

The SISPLPA data set includes the ISRSCAN load module. It is not executed under TSO but is invoked from batch jobs submitted through the ISPF PDF batch processing option. The ISRSUPC and 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 ISRJS*xxx* 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:

```
//*ISRSCAN step
//SCANSTEP EXEC PGM=ISRSCAN,PARM='member name'
//STEPLIB DD DSN= (Library name if ISRSCAN is not in
       your system library)
```

```
//IN      DD ... Input data set in which the source
       member is found.  Can be a concatenation of up to 8 data sets.
//OUT     DD ... Sequential output data set.
/*
```

Return Codes: ISRSCAN sets one of the following return codes in register 15:

0 Normal completion

8 One of the following:
- 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 below.

```
//*ISRLEMX step
//LEMXSTEP EXEC PGM=ISRLEMX,PARM='parm1,parm2,...,parm15'
//STEPLIB DD DSN= (Library name if ISRLEMX is not in your
       system library)
//ISRLCODE DD ... Input data set in which the source member(s)
       are found.  Can be a concatenation of up to 8 data sets.
//ISRLEXPD DD ... Sequential output data set for expansion.
//ISRLXREF DD ... Sequential output data set for mem prts lst.
//ISRLMSG DD SYSOUT=A /* program error messages print here */

  - or -

//ISRLMSG DD DSN=&ZPREFIX;.&LID;.LMSG,UNIT=SYSDA,
            SPACE=(TRK,(1,1)),DISP=(MOD,CATLG),
            DCB=(RECFM=FBA,LRECL=133,BLKSIZE=3059)
/*
```

Return Codes: ISRLEMX sets one of the following return codes in register 15:

0 Normal completion

n Parameter *n* is 1-15 (too long)

16 Too many parameters

17 Too few parameters

20 Severe error in expand module—an error is printed in the ISRLMSG data set.

Notes:

1. DDNAME ISRLEXPD is needed if parameter 5 is either 'E' or blank. DDNAME ISRLXREF is needed if parameter 5 is 'L'. Refer to the following figure for additional information.
2. ISRLEMX reads data that is presented in a BSAM compatible form. If the SUBSYS parameter (or any parameter that will cause the UCB pointer in the TIOT to be 0) is used, PDF cannot verify that the input is on DASD. It is the responsibility of the user to ensure ISRLEMX will see the data in the correct form. If the data is not presented in a BSAM compatible form, results will be unpredictable.

The ISRLEMX parameter string contains up to 15 parameters, each separated from the next by commas. The parameters are:

Parameter	Length	Description
1	CHAR(3)	Language type of the input member to be processed: ASM - Assembler COB - COBOL FOR - FORTRAN PAS - Pascal PLI - PL/I SCR - Script
2	CHAR(8)	Member name of member to be expanded or of first member to be processed to create a member parts list.
3	CHAR(1)	'B' - The request is being run in batch 'F' - The request is being run in foreground
4	CHAR(1)	'Y' - Allocate a temporary sort data set if needed 'N' - Do not allocate a sort data set
5	CHAR(1)	'E' - Expand and unpack the specified member and all included members into one sequential data set 'L' - Create a member parts list starting with the specified member blank - Copy the specified input member to a temporary data set, unpacking the member but not expanding.
6	CHAR(1)	Number of concatenated input libraries that should be scanned to find the specified input number value can be 1-9
7	CHAR(20)	User trigger, a character string of maximum length 20 to be processed as an INCLUDE, COPY or IMBED statement when found in the member being processed. Enter a comma to skip this parameter if no user trigger is being used.
8	CHAR(2)	User trigger start column, specifies which column the user trigger listed above will start in the member being processed. Enter a comma to skip this parameter if no user trigger is being used.
9	CHAR(3)	Indicates the National language in use: ENU - English DEU - German (Deutsch) KAN - Japanese (Kanji) DES - Swiss German This is used to build the name of the literal load module that is loaded by member expansion. If the language is not English and the load fails, the English table is loaded.
10	CHAR(1)	Position of the month value in the date string; for example, *yy/mm/dd* (default '4')
11	CHAR(1)	Position of the day value in the date string; for example, *yy/mm/dd* (default '7')
12	CHAR(1)	Position of the year value in the date string; for example, *yy/mm/dd* (default '1')
13	CHAR(1)	Delimiter to use in the date string; for example, *yy/mm/dd* (default '/')
14	CHAR(8)	Unit name to be used for all temporary data sets used by ISRLEMX. The unit name must be specified.

Parameter	Length	Description
15	CHAR(4)	The number of blocks used when allocating the temporary sort data sets. These are allocated to ddnames ISRKLWKnn, where nn is 1 through 4. The default for this parameter is 0100. It can be increased for very large or complex expansions.

Note: You can specify the date notation in the national language format (for example, `yy/mm/dd, mm/dd/yy, dd/mm/yy`). Therefore, you must specify the index for each portion. Use any valid character to delimit the date string (for example, `yy/mm/dd, dd.mm.yy`). This delimiter is required.

Adding User-Defined Triggers

During installation time, you can add a user-defined trigger to the member expansion or member parts list functions, or to the foreground or batch options of PDF. The user defined trigger is interpreted as though it were an INCLUDE, COPY, or `.im` keyword.

Adding a User-Defined Trigger to the Member Expansion Function: The user can define the member expansion function trigger by using up to 20 characters. The trigger must start in the user-defined start column and be followed by at least one blank. The next nonblank string is interpreted as the included member name. For more information about member expansion and triggers, see *z/OS ISPF User's Guide Vol II*.

Adding a User-Defined Trigger to the Foreground Options: To define a user trigger for the PDF Foreground options, change the ISRLEMX SELECT statement in the CLIST for the appropriate option.

Change from:

```
ISPEXEC SELECT PGM(ISRLEMX) +
 PARM('XXX,&ZMEM,F,N,E,4, ,00,&ZFPRLANG,&ZFPRMMIX,+
 &ZFPRDDIX,&ZFPRYYIX,&ZFPRDLIM,&Z4UNIT')
```

To:

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

where *nn* is the trigger start column.

The CLIST names are as follows:

```
OPTION 4.1     ISRFC01
OPTION 4.2     ISRFC02
OPTION 4.3     ISRFC03
OPTION 4.5     ISRFC05
OPTION 4.6     ISRFC06
OPTION 4.9     ISRFC09
```

Adding a User-Defined Trigger to the Batch Options: To define a user trigger for the PDF Batch options, change the ISRLEMX EXEC statement in the skeleton for the appropriate option.

Change from:

```
//EXPAND EXEC PGM=ISRLEMX,COND=(12,LE),
//  PARM=('XXX,&ZMEM,B,N,E,4, ,00,&ZJB2LANG,;
//  &ZJB2MMIX,&ZJB2DDIX,&ZJB2YYIX,&ZJB2DLIM,&Z5UNIT')
```

To:

```
//EXPAND EXEC PGM=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:

```
OPTION 5.1    ISRJS01I, ISRJS01J, ISRJS01P, ISRJS01S
OPTION 5.2    ISRJS02I, ISRJS02J, ISRJS02P, ISRJS02S
OPTION 5.3    ISRJS03I, ISRJS03J, ISRJS03P, ISRJS03S
OPTION 5.5    ISRJS05I, ISRJS05J, ISRJS05P, ISRJS05S
OPTION 5.6    ISRJS06I, ISRJS06J, ISRJS06P, ISRJS06S
```

Adding a User-Defined Trigger to the Member Parts List Function: The user can define the trigger for the member parts list function by using up to 20 characters. The trigger must start in the user-defined start column and be followed by at least one blank. The next nonblank string is interpreted as the included member name.

To define the trigger, change two SET statements in the CLISTs ISRFC12 (Foreground) and ISRJC12 (Batch), as follows:

Change from:

```
SET &UT = &STR(&Z)
SET &UTC = &STR(00)
```

To:

```
SET &UT = &STR(trigger)
SET &UTC = &STR(nn)
```

where *nn* is the trigger start column.

Changing the size of the data set for the member parts list function: The member parts list function allocates a temporary data set for collecting cross-reference data. The default size of this data set is 100 blocks each of primary and secondary storage. To change this size, change the SSPACE variable in the panels ISRFP12 (foreground) and ISRJP12 (batch), as follows:

Change from:

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

To:

```
)PROC
&SSPACE = 'nnnn'
```

where *nnnn* is the number of blocks to be allocated.

Customizing Browse and Edit

This section describes how to provide customized Browse and Edit panels, and how to enable users to browse data that is stored in a Unicode format.

Providing customized Browse and Edit panels

Dialog developers can provide customized Browse or Edit data display panels for the dialogs they create. The name of the customized panel is passed as a parameter to the BROWSE or EDIT dialog service.

Note: Do not use the names ISREDDE, ISREDDE2, ISREDDE3, ISREDDE4, or ISREDDE5 as a panel name passed to the EDIT or VIEW services. When ISPF is using any of these panels, it can dynamically switch among any of these panels based on system configuration and edit profile settings.

Customized Browse panels must be patterned after the distributed panel ISRBROB (for Browse, as shown in Figure 66 on page 216). Customized Edit panels must be patterned after the following panels:

ISREFR01 - Edit without action bars or extended highlighting (as shown in Figure 67 on page 217)
ISREFR02 - Edit with action bars and extended highlighting
ISREFR03 - Edit with action bars and no extended highlighting
ISREFR04 - Edit with extended highlighting but no action bars

The customized panel must include a Command field named ZCMD and a dynamic area named ZDATA. As shown in Figure 67 on page 217, 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. Refer to the *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.

```
)ATTR DEFAULT(   ) FORMAT(MIX)            /* ISRBROB - ENGLISH - 5.5 */
 05 TYPE(PT)
 09 TYPE(FP)
 0A TYPE(NT)
 13 TYPE(NEF) PADC(USER)
 16 TYPE(VOI) PADC(USER)
 26 AREA(DYNAMIC) EXTEND(ON) SCROLL(ON)
 01 TYPE(DATAOUT) INTENS(LOW)
 02 TYPE(DATAOUT)
 0B TYPE(DATAOUT) FORMAT(DBCS) OUTLINE(L)
 0C TYPE(DATAOUT) FORMAT(EBCDIC) OUTLINE(L)
 0D TYPE(DATAOUT) FORMAT(&MIXED) OUTLINE(L)
 10 TYPE(DATAOUT) INTENS(LOW) FORMAT(DBCS) OUTLINE(L)
 11 TYPE(DATAOUT) INTENS(LOW) FORMAT(EBCDIC) OUTLINE(L)
 12 TYPE(DATAOUT) INTENS(LOW) FORMAT(&MIXED) OUTLINE(L)
)BODY  EXPAND(//) WIDTH(&ZWIDTH)  CMD(ZCMD)
 BROWSE    Z/ /                                      Line Z        Col Z
 Command ===> Z/ /                                           Scroll ===> Z
 ZDATA/ /
 / /
 / /
)INIT
.ZVARS = '(ZTITLE ZLINES ZCOLUMS ZCMD ZSCBR)'
.HELP = ISR1B000
&ZCMD = ' '
VGET (ZSCBR) PROFILE     /* Fill Scroll Vars if       */
IF (&ZSCBR = ' ')  &ZSCBR  = 'PAGE'  /* Blank with Page */
IF (&ZMEMB ¬= ' ') &ZTITLE = '&ZDSNT(&ZMEMB)&ZLEVEL ' /* OZ91708 */
IF (&ZMEMB = ' ')  &ZTITLE = '&ZDSNT&ZLEVEL '
&MIXED = MIX
IF (&ZPDMIX = N) &MIXED = EBCDIC
)REINIT
REFRESH(ZCMD,ZSCBR,ZDATA,ZLINES,ZCOLUMS,ZTITLE)
)PROC
&ZCURSOR = .CURSOR
&ZCSROFF = .CSRPOS
VPUT (ZSCBR) PROFILE     /*                        */
&ZLVLINE = LVLINE(ZDATA)
)END
/* 5647-A01 (C) COPYRIGHT IBM CORP 1995, 1999 */
```

Figure 66. Browse Panel (ISRBROB)

```
)ATTR /* EDIT PANEL WITH NO ACTION BAR & NO HIGHLIGHTING */
   _ TYPE(INPUT) CAPS(OFF) INTENS(HIGH) FORMAT(&MIXED)
   | AREA(DYNAMIC) EXTEND(ON) SCROLL(ON) USERMOD(20)
   ! TYPE(OUTPUT) INTENS(HIGH) PAD(-)
  01 TYPE(DATAOUT) INTENS(LOW)
  02 TYPE(DATAOUT) INTENS(HIGH)
  03 TYPE(DATAOUT) SKIP(ON) /* FOR TEXT ENTER CMD. FIELD */
  04 TYPE(DATAIN)  INTENS(LOW)  CAPS(OFF) FORMAT(&MIXED)     /*@OY36835*/
  05 TYPE(DATAIN)  INTENS(HIGH) CAPS(OFF) FORMAT(&MIXED)     /*@OY36835*/
  06 TYPE(DATAIN)  INTENS(LOW)  CAPS(IN)  FORMAT(&MIXED)
  07 TYPE(DATAIN)  INTENS(HIGH) CAPS(IN)  FORMAT(&MIXED)     /*@OY36835*/
  08 TYPE(DATAIN)  INTENS(LOW)  FORMAT(DBCS) OUTLINE(L)      /*@OW39183*/
  09 TYPE(DATAIN)  INTENS(LOW)  FORMAT(EBCDIC) OUTLINE(L)    /*@OW39183*/
  0A TYPE(DATAIN)  INTENS(LOW)  FORMAT(&MIXED) OUTLINE(L)    /*@OW39183*/
  0D TYPE(DATAIN)  INTENS(LOW)  CAPS(IN)  FORMAT(&MIXED) COLOR(BLUE)
  20 TYPE(DATAIN)  INTENS(LOW) CAPS(IN) FORMAT(&MIXED)
)BODY WIDTH(&ZWIDTH) EXPAND(//)
%EDIT -----!ZTITLE ------------------/-/------------------%COLUMNS!ZCL
%COMMAND ===>_ZCMD                   / /                    %SCROLL ==
|ZDATA -------------------------------/-/-------------------------------
|                                    / /
| ----------------------------------/-/-------------------------------
)INIT
  IF (&ZVMODET = 'VIEW')   /* VIEW MODE                */
    .HELP = ISR10000       /* DEFAULT TUTORIAL NAME    */
  ELSE                     /* EDIT MODE                */
    .HELP = ISR20000       /* DEFAULT TUTORIAL NAME    */
  .ZVARS = 'ZSCED'         /* SCROLL AMT VARIABLE NAME */
  &MIXED = MIX             /* SET FORMAT MIX           */
  IF (&ZPDMIX = N)         /* IF EBCDIC MODE REQUESTED */
    &MIXED = EBCDIC        /*  SET FORMAT EBCDIC       */
  VGET (ZSCED) PROFILE     /* FILL SCROLL VARS IF      */
  IF (&ZSCED = ' ')        /* BLANK WITH PAGE.         */
   &ZSCED = 'PAGE'         /*                          */

)REINIT
  REFRESH(ZCMD,ZSCED,ZDATA,ZTITLE,ZCL,ZCR)
  IF (&ZVMODET = 'VIEW')   /* VIEW MODE                */
    .HELP = ISR10000       /* DEFAULT TUTORIAL NAME    */
  ELSE                     /* EDIT MODE                */
    .HELP = ISR20000       /* DEFAULT TUTORIAL NAME    */

)PROC
  &ZCURSOR = .CURSOR
  &ZCSROFF = .CSRPOS
  &ZLVLINE = LVLINE(ZDATA)
  VPUT (ZSCED) PROFILE

)END

/*********************************************************************/
/* Use variable ZDSVOL to display the volume of the data set        */
/*********************************************************************/
/*  DYNAMIC AREA SCREEN WIDTH FROM PQUERY. (80,132,160)
/*  DYNAMIC AREA SCREEN DEPTH FROM PQUERY. (24,32,43,27,60)
/*                                                                   */
/* 5645-001, 5655-042 (C) COPYRIGHT IBM CORP 1980, 1996              */
```

Figure 67. Edit Panel (ISREFR01)

The distributed panels have a dynamic area that automatically fills the available width and depth of the screen (after the first two lines). A customized panel can have a dynamic area that is fixed in width (in which case the WIDTH and EXPAND keywords should be omitted from the)BODY header statement), fixed in

depth (in which case the EXTEND keyword should be omitted on the AREA statement in the)ATTR section), or fixed in both width and depth.

Note: If the dynamic area is less than the full width of the screen, the panel definition must include attribute bytes for protected fields on either side of the area.

Refer to 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 above) 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 40 and Table 41 on page 219 list the ISPF-provided output-only variables you can use on Browse and Edit panels.

Table 40. Browse Output-Only Variables

Variable	Description	Format/Length
ZCOLUMS	First and last columns being displayed	CHAR 7
ZCSROFF	Cursor offset within cursor field	CHAR 4
ZCURSOR	Name of field where cursor is placed	CHAR 8
ZDADWD	Width of the dynamic area (ZDATA)	CHAR 4
ZDAMLN	Length of the dynamic area (ZDATA)	CHAR 4
ZDSN	Name of data set being displayed	CHAR 44
ZLEVEL	Version and mod level of member	CHAR 8
ZLINES	Top line of the data display	CHAR 8
ZLVLINE	Last visible line of ZDATA after last interaction	CHAR 4
ZMEMB	Name of member being displayed	CHAR 8

Table 40. Browse Output-Only Variables (continued)

Variable	Description	Format/Length
ZTITLE	The title line DSN(MEMB) - VER.MOD	CHAR 65
ZWIDTH	Width of the panel	CHAR 4
ZDSVOL	Volume of the data set or first library in the concatenation	CHAR 6

Table 41. Edit Output-Only Variables

Variable	Description	Format/Length
ZCL	Left column of the data display	CHAR 5
ZCR	Right column of the data display	CHAR 5
ZCSROFF	Cursor offset within cursor field	CHAR 4
ZCURSOR	Name of field where cursor is placed	CHAR 8
ZDADLN	Length of the dynamic area (ZDATA)	CHAR 4
ZDADWD	Width of the dynamic area (ZDATA)	CHAR 4
ZDSN	Name of data set being displayed	CHAR 44
ZLEVEL	Version and mod level of member	CHAR 8
ZLVLINE	Last visible line of ZDATA after last interaction	CHAR 4
ZMEMB	Name of member being displayed	CHAR 8
ZTITLE	The title line DSN(MEMB) -VER.MOD	CHAR 65
ZWIDTH	Width of the panel	CHAR 4
ZDSVOL	Volume of the data set or first library in the concatenation.	CHAR 6

Enabling Browse panels to display Unicode data

You can enable users to browse data that is stored in a Unicode format. To do this, MVS Conversion Services must first be set up for the appropriate conversions. See *z/OS Support for Unicode: Unicode Services*.

Customizing Member List Panels

Dialog developers can also provide customized member list display panels for their dialogs. The name of the customized member list display panel is passed as a parameter to the LMMDISP service.

A customized member list panel should be modeled after the LMMDISP default member list panel, ISRML000.

Note: Panel ISRML000 is 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, 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 SISPG*xxx* libraries, where *xxx* is the designator for a specified language.

The customized panel must include a Command field named ZCMD and a dynamic area named ZDATA. In addition, for customized member list panels, the dynamic area must be 80 characters wide. The Scroll field is optional. The name of

the Scroll field on member list panels is ZSCML. This same name can be used on a customized member list panel. Inclusion of the variables that output the data set name, relative row number, and total number of rows is optional. If you want to display the volume of the data set you can use variable ZDSVOL. Customized member list panels should also meet the requirements for a movable Command line. Figure 68 shows an example of a customized member list panel that was generated by Dialog Tag Language code.

Refer to the *z/OS ISPF Dialog Developer's Guide and Reference* for a discussion of these needs.

```
)PANEL KEYLIST(ISRSAB,ISR)
)ATTR DEFAULT(|¬–) FORMAT(MIX)
 0B TYPE(AB)
 0D TYPE(PS)
 2E TYPE(ABSL) GE(ON)
 2F TYPE(PT)
 29 TYPE(FP)
 0A TYPE(NT)
 13 TYPE(NEF) PADC(USER)
 16 TYPE(VOI) PADC(USER)
 26 TYPE(VOI) CAPS(ON) PADC(USER)
 27 AREA(DYNAMIC)
 08 TYPE(DATAOUT) PAS(ON)
 09 TYPE(DATAOUT)
 2A AREA(DYNAMIC) EXTEND(ON) SCROLL(ON)
 01 TYPE(DATAIN) CAPS(ON) PADC(&ZMLPAD)
 02 TYPE(DATAOUT) INTENS(&MLI2) COLOR(&MLC2) HILITE(&MLH2)
 03 TYPE(DATAIN) CAPS(ON)
 04 TYPE(DATAOUT) INTENS(&MLI3) COLOR(&MLC3) HILITE(&MLH3)
 05 TYPE(DATAOUT)
 06 TYPE(DATAOUT) INTENS(LOW)
 15 TYPE(NEF) CAPS(ON) PADC(USER)
)ABC DESC('Functions') MNEM(1)
PDC DESC('Save') MNEM(1) ACTION RUN(SAVE)
PDC DESC('Change Colors...') MNEM(1) ACTION RUN(MLC)
)ABCINIT
  .ZVARS=MEMOPT

)ABC DESC('Help') MNEM(1)
PDC DESC('General') MNEM(1) ACTION RUN(TUTOR) PARM('ISR01130')
PDC DESC('Scrolling') MNEM(1) ACTION RUN(TUTOR) PARM('ISR01131')
PDC DESC('Pattern matching') MNEM(1) ACTION RUN(TUTOR) PARM('ISR01232')
PDC DESC('LOCATE command') MNEM(1) ACTION RUN(TUTOR) PARM('ISR01132')
PDC DESC('SORT command') MNEM(2) ACTION RUN(TUTOR) PARM('ISR01226')
PDC DESC('SAVE command') MNEM(2) ACTION RUN(TUTOR) PARM('ISR01229')
PDC DESC('RESET command') MNEM(1) ACTION RUN(TUTOR) PARM('ISR01138')
PDC DESC('SELECT command') MNEM(2) ACTION RUN(TUTOR) PARM('ISR01133')
PDC DESC('S line command') MNEM(4) ACTION RUN(TUTOR) PARM('ISR01134')
PDC DESC('Statistics') MNEM(2) ACTION RUN(TUTOR) PARM('ISR01140')
PDC DESC('Appendices') MNEM(5) ACTION RUN(TUTOR) PARM('ISR00004')
PDC DESC('Index') MNEM(3) ACTION RUN(TUTOR) PARM('ISR91000')

)ABCINIT
  .ZVARS=MEMLHELP
```

Figure 68. Customized Member List Panel (Part 1 of 2)

```
)BODY  CMD(ZCMD)

+
   Functions  Help%-------------------------------------------------------------
%MEMBER LIST -- &ZDSN ----------------------------------------%ROW ZMLCR%OF ZMLTR+
%COMMAND ===>_ZCMD                                            %SCROLL ===>_Z   +
'ZMLCOLD                                                                       '
'ZDATA                                                                         '
'                                                                              '
)INIT
  .ZVARS = '(ZDSN ZMLCR ZMLTR ZCMD ZSCML)'
  .HELP = ISR01130
  .ATTR(ZMLCR)='JUST(RIGHT) PAD('0'')'
  .ATTR(ZMLTR)='JUST(RIGHT) PAD('0'')'
VGET (MLC1 MLC2 MLC3 MLH1 MLH2 MLH3 ZMLPD) PROFILE

IF (&ZGUI = ' ' AND &ZMLPD = ' ')
  &ZMLPAD = '_'
ELSE
  &ZMLPAD = 'USER'
&ZSCKEY = 'ZFRML'
IF (&MLC1 = ' ')
  &MLC1 = 'TURQ'
IF (&MLC2 = ' ')
  &MLC2 = 'BLUE'
IF (&MLC3 = ' ')
  &MLC3 = 'GREEN'

VGET (ZSCML) PROFILE     /* Fill Scroll Vars if       */
IF (&ZSCML = ' ')        /* Blank with page.          */
 &ZSCML = 'PAGE'         /*                           */
)PROC
VPUT (ZSCML) PROFILE
IF (.CURSOR = ZDATA OR .CURSOR = ZMLCOLD)
  &ZMSCPOS = &ZCURPOS;
ELSE
  &ZMSCPOS = '0000'
/*********************************************************************/
/* Use variable ZDSVOL to display the volume of the data set        */
/*********************************************************************/
)PNTS
)END
/* 5655-042 (C) COPYRIGHT IBM CORP 1980, 1995 */
/* ISPDTLC Release: 4.3.  Level: PID                                 */
/* OS/390 R.3.  Created - Date: 14 Oct 1996, Time: 11:18:33          */
```

Figure 68. Customized Member List Panel (Part 2 of 2)

As with edit and browse customized panels, the distributed member list panels have a dynamic area that extends to fill the available depth of the screen. But, unlike edit and browse panels, (the dynamic area on a member list panel does not expand to fill the width of the screen), it must be 80 characters wide. The dynamic area can also be fixed in length by removing the EXTEND keyword from the)ATTR section of the panel definition.

Refer to the description of dynamic areas in the *z/OS ISPF Dialog Developer's Guide and Reference*.

See "Customizing Browse and Edit" on page 214 for a more complete discussion on customizing panels.

Table 42 lists the ISPF-provided output-only variables you can use on member list panels.

Table 42. Member List Panel Output-Only Variables

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

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 the following 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 on the following page.

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
  ```

PDF Installation-Wide Exits

PDF allows installations to satisfy unique processing needs by providing installation-wide exits at the following 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

Programming Interface information

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 "Disabling Generic High-Level Qualifiers" on page 91 for an example of a data set list filter exit.

Analyze Dsname and Volume

Programming Interface information

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

Analyze Data Set Name

Programming Interface information

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

Return Codes

Programming Interface information

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

Programming Interface information

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

Programming Interface information

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 the following 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.

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 43 on page 226.

Table 43. Parameter List During Allocation

Key	#Parms	Length	Parameter	Description	Exit may modify these parameters
0001	1	8	DDNAME	DDNAME to allocate data set to	No
0002	1	44	DSNAME	Data set name to allocate	No
0004	1	1	STATUS	Data set status	Yes
0005	1	1	NDISP	Normal data set disposition	Yes
0006	1	1	CDISP	Conditional data set disposition	Yes
0010	1	6	VLSER	Volume serial	No
0015	1	8	UNIT	unit name	No
0050	1	8	PASSW	Password for protected data set if a password was specified	No
0057	1	8	RTORG	Return data set organization	No
005D	1	6	RTVOL	Return volume serial from allocation	No

If the request is to *concatenate* a data set, the parameter list contains the information in Table 44.

Table 44. Parameter List During Concatenation

Key	#Parms	Length	Parameter	Description	Exit may modify these parameters
0001	1	8	DDNAMEx	List of DDNAMES corresponding to the data sets being concatenated	No
0004	0	0	PERMCC	Permanently concatenated attribute	No

If the request is to *create* a data set, the parameter list contains the information in Table 45.

Table 45. Create Data Set Allocation Parameter List

Key	#Parms	Length	Parameter	Description	Modifiable?
0001	1	8	DDNAME	ddname to allocate data set to	No
0002	1	44	DSNAME	Data set name to allocate	No
0004	1	1	STATUS	Data set status	Yes
0005	1	1	NDISP	Normal data set disposition	Yes
0006	1	1	CDISP	Conditional data set disposition	Yes
0007	1	0	TRKS	Space allocated in tracks or	Yes
0008	1	0	CYLS	Space allocated in cylinders,	Yes
0009	1	3	BLKS	or Space allocated in blocks	Yes
000A	1	3	PSPACE	Primary space quantity	Yes

Table 45. Create Data Set Allocation Parameter List (continued)

Key	#Parms	Length	Parameter	Description	Modifiable?
000B	1	3	SSPACE	Secondary space quantity	Yes
000C	1	3	DBLKS	Number of directory blocks	Yes
0010	1	6	VOLSER	Volume serial	Yes
0015	1	8	UNIT	Unit group (esoteric) name Device type Specific unit address	Yes
0020	1	1	PASPR	Data set is password protected: X'10' - Data set cannot be read, changed, extended, or deleted. X'30' - Data set can be read, but not changed, extended, or deleted.	Yes
0022 006D 0023	1 1 1	5 7 2	EXPDT EXPDTL RETPD	Expiration date - YYDDD or Expiration date - YYYYDDD or Retention period	Yes Yes Yes
0030	1	2	BLKSZ	Block size	Yes
003C	1	2	DSORG	Data set organization	Yes
0042	1	2	LRECL	Logical record length	Yes
0049	1	1	RECFM	Record format	Yes
0050	1	8	PASSWORD	Data set password if specified	Yes
0052	0	0	PERM	Permanently allocated attribute	No
0057	1	2	RTORG	Return data set organization	No
005D	1	6	RTVOL	Return volume serial from allocation	No
8004	1	8	STORCLAS	Storage class used to allocate the data set	Yes
8005	1	8	MGMTCLAS	Management class used to allocate the data set	Yes
8006	1	8	DATACLAS	Data class used to allocate the data set	Yes
8010	1	1	AVGREC	Unit of allocation in terms of average record size: X'80'–Allocation unit is bytes. X'40'–Allocation unit is kilobytes. X'20'–Allocation unit is megabytes.	Yes
8012	1	1	DSNTYPE	Data set name type used to allocate the data set: X'80'–Library. X'40'–PDS. X'10'–HFS. X'08'–Extreq. X'04'–Extpref. X'02'–Basic. X'01'–Large.	Yes

If the request is to *deallocate* a data set, the parameter list contains the information in Table 46 on page 228.

Table 46. Parameter List During Deallocation

Key	#Parms	Length	Parameter	Description	Exit may modify these parameters
0001	1	8	DDNAME	DDNAME to free	No
0007	0	0	UNALC	Unallocate option	No

If the request is to *delete* a data set, the parameter list contains the information in Table 47.

Table 47. Parameter List During Deletion

Key	#Parms	Length	Parameter	Description	Exit may modify these parameters
0002	1	44	DSNAME	DSNAME to free	No
0007	0	0	UNALL	Unallocate option	No

If the request is to *delete* a data set, it is done through the MVS SCRATCH macro. This allows unexpired data sets to be deleted.

Return Codes

Programming Interface information

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) the following 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

Programming Interface information

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 the following 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

Programming Interface information

PDF uses the following parameters to communicate with the monitoring exit.

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

Usage Notes

Programming Interface information

- 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 an write to the data set as needed. The ISPF termination exit could close and free the data set.

Exit 1: BROWSE Primary Command Start

Programming Interface information

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

Programming Interface information

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

Programming Interface information

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

Programming Interface information

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

Programming Interface information

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

Programming Interface information

This exit point marks the end of a user-entered EDIT primary command. The exit routine is given command immediately after the macro is performed. It can be used together with the EDIT macro command start exit point for monitoring purposes.

Exit 7: EDIT Line Command Start

Programming Interface information

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

Programming Interface information

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

Programming Interface information

This exit point marks the start of a sub-function from either the Library (option 3.1) or Data Set (option 3.2) Utilities. The following are the sub-function identifiers and their descriptions:

A Allocate data set
C Catalog data set
D Delete data set
I Data set information
M Display member list
MB Browse member
MD Delete member
MP Print member
MR Rename member
P Print entire data set

R Rename data set
RU Rename data set using ISPF option 3.2 (catalog is not updated)
S Data set information (short)
U Uncatalog data set
X Print data set index listing
Z Compress data set.

The exit routine is given control immediately before the sub-function is performed.

If multiple line commands (member list) are entered, the exit routine is given control once for each sub-function.

Exit 10: Library/Data Set Utility Sub-function End

Programming Interface information

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

Programming Interface information

This exit point marks the start of a line command from the Data Set List (option 3.4) Utility. The following are the subfunction identifiers and their descriptions:

Identifier	Description
B	Browse data set
C	Catalog data set
D	Delete data set
E	Edit data set
F	Free unused space
I	Data set information
M	Display member list
MB	Browse member
MD	Delete member
ME	Edit member
MP	Print member
MR	Rename member
P	Print entire data set
R	Rename data set
S	Data set information (short)
U	Uncatalog data set
X	Print data set index listing
Z	Compress data set.

In addition to the one character command name, the name of the data set is provided for all commands that operate on entire data sets. For example, when an R is typed next to data set 'MY.TEST.DATA', the command pointed to by the command parameter is:

```
R 'MY.TEST.DATA'
```

Note that the data set name provided is the one shown in the data set list, even if the name has been changed by a data set name change exit.

For the M (member list) command, the command parameter includes the data set name. However, for commands executed within the member list, just the command code is given (such as MB for the B (browse) line command).

The exit routine is given control immediately before the command is performed.

If multiple line commands are entered, the exit routine is given control once for each line command. Only built-in commands cause the exit to be called.

Exit 12: Data Set List Utility Line Command End

Programming Interface information

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

Programming Interface information

If a nonzero return code is returned from an exit call, a message is conditionally set.

Return Codes

Programming Interface information

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

Programming Interface information

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

Programming Interface information

ISPF uses the following parameters to communicate with the line command exit routine.

Parameter	Type	Len	Description	Modifiable
LINE COMMAND	CHAR	255	A 255-line character field containing the data set list line command. The field is left-justified with trailing blanks.	Yes

Usage Notes

Programming Interface information

- 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 the following 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

Programming Interface information

If a nonzero return code is returned from an exit call, a message is conditionally set.

Return Codes

Programming Interface information

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

Programming Interface information

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 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

Programming Interface information

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.

The variables available to the print exit routine are:

Variable	Type	Len	Description	Modifiable
ZPRDSN	CHAR	44	Fully qualified data set name (no quotes) of the data set to be printed	No
ZPRMEM	CHAR	8	Member to be printed, if the data set is a partitioned data set and a member is defined	No
ZPRPMD	CHAR	5	Print mode - BATCH, LOCAL, or TSO/E ICF	Yes
ZPROPT	CHAR	1	Disposition of the data set as specified by the user - K (keep) or D (delete)	Yes
ZPRDSORG	CHAR	8	Organization of the data set - PO or PS	No
ZPRRECFM	CHAR	6	Record format of the data set	No
ZPRLRECL	CHAR	5	Logical record length of the data set	No
ZPRBLKSZ	CHAR	5	Physical block size of the data set	No
ZPRVOLSE	CHAR	6	Volume serial of the data set	No
ZPRPASS	CHAR	8	Data set password (if it is password protected)	No
ZPRSYSO	CHAR	80	Batch SYSOUT class as specified by the user	Yes
ZPRSYSOL	CHAR	80	Local SYSOUT class as specified by the user	Yes
ZPRLPRT	CHAR	8	Local printer ID as specified by the user	Yes
ZPRICFPL	CHAR	15	TSO/E Information Center Facility printer location as specified by the user	Yes
ZPRICFPF	CHAR	8	TSO/E Information Center Facility printer format as specified by the user	Yes
ZPRICFPT	CHAR	8	TSO/E Information Center Facility printer type as specified by the user	Yes
ZPRICFNC	CHAR	3	TSO/E Information Center Facility number of copies as specified by the user	Yes
ZPRICFEF	CHAR	1	TSO/E Information Center Facility enable font selection	Yes
ZPRJB1	CHAR	72	Print job card 1 as specified by the user	Yes
ZPRJB2	CHAR	72	Print job card 2 as specified by the user	Yes
ZPRJB3	CHAR	72	Print job card 3 as specified by the user	Yes
ZPRJB4	CHAR	72	Print job card 4 as specified by the user	Yes

Return Codes

0 Successful termination of print routine. PDF does no further processing.

2 Successful termination of print routine. PDF submits any generated JCL.

4 Successful termination of print routine. PDF prints the data according to the information in the dialog variables located in the ISPF shared pool. If no message is waiting to be displayed, PDF issues a message.

20 Installation print routine failed. PDF keeps the data and displays a severe error panel with the print routine's error message.

Error Processing

Programming Interface information

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. 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 48 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.

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.

The variables available to the print exit routine are:

Table 48. Print Exit Routine Variables

Variable	Type	Len	Description	Modifiable
ZPRLGLST	CHAR	4	Value is LOG if processing the LOG, LIST if processing the LIST.	No
ZPRLGDSN	CHAR	44	Fully qualified log data set name (no quotes) of the data set to print.	No
ZPRLSDSN	CHAR	44	Fully qualified list data set name (no quotes) of the data set to print.	No
ZPRLGPMD	CHAR	5	Log print mode, BATCH or LOCAL.	Yes
ZPRLSPMD	CHAR	5	List print mode, BATCH or LOCAL.	Yes

Table 48. Print Exit Routine Variables (continued)

Variable	Type	Len	Description	Modifiable
*ZPRLGOPT	CHAR	1	Disposition of the log data set as specified by the user D (delete)	Yes
*ZPRLSOPT	CHAR	1	Disposition of the list data set as specified by the user D (delete)	Yes
*ZPRRECFM	CHAR	6	Record format of the list data set.	No
*ZPRLRECL	CHAR	5	Logical record length of the list data set.	No
*ZPRLGSYS	CHAR	15	Batch SYSOUT class for the log data set as specified by the user.	Yes
ZPRLGSYSL	CHAR	15	Local SYSOUT class for the log data set as specified by the user	Yes
*ZPRLSSYS	CHAR	15	Batch SYSOUT class for the list data set as specified by the user	Yes
ZPRLSSYL	CHAR	15	Local SYSOUT class for the list data set as specified by the user	Yes
*ZPRLGPRT	CHAR	17	Local printer ID for the log data set as specified by the user.	Yes
*ZPRLSPRT	CHAR	17	Local printer ID for the list data set as specified by the user.	Yes
*ZPRJB1	CHAR	72	Print job card 1 as specified by the user	Yes
*ZPRJB2	CHAR	72	Print job card 2 as specified by the user	Yes
*ZPRJB3	CHAR	72	Print job card 3 as specified by the user	Yes
*ZPRJB4	CHAR	72	Print job card 4 as specified by the user	Yes

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 the following 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 other JCL needed to print the job. ISPF submits the job.

Local Exit provides the necessary information to print the job. Exit issues the print command and deals with the disposition of the log and list data sets.

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: For return codes 0, 2, and 4, ISPF issues a message regarding the disposition of the data set. The message is based on the dialog variables ZPRLGOPT, ZPRLSOPT, ZPRLGPRT, and ZPRLSPRT located in the ISPF shared pool. If ZPRLGOPT or ZPRLSOPT values are anything other than D (delete) or K (keep), no message is issued.

20 The print routine failed. ISPF keeps the data and displays an error message.

Error Processing

Programming Interface information

The exit routine is responsible for handling all errors that occur while it is in control.

Compress Request Exit

Programming Interface information

In the ISPF configuration table keyword file, this exit is set with the keywords COMPRESS_UTILITY_PROGRAM_EXIT (if your exit is a program) and COMPRESS_UTILITY_CLIST_EXIT (if your exit is a CLIST).

Instead of using the IEBCOPY system utility interface that PDF provides, you can specify that PDF use an installation-written exit routine to handle requests to compress partitioned data sets under options 3.1, 3.4, and the LMCOMP service. When the compress exit gets control, the data set will have already been allocated and enqueued as exclusive. The exit will have to re-allocate the data set to a specific DDNAME. As part of termination processing, the exit must free the allocated DDNAME. The FREE operation must not be done by data set name. PDF allows the exit routine to:

- Use an alternate compression technique
- Provide backup before allowing normal PDF compression
- Selectively prevent compression.

PDF calls the exit routine with the SELECT service. The routine can be either a program or CLIST. All ISPF and system services are available.

Exit Parameters

Programming Interface information

PDF communicates with the exit routine through variables in the shared pool. PDF

considers these variables to be read-only and ignores any changes the exit routines make to the parameters. A VERASE is done at the end of the compress to delete the variables from all variable pools.

The variables available to the compress exit routine are:

Variable	Type	Len	Description
ZCMPDSN	CHAR	44	fully qualified data set name with no quotes
ZCMPVOL	CHAR	6	Volume serial
ZCMPPSWD	CHAR	8	Data set password
ZCMPORIG	CHAR	8	Origin of the call. **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

Programming Interface information

0 Data set successfully compressed
2 PDF should compress the data set
4 Data set is not eligible for compression
8 Data set could not be allocated
12 Data set not cataloged
16 Error in exit; PDF should continue processing
17 LMCOMP recursion error
18 Unknown return code from exit
19 Error encountered invoking exit
20 Severe error; PDF should not continue processing
Other PDF treats the error as return code 16.

Error Processing

Programming Interface information

The exit routine is responsible for handling all errors that occur while it is in control. Failure to use correct allocation, serialization, and deallocation causes unpredictable results.

Data Set Name Change Exit

Programming Interface information

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

Programming Interface information

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 the following parameters to communicate with the data set name change exit:

Variable	Type	Len	Description
PROJECT	CHAR	8	Project name as it was entered on the panel
GROUP	CHAR	8	Group name as it was entered on the panel
TYPE	CHAR	8	Type name as it was entered on the panel
MEMBER	CHAR	8	Member name as it was entered on the panel
OTHER DATA SET	CHAR	56	On input to the exit, the other data set name as entered on the panel. On output from the exit, blank or the data set name to be used. This can be used to replace an ISPF library.

Variable	Type	Len	Description
REASON	CHAR	8	Reason the data set is being allocated: **RECOVERY** The data set is the edit recovery data set. The Other Data Set field contains the fully qualified data set name. The data set name change exit is not called for edit recovery data sets which are listed in the edit recovery table with an associated disposition of 'K'. **TEMP** The data set is a temporary PDF data set. This includes the following: • The data set name specified on the member list SAVE command • The data set name specified on the data set list SAVE command • The data set used during the outlist utility processing (ISPF option 3.8, when using program ISRUOLP instead of CLIST ISRUOL). **blank** • Any data set entered on an PDF data entry panel • Data sets specified on the LMINIT service • Data sets specified on any service that does not need an LMINIT to have been previously performed.

Any of the parameters can be modified and are picked up by PDF with the exception of the Reason field, which is for the information of the exit only. Both an ISPF library (a PROJECT/GROUP/TYPE combination) and an OTHER DATA SET can be present in the parameter list. In this case, the OTHER DATA SET name supersedes the ISPF library name. If an ISPF library combination is specified as input to the exit and the exit wishes to create a new name that does not fit the ISPF library naming convention, the ISPF library name can be replaced by the exit filling in the Other Data Set field.

After the data set name has been changed by the exit, PDF does its normal data set processing, such as appending the user's prefix if an OTHER DATA SET name without quotes is returned by the exit. The changed data set name values are not saved in any variable pool, but are saved internally by PDF. The original input fields on the data entry panel are not changed, but any title line on a panel that contains a data set name can reflect either the original or the changed data set name.

Usage Notes

Programming Interface information

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 following:
   - 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

Programming Interface information

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 situation in which the member list filter exit routine would be invoked, it is called first to allow verification/modification of the specified pattern, and then once for each member in the data set that matches the pattern. The exit routine, through return codes, dictates which members are to be included in the member list, or selected if the SELECT primary command is issued.

PDF invokes the member list filter exit routine using standard linkage conventions. The exit routine must be a program.

From the exit routine, any of the PDF or ISPF services can be invoked. However, you should be careful when invoking services that generate a member list as part of their internal processing (LMMLIST, LMMDISP, LMMOVE, LMPROM, LMPRINT, and LMMSTATS). These invocations would result in a recursive call to the exit routine.

Analyze Patterns

Programming Interface information

Exit Parameters:

Programming Interface information

PDF uses the following parameters to communicate with the exit routine when it is invoked to verify/modify the pattern that you entered either on a member list entry panel, as a parameter to an PDF service, or with the SELECT primary command.

Parameter	Type	Len	Description	Modifiable
CODE	FIXED	4	Code that indicates that this is the analyze pattern call. Call=1.	No
PATTERN	CHAR	8	Pattern as entered by the user	Yes

Return Codes:

Programming Interface information

The following return codes are expected from the member list filter exit routing to indicate the described conditions.

0 The pattern that you entered generates the member list.

4 The pattern is updated by the member list exit program and generates the member list.

8 The member list request or SELECT command request is canceled and a conditional SETMSG is issued.

Analyze Member Names

Programming Interface information

Exit Parameters:

Programming Interface information

PDF uses the following parameters to communicate with the exit routine when the routine determines which members that matched the pattern are to be included in the member list, or which members are to be selected if a SELECT command was issued.

Parameter	Type	Len	Description	Modifiable
CODE	FIXED	4	Code indicating that this is the analyze member name call. Call=2.	No
MEMNAME	CHAR	8	Member name.	No

Return Codes:

Programming Interface information

The following return codes are expected from the member list filter exit program to indicate the described conditions.

0 Include this member in the member list.
4 Do not include this member in the member list.
8 The list is stopped; no more items are added to the list.

Member List Built-in Line Command Exit

Programming Interface information

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

Programming Interface information

PDF uses the following parameters to communicate with the line command exit routine. These parameters are passed as a single 65-character string. A single blank is placed between parameters.

Parameter	Type	Len	Description	Modifiable
LINE COMMAND	CHAR	1	A 1-character field containing the data set list line command.	No
DATA SET NAME AND MEMBER	CHAR	56	A 56-character field containing data set name and member against which the line command is run. The format of this field is: 'DATA.SET.NAME(MEMBER)'. The field is left-justified with trailing blanks.	No
VOLUME SERIAL	CHAR	6	A 6-character field containing the volume on which the data set specified in DATA SET NAME AND MEMBER can be found.	No

Usage Notes

Programming Interface information

- 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

Programming Interface information

If a nonzero return code is returned from an exit call, a message is conditionally set.

Return Codes

Programming Interface information

0 Use line command as specified by the user

8 Do not invoke this built-in command

20 Severe error occurred.

Appendix A. ISPF Enqueue Processing for Data Integrity

This appendix describes the enqueue processing ISPF performs to maintain the integrity of data sets, PDS members, member queues, workstation files, and the SCLM VSAM database.

Serializing with Non-ISPF TSO and BATCH

ISPF relies on MVS allocation to serialize access to resources with concurrent batch or non-ISPF TSO users. ISPF uses dynamic allocation and allocates partitioned data sets with DISP=SHR before any ISPF generated ENQUEUE. To ensure integrity when batch or TSO users are not using ISPF services, if you are updating a data set, you must allocate the data set with DISP=OLD.

Note: ISPF's allocation with DISP=SHR causes MVS allocation to issue a shared ENQUEUE on Qname SYSDSN as follows:

```
ENQ SYSDSN,dsname,S,44,SYSTEM
```

Non-ISPF TSO and Batch allocation with DISP=OLD causes MVS allocation to issue an exclusive ENQUEUE on Qname SYSDSN as follows:

```
ENQ SYSDSN,dsname,E,44,SYSTEM
```

ISPF also issues ENQ, DEQ, and RESERVE macro instructions to serialize access to resources among multiple ISPF users. It is possible for the LMMOVE service or option 3.3 Move function to be used such that each of 2 users is holding a RESERVE (therefore an exclusive ENQ is held) that is being waited on by the other user. This occurs when the user 1 is moving from data set A to data set B, while at the same time, user 2 is moving from data set B to data set A. Each user holds a reserve on the output data set for the move and is requesting a reserve on the input for purposes of deleting the moved member. If this situation occurs, cancel one user off TSO, log back on, and reissue the move request.

ISPF Data Set Integrity Enqueue

To allow you to update a data set on DASD (EDIT's SAVE, PDF option 3 COPY/MOVE/DELETE/RENAME, TBSAVE, and so on), ISPF issues the following macro to protect the entire partitioned data set:

```
RESERVE SPFEDIT,dsname,E,44,SYSTEMS
```

Also, when writing to a data set on a DASD (COPY/MOVE) that has a RECFM of 'U', ISPF serializes with the linkage editor using the following macro to protect the entire partitioned data set:

On a shared volume:

```
RESERVE SYSIEWLP,dsname,E,44,SYSTEMS
```

On a volume that is not shared:

```
ENQ SYSIEWLP,dsname,E,44,SYSTEM
```

Attention: If SPF and ISPF are both installed on the same system, there is a danger of destroying PDSs that are being updated by SPF and ISPF at the same time because SPF uses a different Qname (SPFDSN) than ISPF.

Member Name Enqueue

To restrict concurrent use of a member of a partitioned data set, while still allowing ISPF users to use different members of the same data set (PDF EDIT, Table Processing, File Tailoring), ISPF issues the following ENQ macro for the member:

```
ENQ SPFEDIT,rname,E,52,SYSTEMS
```

where

rname the data set name, length of 44, padded with blanks, followed by the member name, length of 8, padded with blanks

ISPF GUI TSO Linemode Message Queue Enqueue

Running ISPF GUI with TSO linemode support (that is, displaying TSO linemode 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 linemode and tput/tget messages so that they will be synchronized correctly. Thus, ISPF issues the following 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 the following 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 the following 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. Dialog Development Model Listings

This appendix lists all of the models shipped with PDF. The table shows the external model name, any qualifiers, a short description, and the internal member name in the SKELS library shipped with PDF. See *z/OS ISPF Edit and Edit Macros* for more information about adding or changing a model.

DM and PDF Services in CLIST Commands

Model Name		Description	Member Name
Display			
ADDPOP		Display pop-up window	ISREMCD5
DISPLAY		Display option	ISREMCD1
PQUERY		Get panel information	ISREMCD4
REMPOP		Remove pop-up window	ISREMCD6
SETMSG		Set message display	ISREMCD3
TBDISPL		Table display information	ISREMCD2
File Tailoring			
FTCLOSE		End file tailoring	ISREMCF3
FTERASE		File tailor erase	ISREMCF4
FTINCL		File tailor include skeleton	ISREMCF2
FTOPEN		File tailor open	ISREMCF1
Library Access			
LMCLOSE		Close a data set	ISREMCL1
LMCOMP		Compress a data set	ISREMCLU
LMCOPY		Copy a data set	ISREMCLQ
LMDDISP		Data set list	ISREMCLZ
LMDFREE		Release a data set list	ISREMCLW
LMDINIT		Establish a data set ID	ISREMCLX
LMDLIST		Obtain a list of data sets	ISREMCLV
LMERASE		Erase a data set or library	ISREMCL2
LMFREE		Release a data set	ISREMCL3
LMGET		Read a record	ISREMCL4
LMINIT		Establish a data ID	ISREMCL5
LMMADD		Add a member	ISREMCL6
LMMDEL		Delete a member	ISREMCL7
LMMDISP		Display member list	ISREMCLO
LMMFIND		Find a member	ISREMCL8
LMMLIST		Create a member list	ISREMCL9
LMMOVE		Move a data set or member	ISREMCLP
LMMREN		Rename a member	ISREMCLA

Model Name		Description	Member Name
LMMREP		Replace a member	ISREMCLB
LMMSTATS		Set member statistics	ISREMCLR
LMOPEN		Open a data set	ISREMCLC
LMPRINT		Write member to list data set	ISREMCLT
LMPUT		Write a record	ISREMCLE
LMQUERY		Provide data set information	ISREMCLF
LMRENAME		Rename a library	ISREMCLG
MEMLIST		Displays Option 3.1 member list	ISREMCLH
Miscellaneous			
BROWSE		Browse service	ISREMCM3
CONTROL		Control service	ISREMCM2
DSINFO		Returns data set information	ISREMCME
EDIT		Edit service	ISREMCM4
EDREC		Edit recovery services	ISREMCM7
GETMSG		Get message service	ISREMCM6
LIBDEF		LIBDEF service	ISREMCM8
LIST		Write data to list data set	ISREMCMA
LOG		Write message or log data set	ISREMCM5
QBASELIB		Query base library information	ISREMCMC
QLIBDEF		Query LIBDEF library information	ISREMCMD
SELECT		Select service	ISREMCM1
Table Functions (General)			
TBCLOSE		Table close	ISREMCG5
TBCREATE		Table create	ISREMCG1
TBEND		Table end	ISREMCG6
TBERASE		Table erase	ISREMCG7
TBOPEN		Table open	ISREMCG2
TBQUERY		Table query	ISREMCG3
TBSAVE		Table save	ISREMCG4
TBSTATS		Table statistics	ISREMCG8
Table Functions (Row)			
TBADD		Table row add	ISREMCR1
TBBOTTOM		Table row pointer to bottom	ISREMCRA
TBDELETE		Table delete	ISREMCR2
TBEXIST		Table exist	ISREMCR6
TBGET		Table get	ISREMCR3
TBMOD		Table modify	ISREMCR5
TBPUT		Table put	ISREMCR4
TBSARG		Table search parameter	ISREMCR7
TBSCAN		Table scan	ISREMCR8
TBSKIP		Table skip	ISREMCRB

Model Name		Description	Member Name
TBSORT		Table sort	ISREMCRD
TBTOP		Table top	ISREMCR9
TBVCLEAR		Table variable clear	ISREMCRC
Variables			
VERASE		Variable erase	ISREMCV8
VGET		Variable get	ISREMCV1
VPUT		Variable put	ISREMCV2
Workstation			
FILESTAT		Statistics for a file	ISREMCX1
FILEXFER		Upload or download a file	ISREMCX2
WSCON		Workstation connection	ISREMCX3
WSDISCON		Workstation disconnect	ISREMCX4

DM and PDF Services in COBOL Programs

Model Name		Description	Member Name
Display			
ADDPOP		Display pop-up window	ISREMBD5
DISPLAY		Display option	ISREMBD1
PQUERY		Get panel information	ISREMBD4
REMPOP		Remove pop-up window	ISREMBD6
SETMSG		Set message display	ISREMBD3
TBDISPL		Table display information	ISREMBD2
File Tailoring			
FTCLOSE		End file tailoring	ISREMBF3
FTERASE		File tailor erase	ISREMBF4
FTINCL		File tailor include skeleton	ISREMBF2
FTOPEN		File tailor open	ISREMBF1
Graphics			
GRERROR		Graphics error block service	ISREMBS3
GRINIT		Graphics initialization	ISREMBS1
GRTERM		Graphics completion service	ISREMBS2
Library Access			
LMCLOSE		Close a data set	ISREMBL1
LMCOMP		Compress a data set	ISREMBLS
LMCOPY		Copy a data set	ISREMBLQ
LMDDISP		Data set list	ISREMBLZ
LMDFREE		Release a data set list	ISREMBLW
LMDINIT		Establish a data set ID	ISREMBLU
LMDLIST		Obtain a list of data sets	ISREMBLV
LMERASE		Erase a data set or library	ISREMBL2

Model Name		Description	Member Name
LMFREE		Release a data set	ISREMBL3
LMGET		Read a record	ISREMBL4
LMINIT		Establish a data ID	ISREMBL5
LMMADD		Add a member	ISREMBL6
LMMDEL		Delete a member	ISREMBL7
LMMDISP		Display member list	ISREMBLO
LMMFIND		Find a member	ISREMBL8
LMMLIST		Create a member list	ISREMBL9
LMMOVE		Move a data set or member	ISREMBLP
LMMREN		Rename a member	ISREMBLA
LMMREP		Replace a member	ISREMBLB
LMMSTATS		Set member statistics	ISREMBLR
LMOPEN		Open a data set	ISREMBLC
LMPRINT		Write member to list data set	ISREMBLT
LMPUT		Write a record	ISREMBLE
LMQUERY		Provide data set information	ISREMBLF
LMRENAME		Rename a library	ISREMBLG
MEMLIST		Displays Option 3.1 member list	ISREMBLH
Miscellaneous			
BRIF		Browse interface service	ISREMCB4
BROWSE	MVS	Browse service (MVS)	ISREMCB1
BROWSE	VM	Browse service (VM)	ISREMCB2
BROWSE	VSE	Browse service (VSE)	ISREMCB3
CONTROL		Control service	ISREMBM2
DSINFO		Returns data set information	ISREMBME
EDIF		Edit interface service	ISREMCE4
EDIREC		Edit recovery for EDIF	ISREMBM9
EDIT	MVS	Edit service (MVS)	ISREMCE1
EDIT	VM	Edit service (VM)	ISREMCE2
EDIT	VSE	Edit service (VSE)	ISREMCE3
EDREC		Edit recovery services	ISREMBM7
GETMSG		Get message service	ISREMBM6
LIBDEF		LIBDEF service	ISREMBM8
LIST		Write to list data set	ISREMBMA
LOG		Write message or log data set	ISREMBM5
QBASELIB		Query base library information	ISREMBMC
QLIBDEF		Query LIBDEF library information	ISREMBMD
SELECT		Select service	ISREMBM1
VIIF		View Interface service	ISREMCE6
Table Functions (General)			
TBCLOSE		Table close	ISREMBG5

Model Name		Description	Member Name
TBCREATE		Table create	ISREMBG1
TBEND		Table end	ISREMBG6
TBERASE		Table erase	ISREMBG7
TBOPEN		Table open	ISREMBG2
TBQUERY		Table query	ISREMBG3
TBSAVE		Table save	ISREMBG4
TBSTATS		Table statistics	ISREMBG8
Table Functions (Row)			
TBADD		Table row add	ISREMBR1
TBBOTTOM		Table row pointer to bottom	ISREMBRA
TBDELETE		Table delete	ISREMBR2
TBEXIST		Table exist	ISREMBR6
TBGET		Table get	ISREMBR3
TBMOD		Table modify	ISREMBR5
TBPUT		Table put	ISREMBR4
TBSARG		Table search parameter	ISREMBR7
TBSCAN		Table scan	ISREMBR8
TBSKIP		Table skip	ISREMBRB
TBSORT		Table sort	ISREMBRD
TBTOP		Table top	ISREMBR9
TBVCLEAR		Table variable clear	ISREMBRC
Variables			
VCOPY		Copy variable	ISREMBV5
VDEFINE		Variable define	ISREMBV3
VDELETE		Variable delete	ISREMBV4
VERASE		Variable erase	ISREMBV8
VGET		Variable get	ISREMBV1
VMASK		Variable mask	ISREMBV9
VPUT		Variable put	ISREMBV2
VREPLACE		Variable replace	ISREMBV6
VRESET		Variable reset	ISREMBV7
Working Storage			
WORKSTOR		Working storage definition	ISREMBW1
Workstation			
FILESTAT		Statistics for a file	ISREMBX1
FILEXFER		Upload or download a file	ISREMBX2
WSCON		Workstation connection	ISREMBX3
WSDISCON		Workstation disconnect	ISREMBX4

Model Name		Description	Member Name
Display			
ADDPOP		Display pop-up window	ISREMED5
DISPLAY		Display option	ISREMED1
PQUERY		Get panel information	ISREMED4
REMPOP		Remove pop-up window	ISREMED6
SETMSG		Set message display	ISREMED3
TBDISPL		Table display information	ISREMED2
File Tailoring			
FTCLOSE		End file tailoring	ISREMEF3
FTERASE		File tailor erase	ISREMEF4
FTINCL		File tailor include skeleton	ISREMEF2
FTOPEN		File tailor open	ISREMEF1
Library Access			
LMCLOSE		Close a data set	ISREMEL1
LMERASE		Erase a data set or library	ISREMEL2
LMFREE		Release a data set	ISREMEL3
LMGET		Read a record	ISREMEL4
LMINIT		Establish a data ID	ISREMEL5
LMMADD		Add a member	ISREMEL6
LMMDEL		Delete a member	ISREMEL7
LMMFIND		Find a member	ISREMEL8
LMMLIST		Create a member list	ISREMEL9
LMMREN		Rename a member	ISREMELA
LMMREP		Replace a member	ISREMELB
LMOPEN		Open a data set	ISREMELC
LMPUT		Write a record	ISREMELE
LMQUERY		Provide data set information	ISREMELF
LMRENAME		Rename a library	ISREMELG
LMSPEC		Specify a new ISPF library	ISREMELH
LMUNSPEC		Unspecify an ISPF library	ISREMELI
Miscellaneous			
BROWSE		Browse service	ISREMEM3
CONTROL		Control service	ISREMEM2
EDIT		Edit service	ISREMEM4
EDREC		Edit recovery services	ISREMEM7
GETMSG		Get message service	ISREMEM6
LIBDEF		LIBDEF service	ISREMEM8
LIST		Write list data set	ISREMEMA
LOG		Write message or log data set	ISREMEM5

Model Name		Description	Member Name
QBASELIB		Query base library information	ISREMEMC
QLIBDEF		Query LIBDEF library information	ISREMEMD
SELECT		Select service	ISREMEM1
Table Functions (General)			
TBCLOSE		Table close	ISREMEG5
TBCREATE		Table create	ISREMEG1
TBEND		Table end	ISREMEG6
TBERASE		Table erase	ISREMEG7
TBOPEN		Table open	ISREMEG2
TBQUERY		Table query	ISREMEG3
TBSAVE		Table save	ISREMEG4
TBSTATS		Table statistics	ISREMEG8
Table Functions (Row)			
TBADD		Table row add	ISREMER1
TBBOTTOM		Table row pointer to bottom	ISREMERA
TBDELETE		Table delete	ISREMER2
TBEXIST		Table exist	ISREMER6
TBGET		Table get	ISREMER3
TBMOD		Table modify	ISREMER5
TBPUT		Table put	ISREMER4
TBSARG		Table search parameter	ISREMER7
TBSCAN		Table scan	ISREMER8
TBSKIP		Table skip	ISREMERB
TBSORT		Table sort	ISREMERD
TBTOP		Table top	ISREMER9
TBVCLEAR		Table variable clear	ISREMERC
Variables			
VGET		Variable get	ISREMEV1
VPUT		Variable put	ISREMEV2
VERASE		Variable erase	ISREMEV8

DM and PDF Services in FORTRAN Programs

Model Name		Description	Member Name
Display			
ADDPOP		Display pop-up window	ISREMFD5
DISPLAY		Display option	ISREMFD1
PQUERY		Get panel information	ISREMFD4
REMPOP		Remove pop-up window	ISREMFD6
SETMSG		Set message display	ISREMFD3
TBDISPL		Table display information	ISREMFD2

Model Name		Description	Member Name
File Tailoring			
FTCLOSE		End file tailoring	ISREMFF3
FTERASE		File tailor erase	ISREMFF4
FTINCL		File tailor include skeleton	ISREMFF2
FTOPEN		File tailor open	ISREMFF1
Graphics			
GRERROR		Graphics error block service	ISREMFS3
GRINIT		Graphics initialization	ISREMFS1
GRTERM		Graphics completion service	ISREMFS2
Library Access			
LMCLOSE		Close a data set	ISREMFL1
LMCOMP		Compress a data set	ISREMFLS
LMCOPY		Copy a data set	ISREMFLQ
LMDDISP		Data set list	ISREMFLZ
LMDFREE		Release a data set list	ISREMFLW
LMDINIT		Establish a data set ID	ISREMFLU
LMDLIST		Obtain a list of data sets	ISREMFLV
LMERASE		Erase a data set or library	ISREMFL2
LMFREE		Release a data set	ISREMFL3
LMGET		Read a record	ISREMFL4
LMINIT		Establish a data ID	ISREMFL5
LMMADD		Add a member	ISREMFL6
LMMDEL		Delete a member	ISREMFL7
LMMDISP		Display member list	ISREMFLO
LMMFIND		Find a member	ISREMFL8
LMMLIST		Create a member list	ISREMFL9
LMMOVE		Move a data set or member	ISREMFLP
LMMREN		Rename a member	ISREMFLA
LMMREP		Replace a member	ISREMFLB
LMMSTATS		Set member statistics	ISREMFLR
LMOPEN		Open a data set	ISREMFLC
LMPRINT		Write member to list data set	ISREMFLT
LMPUT		Write a record	ISREMFLE
LMQUERY		Provide data set information	ISREMFLF
LMRENAME		Rename a library	ISREMFLG
MEMLIST		Displays Option 3.1 member list	ISREMFLH
Miscellaneous			
BRIF		Browse interface service	ISREMFB4
BROWSE	MVS	Browse service (MVS)	ISREMFB1
BROWSE	VM	Browse service (VM)	ISREMFB2
BROWSE	VSE	Browse service (VSE)	ISREMFB3

Model Name		Description	Member Name
CONTROL		Control service	ISREMFM2
DSINFO		Returns data set information	ISREMFME
EDIF		Edit interface service	ISREMFE4
EDIREC		Edit recovery for EDIF	ISREMFM9
EDIT	MVS	Edit service (MVS)	ISREMFE1
EDIT	VM	Edit service (VM)	ISREMFE2
EDIT	VSE	Edit service (VSE)	ISREMFE3
EDREC		Edit recovery services	ISREMFM7
GETMSG		Get message service	ISREMFM6
LIBDEF		LIBDEF service	ISREMFM8
LIST		Write to list data set	ISREMFMA
LOG		Write message or log data set	ISREMFM5
QBASELIB		Query base library information	ISREMFMC
QLIBDEF		Query LIBDEF library information	ISREMFMD
SELECT		Select service	ISREMFM1
VIIF		View Interface service	ISREMFE6
Table Functions (General)			
TBCLOSE		Table close	ISREMFZ5
TBCREATE		Table create	ISREMFZ1
TBEND		Table end	ISREMFZ6
TBERASE		Table erase	ISREMFZ7
TBOPEN		Table open	ISREMFZ2
TBQUERY		Table query	ISREMFZ3
TBSAVE		Table save	ISREMFZ4
TBSTATS		Table statistics	ISREMFZ8
Table Functions (Row)			
TBADD		Table row add	ISREMFR1
TBBOTTOM		Table row pointer to bottom	ISREMFRA
TBDELETE		Table delete	ISREMFR2
TBEXIST		Table exist	ISREMFR6
TBGET		Table get	ISREMFR3
TBMOD		Table modify	ISREMFR5
TBPUT		Table put	ISREMFR4
TBSARG		Table search parameter	ISREMFR7
TBSCAN		Table scan	ISREMFR8
TBSKIP		Table skip	ISREMFRB
TBSORT		Table sort	ISREMFRD
TBTOP		Table top	ISREMFR9
TBVCLEAR		Table variable clear	ISREMFRC
Variables			
VCOPY		Copy variable	ISREMFV5

Model Name		Description	Member Name
VDEFINE		Variable define	ISREMFV3
VDELETE		Variable delete	ISREMFV4
VERASE		Variable erase	ISREMFV8
VGET		Variable get	ISREMFV1
VMASK		Variable mask	ISREMBV9
VPUT		Variable put	ISREMFV2
VREPLACE		Variable replace	ISREMFV6
VRESET		Variable reset	ISREMFV7
Workstation			
FILESTAT		Statistics for a file	ISREMFX1
FILEXFER		Upload or download a file	ISREMFX2
WSCON		Workstation connection	ISREMFX3
WSDISCON		Workstation disconnect	ISREMFX4

Message Format

Model Name		Description	Member Name
MSGS		Message member selection	ISREMMSG

Panel Formats and Statements

Model Name	Qualifier	Description	Member Name
Panel Formats			
ACTION		'Panel with action bar'	ISREMMF6
ENTRY		'Data entry panel'	ISREMMF1
HELPSCR		'Help panel with scrollable area'	ISREMMF8
MULTIPLE	SELECT2	'Double-column selection panel'	ISREMSE1
MULTIPLE	ENTRY2	'Double-column entry panel'	ISREMSE2
SCROLL		'Panel with scrollable area'	ISREMMF7
SELECTION		'Choice panel'	ISREMMF3
SELECTION	CUA	'Choice panel'	ISREMMF9
TBDISPL		'Table display table'	ISREMMF4
TUTORIAL		'Help panel'	ISREMMF5
Panel Statements			
ABC		'Action bar'	ISREMMSE
AREA		'AREA section header'	ISREMMPI
AREA	DYNAMIC	'Dynamic area attribute'	ISREMMA1
AREA	GRAPHIC	'Graphic area attribute'	ISREMMA2
AREA	SCRL	'Scrollable area attribute'	ISREMMA3
ASSIGN	SIMPLE	'Simple assignment statement'	ISREMAS1
ASSIGN	TRANS	'Trans assignment statement'	ISREMAS2

Model Name	Qualifier	Description	Member Name
ASSIGN	TRUNC	'Trunc assignment statement'	ISREMAS3
ASSIGN	TRANSTRU	'Nested translate truncate statement'	ISREMAS4
ASSIGN	PFKEY	'Function key built-in function'	ISREMAS5
ASSIGN	LVLINE	'Last visible line built-in function'	ISREMAS6
ATTR		'Attribute section header'	ISREMMS2
ATTRIB		'New attribute character definition'	ISREMMS3
ATTRIBA		'New attrib char definition for area'	ISREMMSB
BODY		'Body section header'	ISREMMS4
CCSID		'CCSID section header'	ISREMMPA
CONTROL	CURSOR	'Control first cursor placement'	ISREMCN1
CONTROL	HELP	'Establish a tutorial panel'	ISREMCN2
CONTROL	MSG	'Identify message to be displayed'	ISREMCN3
CONTROL	RESP	'Show user response to panel'	ISREMCN4
CONTROL	TRAIL	'Contain remainder from TRUNC function'	ISREMCN5
CONTROL	ALARM	'Shows the alarm is to be sounded'	ISREMCN6
CONTROL	ATTR	'Override field attr by field name'	ISREMCN7
CONTROL	ATTRCHAR	'Override field attr by character'	ISREMCN8
CONTROL	AUTOSEL	'Control table display row selection'	ISREMCN9
CONTROL	CSRPOS	'Shows position of cursor in field'	ISREMCNA
CONTROL	CSRROW	'Row where cursor is positioned'	ISREMCNB
CONTROL	PFKEY	'Function key pressed by user'	ISREMCND
CONTROL	ZVARS	'Define names of placeholder fields'	ISREMCNC
CUAATTR		'CUA attributes'	ISREMMSJ
END		'END section header'	ISREMMPP
GROUPBOX		'TYPE(GRPBOX) attribute'	ISREMMA4
HELP		'HELP section'	ISREMMPM
IF		'IF statement'	ISREMMS6
INIT		'INIT section header'	ISREMMPJ
KEYLIST		'Keylist specification'	ISREMMSF
LIST		'LIST section'	ISREMMPO
MODEL		'Model section header'	ISREMMS7
PANEXIT		'Panel language exit'	ISREMMSD
PDC		'Action bar pull-down'	ISREMMSH
PNTS		'Point-and-shoot section'	ISREMMPN
PROC		'PROC section header'	ISREMMPL
REFRESH		'Retrieve variables before redisplay'	ISREMMSA
REINIT		'REINIT section header'	ISREMMPK
SC	ATTR	'TYPE(SC) attribute'	ISREMMA5
VEDIT		'Validate available'	ISREMMSI
VERIFY	ALPHA	'Alphabetic or special characters'	ISREMVE1

Model Name	Qualifier	Description	Member Name
VERIFY	ALPHAB	'Alphabetic characters'	ISREMVEE
VERIFY	BIT	'Binary characters'	ISREMVE2
VERIFY	DSNAME	'TSO data set name'	ISREMVE3
VERIFY	DSNAMEF	'TSO data set name with filters'	ISREMVEL
VERIFY	DSNAMEFM	'TSO data set name with filter member name only'	ISREMVEO
VERIFY	DSNAMEPQ	'TSO data set name' (adds missing end parenthesis and quote)	ISREMVEx
VERIFY	DSNAMEQ	'TSO data set name' (adds missing end quote)	ISREMVEJ
VERIFY	ENUM	'Extended numeric'	ISREMVED
VERIFY	FILEID	'CMS file ID'	ISREMVE4
VERIFY	IPADDR4	IP Version 4 address	ISREMVEV
VERIFY	HEX	'Hexadecimal characters'	ISREMVE5
VERIFY	IDATE	'International date'	ISREMVEP
VERIFY	INCLUDE	'Specify list of types'	ISREMVEF
VERIFY	ITIME	'International time'	ISREMVET
VERIFY	JDATE	'Julian date'	ISREMVER
VERIFY	JSTD	'Julian standard date'	ISREMVES
VERIFY	LEN	'Length of data stored in variable'	ISREMVEC
VERIFY	LIST	'List of valid values'	ISREMVE6
VERIFY	LISTV	'Specify list of values'	ISREMVEG
VERIFY	LISTVX	'Specify list of excluded values'	ISREMVEI
VERIFY	LISTX	'Specify list of excluded values'	ISREMVEH
VERIFY	NAME	'Data set member name'	ISREMVE7
VERIFY	NAMEF	'Data set member name with filters'	ISREMVEM
VERIFY	NONBLANK	'Verify nonblank field'	ISREMVE8
VERIFY	NUM	'Numeric characters'	ISREMVE9
VERIFY	PICT	'Mixed characters matching picture'	ISREMVEA
VERIFY	PICTCN	'Constants and mixed characters matching picture'	ISREMVEK
VERIFY	RANGE	'Numeric value within specified limits'	ISREMVEB
VERIFY	STDDATE	'Standard date'	ISREMVEQ
VERIFY	STDTIME	'Standard time'	ISREMVEU
VGET		'Variable get statement'	ISREMMSC
VPUT		'Variable put statement'	ISREMMS9

DM and PDF Services in PL/I Programs

Model Name		Description	Member Name
Display			

Model Name		Description	Member Name
ADDPOP		'Display pop-up window'	ISREMPD5
DISPLAY		'Display option'	ISREMPD1
PQUERY		'Get panel information'	ISREMPD4
REMPOP		'Remove pop-up window'	ISREMPD6
SETMSG		'Set message display'	ISREMPD3
TBDISPL		'Table display information'	ISREMPD2
File Tailoring			
FTCLOSE		'End file tailoring'	ISREMPF3
FTERASE		'File tailor erase'	ISREMPF4
FTINCL		'File tailor include skeleton'	ISREMPF2
FTOPEN		'File tailor open'	ISREMPF1
Graphics			
GRERROR		'Graphics error block service'	ISREMPS3
GRINIT		'Graphics initialization'	ISREMPS1
GRTERM		'Graphics completion service'	ISREMPS2
Library Access			
LMCLOSE		'Close a data set'	ISREMPL1
LMCOMP		'Compress a data set'	ISREMPLS
LMCOPY		'Copy a data set'	ISREMPLQ
LMDDISP		Data set list	ISREMPLZ
LMDFREE		Release a data set list	ISREMPLW
LMDINIT		Establish a data set ID	ISREMPLU
LMDLIST		Obtain a list of data sets	ISREMPLV
LMERASE		'Erase a data set or library'	ISREMPL2
LMFREE		'Release a data set'	ISREMPL3
LMGET		'Read a record'	ISREMPL4
LMINIT		'Establish a data ID'	ISREMPL5
LMMADD		'Add a member'	ISREMPL6
LMMDEL		'Delete a member'	ISREMPL7
LMMDISP		'Display member list'	ISREMPLO
LMMFIND		'Find a member'	ISREMPL8
LMMLIST		'Create a member list'	ISREMPL9
LMMOVE		'Move a data set or member'	ISREMPLP
LMMREN		'Rename a member'	ISREMPLA
LMMREP		'Replace a member'	ISREMPLB
LMMSTATS		'Set member statistics'	ISREMPLR
LMOPEN		'Open a data set'	ISREMPLC
LMPRINT		'Write member to list data set'	ISREMPLT
LMPUT		'Write a record'	ISREMPLE
LMQUERY		'Provide data set information'	ISREMPLF
LMRENAME		'Rename a library'	ISREMPLG

Model Name		Description	Member Name
MEMLIST		Displays Option 3.1 member list	ISREMPLH
Miscellaneous			
BRIF		'Browse interface service'	ISREMPB4
BROWSE	MVS	'Browse service (MVS)'	ISREMPB1
BROWSE	VM	'Browse service (VM)'	ISREMPB2
BROWSE	VSE	'Browse service (VSE)'	ISREMPB3
CONTROL		'Control service'	ISREMPM2
DSINFO		Returns data set information	ISREMPME
EDIF		'Edit interface service'	ISREMPE4
EDIREC		'EDIF recovery service'	ISREMPM9
EDIT	MVS	'Edit service (MVS)'	ISREMPE1
EDIT	VM	'Edit service (VM)'	ISREMPE2
EDIT	VSE	'Edit service (VSE)'	ISREMPE3
EDREC		'Edit recovery services'	ISREMPM7
GETMSG		'Get message service'	ISREMPM6
LIBDEF		'LIBDEF service'	ISREMPM8
LIST		'Write to list data set'	ISREMPMA
LOG		'Write message or log data set'	ISREMPM5
QBASELIB		Query base library information	ISREMPMC
QLIBDEF		Query LIBDEF library information	ISREMPMD
SELECT		'SELECT service'	ISREMPM1
VIIF		'View Interface service'	ISREMPE6
Table Functions (General)			
TBCLOSE		'Table close'	ISREMPZ5
TBCREATE		'Table create'	ISREMPZ1
TBEND		'Table end'	ISREMPZ6
TBERASE		'Table erase'	ISREMPZ7
TBOPEN		'Table open'	ISREMPZ2
TBQUERY		'Table query'	ISREMPZ3
TBSAVE		'Table save'	ISREMPZ4
TBSTATS		'Table statistics'	ISREMPZ8
Table Functions (Row)			
TBADD		'Table row add'	ISREMPR1
TBBOTTOM		'Table row pointer to bottom'	ISREMPRA
TBDELETE		'Table delete'	ISREMPR2
TBEXIST		'Table exist'	ISREMPR6
TBGET		'Table get'	ISREMPR3
TBMOD		'Table modify'	ISREMPR5
TBPUT		'Table put'	ISREMPR4
TBSARG		'Table search parameter'	ISREMPR7
TBSCAN		'Table scan'	ISREMPR8

Model Name		Description	Member Name
TBSKIP		'Table skip'	ISREMPRB
TBSORT		'Table sort'	ISREMPRD
TBTOP		'Table top'	ISREMPR9
TBVCLEAR		'Table variable clear'	ISREMPRC
Variables			
VCOPY		'Copy variable'	ISREMPV5
VDEFINE		'Variable define'	ISREMPV3
VDELETE		'Variable delete'	ISREMPV4
VERASE		'Variable erase'	ISREMPV8
VGET		'Variable get'	ISREMPV1
VMASK		'Variable mask'	ISREMPV9
VPUT		'Variable put'	ISREMPV2
VREPLACE		'Variable replace'	ISREMPV6
VRESET		'Variable reset'	ISREMPV7
Workstation			
FILESTAT		Statistics for a file	ISREMPX1
FILEXFER		Upload or download a file	ISREMPX2
WSCON		Workstation connection	ISREMPX3
WSDISCON		Workstation disconnect	ISREMPX4

File Tailoring Control Statements

Model Name		Description	Member Name
BLANK		Create blank lines	ISREMSK8
CM		Define comment statement	ISREMSK1
DEFAULT		Change control character defaults	ISREMSK2
DOT		Define DO group for table row	ISREMSK3
IM		Imbed specified data set skeleton	ISREMSK4
SEL		Conditional processing definition	ISREMSK5
SET		Set dialog variable value	ISREMSK6
TB		Set tab stop position	ISREMSK7

DM and PDF Services in Pascal Programs

Model Name		Description	Member Name
Display			
ADDPOP		'Display pop-up window'	ISREMQD5
DISPLAY		'Display option'	ISREMQD1
PQUERY		'Get panel information'	ISREMQD4
REMPOP		'Remove pop-up window'	ISREMQD6
SETMSG		'Set message display'	ISREMQD3

Model Name		Description	Member Name
TBDISPL		'Table display information'	ISREMQD2
File Tailoring			
FTCLOSE		'End file tailoring'	ISREMQF3
FTERASE		'File tailor erase'	ISREMQF4
FTINCL		'File tailor include skeleton'	ISREMQF2
FTOPEN		'File tailor open'	ISREMQF1
Graphics			
GRERROR		'Graphics error block service'	ISREMQS3
GRINIT		'Graphics initialization'	ISREMQS1
GRTERM		'Graphics completion service'	ISREMQS2
Library Access			
LMCLOSE		'Close a data set'	ISREMQL1
LMCOMP		'Compress a data set'	ISREMQLS
LMCOPY		'Copy a data set'	ISREMQLQ
LMDDISP		Data set list	ISREMQLZ
LMDFREE		Release a data set list	ISREMQLW
LMDINIT		Establish a data set ID	ISREMQLU
LMDLIST		Obtain a list of data sets	ISREMQLV
LMERASE		'Erase a data set or library'	ISREMQL2
LMFREE		'Release a data set'	ISREMQL3
LMGET		'Read a record'	ISREMQL4
LMINIT		'Establish a data ID'	ISREMQL5
LMMADD		'Add a member'	ISREMQL6
LMMDEL		'Delete a member'	ISREMQL7
LMMDISP		'Display member list'	ISREMQLO
LMMFIND		'Find a member'	ISREMQL8
LMMLIST		'Create a member list'	ISREMQL9
LMMOVE		'Move a data set or member'	ISREMQLP
LMMREN		'Rename a member'	ISREMQLA
LMMREP		'Replace a member'	ISREMQLB
LMMSTATS		'Set member statistics'	ISREMQLR
LMOPEN		'Open a data set'	ISREMQLC
LMPRINT		'Write member to list data set'	ISREMQLT
LMPUT		'Write a record'	ISREMQLE
LMQUERY		'Provide data set information'	ISREMQLF
LMRENAME		'Rename a library'	ISREMQLG
MEMLIST		'Displays Option 3.1 member list'	ISREMQLH
Miscellaneous			
BRIF		'Browse interface service'	ISREMQB4
BROWSE	MVS	'Browse service (MVS)'	ISREMQB1
BROWSE	VM	'Browse service (VM)'	ISREMQB2

Model Name		Description	Member Name
BROWSE	VSE	'Browse service (VSE)'	ISREMQB3
CONTROL		'Control service'	ISREMQM2
DSINFO		'Returns data set information'	ISREMQME
EDIF		'Edit interface service'	ISREMQE4
EDIREC		'Edit recovery for EDIF'	ISREMQM9
EDIT	MVS	'Edit service (MVS)'	ISREMQE1
EDIT	VM	'Edit service (VM)'	ISREMQE2
EDIT	VSE	'Edit service (VSE)'	ISREMQE3
EDREC		'Edit recovery services'	ISREMQM7
GETMSG		'Get message service'	ISREMQM6
LIBDEF		'LIBDEF service'	ISREMQM8
LIST		'Write to list data set'	ISREMQMA
LOG		'Write message or log data set'	ISREMQM5
QBASELIB		Query base library information	ISREMQMC
QLIBDEF		Query LIBDEF library information	ISREMQMD
SELECT		'Select service'	ISREMQM1
VIIF		'View Interface service'	ISREMQE6
Pascal Definitions			
PASDEFS		'Pascal Definitions'	ISREMQPD
Table Functions (General)			
TBCLOSE		'Table close'	ISREMQZ5
TBCREATE		'Table create'	ISREMQZ1
TBEND		'Table end'	ISREMQZ6
TBERASE		'Table erase'	ISREMQZ7
TBOPEN		'Table open'	ISREMQZ2
TBQUERY		'Table query'	ISREMQZ3
TBSAVE		'Table save'	ISREMQZ4
TBSTATS		'Table statistics'	ISREMQZ8
Table Functions (Row)			
TBADD		'Table row add'	ISREMQR1
TBBOTTOM		'Table row pointer to bottom'	ISREMQRA
TBDELETE		'Table delete'	ISREMQR2
TBEXIST		'Table exist'	ISREMQR6
TBGET		'Table get'	ISREMQR3
TBMOD		'Table modify'	ISREMQR5
TBPUT		'Table put'	ISREMQR4
TBSARG		'Table search parameter'	ISREMQR7
TBSCAN		'Table scan'	ISREMQR8
TBSKIP		'Table skip'	ISREMQRB
TBSORT		'Table sort'	ISREMQRD
TBTOP		'Table top'	ISREMQR9

Model Name		Description	Member Name
TBVCLEAR		'Table variable clear'	ISREMQRC
Variables			
VCOPY		'Copy variable'	ISREMQV5
VDEFINE		'Variable define'	ISREMQV3
VDELETE		'Variable delete'	ISREMQV4
VERASE		'Variable erase'	ISREMQV8
VGET		'Variable get'	ISREMQV1
VMASK		'Variable mask'	ISREMQV9
VPUT		'Variable put'	ISREMQV2
VREPLACE		'Variable replace'	ISREMQV6
VRESET		'Variable reset'	ISREMQV7
Workstation			
FILESTAT		Statistics for a file	ISREMQX1
FILEXFER		Upload or download a file	ISREMQX2
WSCON		Workstation connection	ISREMQX3
WSDISCON		Workstation disconnect	ISREMQX4

DM and PDF Services in TSO/REXX Commands

Model Name		Description	Member Name
Display			
ADDPOP		'Display pop-up window'	ISREMRD5
DISPLAY		'Display option'	ISREMRD1
PQUERY		'Get panel information'	ISREMRD4
REMPOP		'Remove pop-up window'	ISREMRD6
SETMSG		'Set message display'	ISREMRD3
TBDISPL		'Table display information'	ISREMRD2
File Tailoring			
FTCLOSE		'End file tailoring'	ISREMRF3
FTERASE		'File tailor erase'	ISREMRF4
FTINCL		'File tailor include skeleton'	ISREMRF2
FTOPEN		'File tailor open'	ISREMRF1
Library Access			
LMCLOSE		'Close a data set'	ISREMRL1
LMCOMP		'Compress a data set'	ISREMRLU
LMCOPY		'Copy a data set'	ISREMRLQ
LMDDISP		'Data set list'	ISREMRLZ
LMDFREE		Release a data set list	ISREMRLW
LMDINIT		Establish a data set ID	ISREMRLX
LMDLIST		Obtain a list of data sets	ISREMRLV
LMERASE		'Erase a data set or library'	ISREMRL2

Model Name		Description	Member Name
LMFREE		'Release a data set'	ISREMRL3
LMGET		'Read a record'	ISREMRL4
LMINIT		'Establish a data ID'	ISREMRL5
LMMADD		'Add a member'	ISREMRL6
LMMDEL		'Delete a member'	ISREMRL7
LMMDISP		'Display member list'	ISREMRLO
LMMFIND		'Find a member'	ISREMRL8
LMMLIST		'Create a member list'	ISREMRL9
LMMOVE		'Move a data set or member'	ISREMRLP
LMMREN		'Rename a member'	ISREMRLA
LMMREP		'Replace a member'	ISREMRLB
LMMSTATS		'Set member statistics'	ISREMRLR
LMOPEN		'Open a data set'	ISREMRLC
LMPRINT		'Write member to list data set'	ISREMRLT
LMPUT		'Write a record'	ISREMRLE
LMQUERY		'Provide data set information'	ISREMRLF
LMRENAME		'Rename a library'	ISREMRLG
MEMLIST		'Displays Option 3.1 member list'	ISREMRLH
Miscellaneous			
BROWSE		'Browse service'	ISREMRM3
CONTROL		'Control service'	ISREMRM2
DSINFO		'Returns data set information'	ISREMRME
EDIT		'Edit service'	ISREMRM4
EDREC		'Edit recovery services'	ISREMRM7
GETMSG		'Get message service'	ISREMRM6
LIBDEF		'LIBDEF service'	ISREMRM8
LIST		'Write to list data set'	ISREMRMA
LOG		'Write message or log data set'	ISREMRM5
QBASELIB		Query base library information	ISREMRMC
QLIBDEF		Query LIBDEF library information	ISREMRMD
SELECT		'SELECT service'	ISREMRM1
Table Functions (General)			
TBCLOSE		'Table close'	ISREMRG5
TBCREATE		'Table create'	ISREMRG1
TBEND		'Table end'	ISREMRG6
TBERASE		'Table erase'	ISREMRG7
TBOPEN		'Table open'	ISREMRG2
TBQUERY		'Table query'	ISREMRG3
TBSAVE		'Table save'	ISREMRG4
TBSTATS		'Table statistics'	ISREMRG8
Table Functions (Row)			

Model Name		Description	Member Name
TBADD		'Table row add'	ISREMRR1
TBBOTTOM		'Table row pointer to bottom'	ISREMRRA
TBDELETE		'Table delete'	ISREMRR2
TBEXIST		'Table exist'	ISREMRR6
TBGET		'Table get'	ISREMRR3
TBMOD		'Table modify'	ISREMRR5
TBPUT		'Table put'	ISREMRR4
TBSARG		'Table search parameter'	ISREMRR7
TBSCAN		'Table scan'	ISREMRR8
TBSKIP		'Table skip'	ISREMRRB
TBSORT		'Table sort'	ISREMRRD
TBTOP		'Table top'	ISREMRR9
TBVCLEAR		'Table variable clear'	ISREMRRC
Variables			
VERASE		'Variable erase'	ISREMRV8
VGET		'Variable get'	ISREMRV1
VPUT		'Variable put'	ISREMRV2
Workstation			
FILESTAT		Statistics for a file	ISREMRX1
FILEXFER		Upload or download a file	ISREMRX2
WSCON		Workstation connection	ISREMRX3
WSDISCON		Workstation disconnect	ISREMRX4

SCLM Architecture Definition Formats

Model Name		Description	Member Name
Architecture Definition Formats			
'CC'		'Compilation Control'	ISREMHAC
'LEC'		'Linkage Editor Control'	ISREMHAL
'HL'		'High Level Definition'	ISREMHAH
'GENERIC'		'Special processing control'	ISREMHAG

SCLM Project Definition Macros and Templates

Model Name		Description	Member Name
Macros			
'FLMABEG'		'Define project name'	ISREMGAB
'FLMAEND'		'End project definition'	ISREMGAE
'FLMAGRP'		'Define authcode group'	ISREMGAG
'FLMALLOC'		'Define ddname'	ISREMGAL
'FLMALTC'		'Define alternate control set'	ISREMGAC

Model Name		Description	Member Name
'FLMATVER'		'Enable audit tracking and versioning'	ISREMGAV
'FLMCMPLB'		'Name compool library'	ISREMGCM
'FLMCNTRL'		'Specify project controls'	ISREMGCN
'FLMCPYLB'		'Name data set for allocation'	ISREMGCP
'FLMEXLIB'		'Define external library name'	ISREMGEX
'FLMGROUP'		'Define group of libraries'	ISREMGGR
'FLMLANGL'		'Define a language'	ISREMGLA
'FLMSYSLB'		'Name system library'	ISREMGSY
'FLMTRNSL'		'Define language translator'	ISREMGTR
'FLMTYPE'		'Define a library type'	ISREMGTY
Templates			
'PROJDEF'		'Project definition'	ISREMGT1
'LANGUAGE'		'Language definition'	ISREMGT2

ISPF Dialog Tag Language Models

Model Name		Description	Member Name
Panel Formats			
'ACTION BAR'		'Action bar panel'	ISREMDP2
ENTRY		'Data entry panel'	ISREMDP1
SELECTION		'Choice panel'	ISREMDP3
'TABLE DISPLAY'		'Scrollable list'	ISREMDP5
TUTORIAL		'Help/Tutorial panel'	ISREMDP4
Command Table Format			
'COMMAND TABLE'		'Command table application'	ISREMDC1

DM and PDF Services in C/370 Programs

Model Name		Description	Member Name
Display			
ADDPOP		'Display pop-up window'	ISREMWD5
DISPLAY		'Display option'	ISREMWD1
PQUERY		'Get panel information'	ISREMWD4
REMPOP		'Remove pop-up window'	ISREMWD6
SETMSG		'Set message display'	ISREMWD3
TBDISPL		'Table display information'	ISREMWD2
File Tailoring			
FTCLOSE		'End file tailoring'	ISREMWF3
FTERASE		'File tailor erase'	ISREMWF4

Model Name		Description	Member Name
FTINCL		'File tailor include skeleton'	ISREMWF2
FTOPEN		'File tailor open'	ISREMWF1
Graphics			
GRERROR		'Graphics error block service'	ISREMWS3
GRINIT		'Graphics initialization'	ISREMWS1
GRTERM		'Graphics completion service'	ISREMWS2
Library Access			
LMCLOSE		'Close a data set'	ISREMWL1
LMCOMP		'Compress a data set'	ISREMWL0
LMCOPY		'Copy a data set'	ISREMWLY
LMDDISP		Data set list	ISREMWLZ
LMDFREE		Release a data set list	ISREMWLW
LMDINIT		Establish a data set ID	ISREMWLU
LMDLIST		Obtain a list of data sets	ISREMWLV
LMERASE		'Erase a data set or library'	ISREMWL2
LMFREE		'Release a data set'	ISREMWL3
LMGET		'Read a record'	ISREMWL4
LMINIT		'Establish a data ID'	ISREMWL5
LMMADD		'Add a member'	ISREMWL6
LMMDEL		'Delete a member'	ISREMWL7
LMMDISP		'Display member list'	ISREMWLL
LMMFIND		'Find a member'	ISREMWL8
LMMLIST		'Create a member list'	ISREMWL9
LMMOVE		'Move a data set or member'	ISREMWLM
LMMREN		'Rename a member'	ISREMWL0
LMMREP		'Replace a member'	ISREMWLB
LMMSTATS		'Set member statistics'	ISREMWLP
LMOPEN		'Open a data set'	ISREMWLC
LMPRINT		'Write member to list data set'	ISREMWLQ
LMPUT		'Write a record'	ISREMWLE
LMQUERY		'Provide data set information'	ISREMWLF
LMRENAME		'Rename a library'	ISREMWLG
MEMLIST		'Displays Option 3.1 member list'	ISREMWLH
Miscellaneous			
BRIF		'Browse interface service'	ISREMWM9
BROWSE	MVS	'Browse service'	ISREMWM3
CONTROL		'Control service'	ISREMWM2
DSINFO		'Returns data set information'	ISREMWME
EDIF		'Edit interface service'	ISREMWMA
EDIREC		'Edit recovery for EDIF'	ISREMWMB
EDIT	MVS	'Edit service'	ISREMWM4

Model Name		Description	Member Name
EDREC		'Edit recovery services'	ISREMWM7
GETMSG		'Get message service'	ISREMWM6
LIBDEF		'LIBDEF service'	ISREMWM8
LIST		'Write to list data set'	ISREMWMC
LOG		'Write message or log data set'	ISREMWM5
QBASELIB		Query base library information	ISREMWQ2
QLIBDEF		Query LIBDEF library information	ISREMWQ1
SELECT		'Select service'	ISREMWM1
VIIF		'View Interface service'	ISREMWMF
Table Functions (General)			
TBCLOSE		'Table close'	ISREMWG5
TBCREATE		'Table create'	ISREMWG1
TBEND		'Table end'	ISREMWG6
TBERASE		'Table erase'	ISREMWG7
TBOPEN		'Table open'	ISREMWG2
TBQUERY		'Table query'	ISREMWG3
TBSAVE		'Table save'	ISREMWG4
TBSTATS		'Table statistics'	ISREMWG8
Table Functions (Row)			
TBADD		'Table row add'	ISREMWR1
TBBOTTOM		'Table row pointer to bottom'	ISREMWRA
TBDELETE		'Table delete'	ISREMWR2
TBEXIST		'Table exist'	ISREMWR6
TBGET		'Table get'	ISREMWR3
TBMOD		'Table modify'	ISREMWR5
TBPUT		'Table put'	ISREMWR4
TBSARG		'Table search parameter'	ISREMWR7
TBSCAN		'Table scan'	ISREMWR8
TBSKIP		'Table skip'	ISREMWRB
TBSORT		'Table sort'	ISREMWRD
TBTOP		'Table top'	ISREMWR9
TBVCLEAR		'Table variable clear'	ISREMWRC
Variables			
VCOPY		'Copy variable'	ISREMWV5
VDEFINE		'Variable define'	ISREMWV3
VDELETE		'Variable delete'	ISREMWV4
VERASE		'Variable erase'	ISREMWV8
VGET		'Variable get'	ISREMWV1
VMASK		'Variable mask'	ISREMWV9
VPUT		'Variable put'	ISREMWV2
VREPLACE		'Variable replace'	ISREMWV6

Model Name		Description	Member Name
VRESET		'Variable reset'	ISREMWV7
Workstation			
FILESTAT		'Statistics for a file'	ISREMWX1
FILEXFER		'Upload or download a file'	ISREMWX2
WSCON		'Workstation connection'	ISREMWX3
WSDISCON		'Workstation disconnect'	ISREMWX4

Dialog Tag Language Models

Model Name		Description	Member Name
AB		Action bar	ISREMDAB
AB (example)		Action bar with pull-down choice	ISREMDAG
ABC		Action bar choice	ISREMDA2
ACTION		Action	ISREMDAC
AREA		Area	ISREMDAR
ASSIGNI		Assignment List Item	ISREMDAI
ASSIGNL		Assignment list	ISREMDAL
ATTENTION		Attention	ISREMDAN
ATTR		Attribute	ISREMDAT
BOTINST		Bottom instruction	ISREMDBI
CAUTION		Caution	ISREMDCU
CHDIV		Choice divider	ISREMDCD
AB		Action bar	ISREMDAB
AB		Action bar	ISREMDAB
CHECKI		Validity Check Item	ISREMDCI
CHECKL		Check validity list	ISREMDCL
CHOFLD		Choice data field	ISREMDCF
CHOICE		Selection choice	ISREMDCH
CMD		Command definition	ISREMDCM
CMDACT		Command action	ISREMDCC
CMDAREA		Command area	ISREMDCA
CMDTBL		Command table	ISREMDCT
CMDTBL example		Command table example	ISREMDC1
COMMENT		Comment	ISREMDCN
COMPOPT		Compiler options	ISREMDCO
COPYR		Copyright	ISREMDCR
DA		Dynamic Area	ISREMDDA
DD		Definition description	ISREMDDD
DDHD		Definition description header	ISREMDD1
DIVIDER		Area divider	ISREMDDI
DL		Definition list	ISREMDDL

Model Name		Description	Member Name
DOCTYPE		Document type	ISREMDTP
DT		Definition term	ISREMDDT
DTACOL		Data column	ISREMDDC
DTAFLD		Data field	ISREMDDF
DTAFLDD		Data field description	ISREMDDX
DTHD		Definition term header	ISREMDD2
ENTITY		Entity	ISREMDEN
FIG		Figure	ISREMDFI
FIGCAP		Figure caption	ISREMDFC
GA		Graphic area	ISREMDGA
GRPHDR		Group header	ISREMDGH
HELP		Help panel	ISREMDHE
HELPDEF		Help default	ISREMDHD
HP		Highlighted phrase	ISREMDHP
H1		Heading 1	ISREMDH1
Hn		Heading (H2, H3, H4)	ISREMDHN
INFO		Information region	ISREMDIN
KEYI		Key item	ISREMDKI
KEYL		Key list	ISREMDKL
LI		List item	ISREMDLI
LINES		Lines	ISREMDLN
LIT		Literal	ISREMDLT
LP		List part	ISREMDLP
LSTCOL		List column	ISREMDLC
LSTFLD		List field	ISREMDLF
LSTGRP		List group	ISREMDLG
LSTVAR		List variable	ISREMDLV
M		Mnemonic	ISREMDMN
MSG		Message	ISREMDMS
MSGMBR		Message member	ISREMDMM
NOTE		Note	ISREMDNO
NOTEL		Note list	ISREMDNL
NT		Note	ISREMDNT
OL		Ordered list	ISREMDOL
Panel example		Panel example	ISREMDP1
Panel example		Panel example	ISREMDP2
Panel example		Panel example	ISREMDP3
Panel example		Panel example	ISREMDP4
Panel example		Panel example	ISREMDP5
PANDEF		Panel default	ISREMDPB
PANEL		Panel	ISREMDPA

Model Name		Description	Member Name
PARML		Parameter list	ISREMDPL
PD		Parameter description	ISREMDPD
PDC		Pull-down choice	ISREMDA3
PDSEP		Pull-down separator	ISREMDA4
PG		Paragraph	ISREMDPG
PNLINST		Panel instruction	ISREMDPI
PS		Point-and-shoot	ISREMDPS
PT		Parameter term	ISREMDPT
Region		Region	ISREMDRE
RP		Reference phrase	ISREMDRP
SF		Selection field	ISREMDSF
SL		Simple list	ISREMDSL
SOURCE		Source	ISREMDSO
T		Truncation	ISREMDTR
TOPINST		Top instruction	ISREMDTI
UL		Unordered list	ISREMDUL
VARCLASS		Variable class	ISREMDVC
VARDCL		Variable declaration	ISREMDVD
VARLIST		Variable list	ISREMDVL
VARSUB		Variable substitution	ISREMDVS
WARNING		Warning	ISREMDWA
XLATI		Translate item	ISREMDXI
XLATL		Translate list	ISREMDXL
XMP		Example	ISREMDXM

Appendix C. Programming Interface Macros For Customers

This appendix lists the macros ISPF and SCLM provide as programming interfaces.

Attention: Do not use any ISPF macros other than those listed in this appendix as programming interfaces.

Table 49. DM Component General-Use Programming Macros

ISPCCSID	ISPMXDEF
ISPMDAD	ISPMXED
ISPMEPT	ISPMXEND
ISPMTCM	ISPMXLST
ISPMXDD	

See Chapter 3, "Customizing DM," on page 105 for further information about DM component macros.

Table 50. SCLM General-Use Programming Macros

FLMABEG	FLMGROUP
FLMAEND	FLMINCLS
FLMAGRP	FLMLANGL
FLMALLOC	FLMSYSLB
FLMALTC	FLMTRNSL
FLMATVER	FLMTSEXT
FLMCNTRL	FLMTYPE
FLMCPYLB	

See the *z/OS ISPF Software Configuration and Library Manager Reference* for more information about SCLM macros.

Appendix D. ISPF Data Set Descriptions

The following figures list the target and distribution libraries (data sets) used by ISPF, and their contents.

Table 51. Target Data Set Descriptions

Library	Description
SISPALIB	APL2 workspace library
SISPCLIB	CLIST library
SISPEXEC	REXX exec library
SISPGDES	Swiss German Language-specific panel source library
SISPGDEU	German Language-specific panel source library
SISPGENP	Uppercase English Language-specific panel source library
SISPGENU	English Language-specific panel source library
SISPGJPN	Japanese Language-specific panel source library
SISPGMLI	Non-language-specific panel source library
SISPGUI	ISPF Client/Server workstation code library
SISPHELP	Help library
SISPLOAD	Linklist load library
SISPLPA	LPA load library
SISPMACS	ISPF Macro library
SISPMDES	Swiss German Language-specific message library
SISPMDEU	German Language-specific message library
SISPMENP	Uppercase English Language-specific message library
SISPMENU	English Language-specific message library
SISPMJPN	Japanese Language-specific message library
SISPPDES	Swiss German Language-specific panel library
SISPPDEU	German Language-specific panel library
SISPPENP	Uppercase English Language-specific panel library
SISPPENU	English Language-specific panel library
SISPPJPN	Japanese Language-specific panel library
SISPSAMP	Sample library
SISPSASC	ISPF Client/Server SASC runtime library
SISPSDES	Swiss German Language-specific skeleton library
SISPSDEU	German Language-specific skeleton library
SISPSENP	Uppercase English Language-specific skeleton library
SISPSENU	English Language-specific skeleton library
SISPSJPN	Japanese Language-specific skeleton library
SISPSLIB	Non-language-specific skeleton library
SISPTDES	Swiss German Language-specific table library
SISPTDEU	German Language-specific table library

Table 51. Target Data Set Descriptions (continued)

Library	Description
SISPTENP	Uppercase English Language-specific table library
SISPTENU	English Language-specific table library
SISPTJPN	Japanese Language-specific table library

Table 52. Distribution Data Set Descriptions

Library	Description
AISPALIB	APL2 workspace library
AISPCLIB	CLIST library
AISPEXEC	REXX exec library
AISPGDES	Swiss German Language-specific panel source library
AISPGDEU	German Language-specific panel source library
AISPGENP	Uppercase English Language-specific panel source library
AISPGENU	English Language-specific panel source library
AISPGJPN	Japanese Language-specific panel source library
AISPGMLI	Non-language-specific panel source library
AISPGUI	ISPF Client/Server workstation code library
AISPHELP	Help library
AISPMACS	ISPF Macro library
AISPMDES	Swiss German Language-specific message library
AISPMDEU	German Language-specific message library
AISPMENP	Uppercase English Language-specific message library
AISPMENU	English Language-specific message library
AISPMJPN	Japanese Language-specific message library
AISPMOD1	Executable modules library
AISPPDES	Swiss German Language-specific panel library
AISPPDEU	German Language-specific panel library
AISPPENP	Uppercase English Language-specific panel library
AISPPENU	English Language-specific panel library
AISPPJPN	Japanese Language-specific panel library
AISPSAMP	Sample library
AISPSASC	ISPF Client/Server SASC runtime library
AISPSDES	Swiss German Language-specific skeleton library
AISPSDEU	German Language-specific skeleton library
AISPSENP	Uppercase English Language-specific skeleton library
AISPSENU	English Language-specific skeleton library
AISPSJPN	Japanese Language-specific skeleton library
AISPSLIB	Non-language-specific skeleton library
AISPTDES	Swiss German Language-specific table library
AISPTDEU	German Language-specific table library
AISPTENP	Uppercase English Language-specific table library

Table 52. Distribution Data Set Descriptions (continued)

Library	Description
AISPTENU	English Language-specific table library
AISPTJPN	Japanese Language-specific table library

Appendix E. Accessibility

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products successfully. The major accessibility features in z/OS enable users to:

- Use assistive technologies such as screen readers and screen magnifier software
- Operate specific or equivalent features using only the keyboard
- Customize display attributes such as color, contrast, and font size

Using assistive technologies

Assistive technology products, such as screen readers, function with the user interfaces found in z/OS. Consult the assistive technology documentation for specific information when using such products to access z/OS interfaces.

Keyboard navigation of the user interface

Users can access z/OS user interfaces using TSO/E or ISPF. Refer to *z/OS TSO/E Primer*, *z/OS TSO/E User's Guide*, and *z/OS ISPF User's Guide Vol I* for information about accessing TSO/E and ISPF interfaces. These guides describe how to use TSO/E and ISPF, including the use of keyboard shortcuts or function keys (PF keys). Each guide includes the default settings for the PF keys and explains how to modify their functions.

z/OS information

z/OS information is accessible using screen readers with the BookServer/Library Server versions of z/OS books in the Internet library at:

`www.ibm.com/servers/eserver/zseries/zos/bkserv/`

Notices

This information was developed for products and services offered in the USA.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
USA

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:
INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Mail Station P300
2455 South Road
Poughkeepsie, NY 12601-5400
USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. 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 IBM's application programming interfaces.

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

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

AD/Cycle
APL2
BookManager
BookMaster
C++/MVS
COBOL/370
Common User Access
CUA
DFSMSrmm
DFSMS/MVS
DFSORT
FFST
GDDM
IBM
Language Environment
MVS
MVS/XA
OS/390
RACF
SAA
Systems Application Architecture
Tivoli
VTAM
z/OS

Microsoft, Windows, and Windows NT are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, and service names may be trademarks or service marks of others.

Index

F

G

H

I

J

K

L

M

N

O

P

R

S

T

U

V

W

Z

Readers' Comments — We'd Like to Hear from You

Interactive System Productivity Facility (ISPF)
Planning and Customizing
z/OS Version 1 Release 7.0

Publication No. GC34-4814-04

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	☐	☐	☐	☐	☐

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	☐	☐	☐	☐	☐
Complete	☐	☐	☐	☐	☐
Easy to find	☐	☐	☐	☐	☐
Easy to understand	☐	☐	☐	☐	☐
Well organized	☐	☐	☐	☐	☐
Applicable to your tasks	☐	☐	☐	☐	☐

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? ☐ Yes ☐ No

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

Name

Address

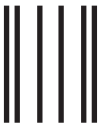
Company or Organization

Phone No.

Readers' Comments — We'd Like to Hear from You
GC34-4814-04

Cut or Fold
Along Line

Fold and Tape | **Please do not staple** | Fold and Tape



NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Corporation
Department J87/D325
555 Bailey Avenue
San Jose, CA
U.S.A. 95141-9989

Fold and Tape | **Please do not staple** | Fold and Tape

Cut or Fold
Along Line

File Number: S370/4300-39
Program Number: 5694-A01

Printed in USA

GC34-4814-04