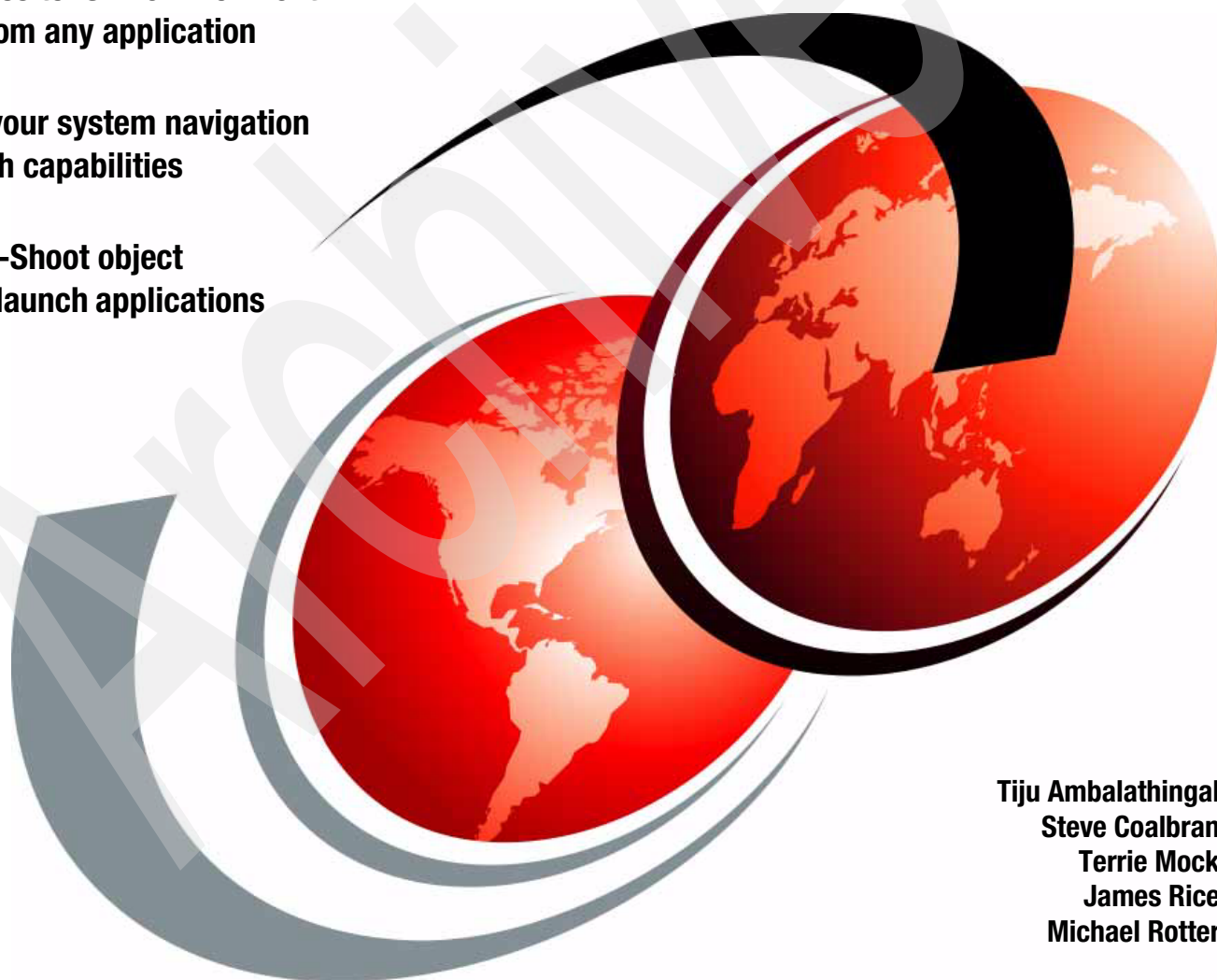


# Improving Your Productivity with the ISPF Productivity Tool V5.9 on z/OS

Easy access to ISPF environment  
objects from any application

Enhance your system navigation  
and search capabilities

Point-and-Shoot object  
names to launch applications



Tiju Ambalathingal  
Steve Coalbran  
Terrie Mock  
James Rice  
Michael Rotter

# Redbooks





International Technical Support Organization

**Improving Your Productivity with the ISPF Productivity  
Tool V5.9 on z/OS**

May 2008

Archived

**Note:** Before using this information and the product it supports, read the information in “Notices” on page vii.

Archived

**First Edition (May 2008)**

This edition applies to ISPF Productivity Tool (ISPF-PT) Version 5, Release 9 for z/OS®.

© Copyright International Business Machines Corporation 2008. All rights reserved.

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

# Contents

<b>Notices</b> .....	vii
Trademarks .....	viii
<b>Preface</b> .....	ix
The team that wrote this book .....	x
Become a published author .....	xii
Comments welcome .....	xii
<b>Chapter 1. Introduction to the ISPF Productivity Tool</b> .....	1
1.1 Definition of terms .....	1
1.2 Who should read this Redbook? .....	2
1.3 ISPF-PT seamless Integration with ISPF .....	2
1.3.1 Learning the ISPF Productivity Tool .....	2
1.3.2 Increased productivity .....	3
1.3.3 Integration with other products .....	3
1.3.4 Object Linking and Embedding (OLE) .....	4
1.3.5 The Object List .....	4
1.3.6 ISPF-PT functional overview .....	5
<b>Chapter 2. The Object List</b> .....	7
2.1 Catalog navigation .....	8
2.1.1 Exploring the system .....	8
2.1.2 Further exploration of system datasets .....	15
2.2 Group execution .....	28
2.2.1 Group execution by number reference .....	28
2.2.2 Group execution by generic reference .....	31
2.3 Permanent Object List .....	34
2.3.1 Permanent Object List for projects .....	34
2.3.2 Creating project 'ELUX' OLIST .....	34
2.3.3 Project ELUX workFlow .....	49
2.4 Object List MEMFIND command .....	64
2.5 Object List FINDTEXT command .....	69
<b>Chapter 3. Member Selection Lists</b> .....	73
3.1 MSL functions .....	74
3.2 Using the ISPF-PT Member Selection List Assist .....	74
3.3 ISPF-PT Search commands .....	76
3.4 Global Find and Global Change command .....	76
3.4.1 Changing the Global ISPF-PT defaults for your session .....	77
3.5 Partitioned Data Set Extended .....	86
3.5.1 Defining PDSEs .....	86

3.6 MSL Point-and-Shoot fields .....	108
3.7 MSL Hotbars .....	111
3.8 Using the TAILOR command .....	113
<b>Chapter 4. Enhanced cut and paste .....</b>	<b>115</b>
<b>Chapter 5. Using IBM File Manager for z/OS with the ISPF Productivity Tool .....</b>	<b>131</b>
5.1 IBM File Manager publications .....	132
5.2 Customizing ISPF-PT to invoke File Manager .....	132
5.3 Using the File Manager with ISPF-PT .....	134
5.4 File Manager walkthrough – applying a Copybook to a VSAM file .....	135
<b>Chapter 6. SCLM integration with the ISPF Productivity Tool .....</b>	<b>141</b>
6.1 Customization to support SCLM .....	142
6.2 SCLM ISPF-PT interface .....	145
6.3 ISPF-PT SCLM walk through .....	149
6.3.1 Creating a new assembler program .....	149
6.3.2 Migrating members to SCLM .....	154
6.3.3 Updating the project SCLM ARCHDEF members .....	158
6.3.4 Building and promoting the SCLM ARCHDEF members .....	161
6.3.5 Removing the obsolete program .....	165
<b>Chapter 7. TSO Command Shell .....</b>	<b>169</b>
7.1 Validating the TSO Command Shell options .....	169
7.2 IPT History Commands List .....	170
7.2.1 Using the TSO Command Shell with the History Command List .....	171
7.3 ISPF-PT Permanent Commands List .....	171
7.3.1 Using the TSO Command Shell with the Permanent Command List .....	172
7.4 TSO Command Shell demonstration .....	172
7.5 Recalling commands .....	177
<b>Chapter 8. First time logon to the ISPF Productivity Tool .....</b>	<b>179</b>
8.1 ISPF-PT LOGON procedure .....	180
8.2 ISPF-PT persistent tables .....	180
8.2.1 ISPF-PT persistent tables defined with first logon .....	180
8.2.2 ISPF-PT persistent tables defined by the system programmer .....	182
8.2.3 No ISPF-PT persistent tables defined by customization .....	182
<b>Chapter 9. Customizing your ISPF keys for Point-and-Shoot .....</b>	<b>185</b>
9.1 Customizing ISPF-PT with View using PF4 .....	185
9.1.1 Defining PF4 for the ISPF-PT View command for SDSF .....	186
9.1.2 Using the Point-and-Shoot commands with a Member Selection List .....	189
<b>Chapter 10. Installing ISPF Productivity Tool V5.9 out-of-the box on the z/OS environment .....</b>	<b>193</b>
10.1 Installation requirements for ISPF-PT V5.9 .....	194
10.2 Contents of the ISPF-PT product package .....	194
10.2.1 Custom-built product delivery offering tapes .....	194
10.3 ISPF Productivity Tool installation .....	196
10.3.1 Sample Jobs .....	197
10.3.2 Allocating SMP/E global zone CSI libraries .....	198
10.3.3 Allocating SMP/E target and distribution zone libraries .....	199
10.3.4 Allocating SMP/E temporary libraries for global zone .....	201
10.3.5 Allocating libraries for target and distribution zone .....	203

10.3.6	Defining global, target, and distribution zone options and creating DDDEF entries	206
10.3.7	Performing SMP/E RECEIVE	217
10.3.8	Performing SMP/E APPLY	218
10.3.9	Performing SMP/E ACCEPT	219
10.4	Customizing the LOGON Proc to invoke ISPF-PT	220
10.4.1	The LPA load library SIQILPA	222
10.4.2	CLIST library SIQICLIB	222
10.4.3	Table library SIQITLIB	222
10.4.4	ISPF-PT load library SIQILOAD	228
10.4.5	CLIST library SIQICLIB	228
10.4.6	Message library SIQIMLIB	228
10.4.7	ISPF-PT panel library SIQIPLIB	228
10.4.8	Skeleton library IQISLIB	228
10.5	Creating RACF profile to use the customized TSO LOGON PROC for ISPF-PT	229
10.6	Using the new TSO LOGON procedure to invoke ISPF-PT	230
10.7	ISPF Productivity Tool maintenance	234
10.8	Ordering ISPF-PT V5.9 for India customers and installing ISPF-PT out-of-the box	235
10.9	Ordering ISPF-PT V5.9 for India customers	235
10.9.1	India/SA Techline - Process	236
<b>Appendix A.</b>	<b>Implementing an OLIST User Object Interface</b>	239
A.1	Proposed method for exploring user-defined objects	240
A.1.1	Creating the development environment and new LOGON procedure	240
A.1.2	Converting the two sample CLISTs into REXX	250
A.1.3	Running the wizard to activate the REXX output exec	264
A.1.4	Extending the exec into a general purpose user defined object	276
A.1.5	Summarizing how the process for those impatient to just-do-it!	276
A.2	Experimenting with other UDOs	276
A.2.1	IQIRDUMY	276
A.2.2	IQIRERRO	277
A.3	More meaningful UDOs	277
<b>Appendix B.</b>	<b>Invoking user commands from an Object List</b>	285
B.1	Example 1: transmitting and receiving files, host-to-host	286
B.1.1	Using the ?XMIT command with an Object List	286
B.1.2	Using the DSLIST command with an Object List	288
B.1.3	Invoking a User Command CLIST or REXX program from the Object List	289
B.1.4	Using a CLIST as a User Command	289
B.1.5	Using a REXX routine as a User Command	291
B.2	Example 2: transmitting and receiving files, host-to-PC-to host	292
<b>Appendix C.</b>	<b>Permanent OLISTs and UNIX System Services objects</b>	299
<b>Appendix D.</b>	<b>Customizing the IBM Personal Communications</b>	305
	Current PCOM mouse settings	305
	Marking the box for cut/paste activities without the mouse	305
	Displaying the pop-up keypad panel without a mouse	306
	Changing PCOM to use the mouse to open and close files	306
	Touring ISPF-PT using the mouse enabled point-and-click function	308
	Restoring the PCOM mouse settings	310

<b>Related publications</b> .....	311
IBM Redbooks .....	311
Other publications .....	311
Online resources .....	311
How to get Redbooks .....	312
Help from IBM .....	312
<b>Index</b> .....	313



# Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

*IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.*

**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.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

## COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.


# Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.shtml>

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

AIX®  
AS/400®  
CICS®  
DB2®  
DFS™  
DFSMS™  
DFSMS/MVS™  
DFSORT™  
FFST™  
GDDM®

IBM®  
IMS™  
Lotus Notes®  
Lotus®  
MVS™  
Notes®  
OS/2®  
OS/390®  
QMFTM  
Redbooks®

Redbooks (logo) ®  
RACF®  
REXX™  
RMF™  
Tivoli®  
VTAM®  
z/OS®

The following terms are trademarks of other companies:

SAP, and SAP logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries.

NOW, and the NetApp logo are trademarks or registered trademarks of NetApp, Inc. in the U.S. and other countries.

Java, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Excel, Windows, and the Windows logo 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.

Other company, product, or service names may be trademarks or service marks of others.

# Preface

In this IBM® Redbooks® publication, we introduce the IBM Interactive System Productivity Facility Productivity Tool (ISPF-PT) Version 5, Release 9 for z/OS®. The ISPF Productivity Tool operates as a seamlessly integrated front-end to ISPF:

- ▶ The ISPF Productivity Tool functionality is available from any panel without modifying any ISPF Primary Option Menus. All ISPF-PT functions are totally integrated. ISPF-PT can perform almost any activity within ISPF or internally invoke the function that can perform the task.
- ▶ The ISPF Productivity Tool combines separately provided ISPF utility functions and new ISPF Productivity Tool features into the Object List (OLIST) and Member Selection List (MSL). The resulting member, data set, and object lists become powerful platforms where you can perform many tasks without navigating to other utilities.
- ▶ The ISPF Productivity Tool relates objects to applications similarly to the way a PC performs Object Linking and Embedding (OLE). By extending the data-set objects that ISPF uses to other object classes, ISPF-PT lets you specify the object to be processed and the action that is performed (such as EDIT or BROWSE). The facility appropriate to the object class for the action you have requested is invoked automatically.
- ▶ The ISPF Productivity Tool provides extensive search capabilities that are both rapid and intuitive. You can easily search for volumes, data sets, members, and text within members. The ISPF Productivity Tool also furnishes automatic drill-down system navigation to examine volumes, data sets, and members.
- ▶ The ISPF Productivity Tool extends the ISPF Action Bar with options that provide access to new functionality so that you do not have to learn new commands or syntax. In addition to the ISPF point-and-shoot capabilities, the ISPF Productivity Tool provides new concepts, such as Hotbars (user-defined fields that execute commands), field-sensitive areas in MSLs and OLISTs, automatic recognition of a data set name on any ISPF panel as a parameter to BROWSE, EDIT, VIEW, or parameters within any Time Sharing Option (TSO) command.
- ▶ The ISPF Productivity Tool provides integrated and enhanced IBM Software Configuration and Library Manager (SCLM) support within the standard member and data set lists. SCLM is a source library management component of ISPF that provides change control, multiple source versions, auditing, a built-in make facility, and automatic check-in and sign-out using standard libraries (Partitioned Data Sets (PDS) and (PDSE) Partitioned Data Set Extended).
- ▶ The ISPF Productivity Tool includes built-in interfaces to various IBM and ISV products.

This book is intended as a supplement to existing product manuals. We start with an overview of the main ISPF-PT concepts and facilities. Next are detailed chapters, each dedicated to a major ISPF-PT function, such as: Object-Lists, Member-Selection-Lists, enhanced Point-and-Shoot, EDIT CUT and PASTE, ISPF-PT with IBM File Manager, ISPF-PT with IBM SCLM, TSO Shell, ISPF-PT installation and customization, and an example of implementing an ISPF-PT User-Defined-Object-List. A set of appendixes explore deeper into selected topics.

Practical scenarios, accompanied by detailed screen captures and coding examples, demonstrate how to take advantage of the ISPF-PT enhanced functionality at every case.

## The team that wrote this book

This book was produced by a team of specialists from around the world working at the International Technical Support Organization, San Jose Center.



**Tiju Ambalathingal** is an Advisory System Analyst at IBM GBS Global Delivery in Bangalore, India. Tiju is currently the Technical Solution Lead for Enterprise Application Modernization associated with AIS. His current responsibilities include providing technical solution outline, approach, and architect solutions to modernize mainframe applications for enterprises around the globe. Tiju joined IBM in December 2002 as an Application Programmer. He has seven years of experience in mainframe application development, maintenance, and migration. In this tenure, he has held the role of Technical Project Leader.



**Steve Coalbran** is an accredited Senior IT Specialist with IBM Global Business Services in Stockholm, Sweden. Steve has close to 30 years of application development and end-to-design experience within IBM. From 1978 he worked in the UK, transferring to Sweden in 1999. Steve specializes in the development of innovative solutions and tools that increase the productivity and efficiency of developers and production systems. He has worked primarily with MVS™ from OS/360 through z/OS and, to a lesser extent, with CMS and OS/2®, AS/400®, DOS, and Windows®. In the past years, his work in Sweden related to the implementation of customized IBM product line solutions, replacing and improving on existing legacy software. Steve especially focused on tailoring ISPF and REXX™ in products within the AD Tools set, working closely with both Software Sales and ISPF development.



**Terrie Mock** is currently a Systems Administrator at Northwest Regional Data Center in Tallahassee Florida. Northwest Regional Data Center serves several Universities, K-12 school districts, State agencies, and county and local governments around the state of Florida. Terrie started her data processing career as an operator with the Florida Department of Labor and has steadily advanced to the level of Systems Programmer. She has worked for the State of Florida for 26 years. Her current responsibilities include installing, testing, and maintaining a variety of mainframe software packages including the ISPF Productivity Tool.



**James Rice** is a Certified IT Specialist, residing near Tampa, Florida. Jim is a z/OS Problem Determination Tools Technical Sales Specialist who works in the Above Country region. Jim joined IBM in 1984 as an Application Developer. He has over 30 years of experience with IBM mainframe computers. He has held job positions as an Application Developer, DB2® Database Administrator, Test Manager, administrator of test z/OS environments, and Technical Sales Specialist for the IBM Problem Determination Tools.



**Michael Rotter** is a Senior Software Engineer at IBM Tivoli® in New Jersey, USA. Mike is currently serving as the Chief Architect/Developer/Level-3 Support for the ISPF Productivity Tool product. He joined IBM in July 2005 during the acquisition (by IBM) of his former company ISOGON. Mike has over 30 years of experience with system-level software development in large system environments with MVS, OS/390®, and z/OS. Prior to joining IBM, Mike held software development leadership positions at ISOGON, STERLING Software, AT&T Federal Systems, E.F. Hutton, J.P. Morgan Bank, Informatics, and more.

Thanks to the following people for their contributions to this project:

Eric Chabert, IBM, Software Sales IT Specialist, Montpellier, France

Howard Givner, IBM, Software Developer, Tivoli, NYC, NY

Amit Kr Joshi, IBM, System Administrator: Mainframe, Bangalore, India

Deanna Polm, IBM, Residency Coordinator, San Jose, CA

and

Joe DeCarlo, Manager, Special Projects, International Technical Support Organization,  
San Jose, CA

## Become a published author

Join us for a two- to six-week residency program! Help write a book dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You will have the opportunity to team with IBM technical professionals, Business Partners, and Clients.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you will develop a network of contacts in IBM development labs, and increase your productivity and marketability.

Find out more about the residency program, browse the residency index, and apply online at:  
[ibm.com/redbooks/residencies.html](http://ibm.com/redbooks/residencies.html)

## Comments welcome

Your comments are important to us!

We want our books to be as helpful as possible. Send us your comments about this book or other IBM Redbooks in one of the following ways:

- Use the online **Contact us** review Redbooks form found at:

[ibm.com/redbooks](http://ibm.com/redbooks)

- Send your comments in an e-mail to:

[redbooks@us.ibm.com](mailto:redbooks@us.ibm.com)

- Mail your comments to:

IBM Corporation, International Technical Support Organization  
Dept. HYTD Mail Station P099  
2455 South Road  
Poughkeepsie, NY 12601-5400

# Introduction to the ISPF Productivity Tool

In this chapter, we provide a general description of the ISPF Productivity Tool. We discuss the features, functions, and facilities that ISPF Productivity Tool provides to enhance ISPF.

## 1.1 Definition of terms

This document references several acronyms:

<b>ISPF-PT</b>	<i>ISPF Productivity Tool:</i> This Redbook is dedicated to the ISPF Productivity Tool. The ISPF Productivity Tool was formerly known as SPIFFY.
<b>OLIST</b>	<i>Object list:</i> An Object List is a series of references to your data stores. Using an Object List, you can organize all the data stores required for your project. An Object List may contain sequential, VSAM, PDS, or PC files, DB2 tables, imbedded OLISTS, SCLM hierarchies, Panvalet or Librarian files, Unix System Service files, DD name references, and user defined processes.
<b>PDS</b>	<i>Partitioned Data set:</i> A data set organization for holding collections of related records in groups called "members".
<b>PDSE</b>	<i>Partitioned Data set Extended:</i> The PDSE is identical to the PDS organization, but has additional sharing and space management capabilities.
<b>MSL</b>	<i>Member Selection List:</i> A member selection list is similar to the ISPF member list presented when opening a PDS or PDSE. However, an ISPF-PT Member Selection List provides enhancements, which perform complex functions, minimizing keystrokes and navigation required to perform a task.

<b>SCLM</b>	<i>IBM Software Configuration and Library Manager:</i> SCLM is a z/OS Software Configuration Management product that manages changes to your application data, performs auditing and versioning, and controls the movement of application data from one set of libraries to the next.
<b>UDO</b>	<i>User Defined Objects:</i> User Defined Objects provide the functionality of your choice. They are installation defined and tailored to handle the user objects. You may specify a User Defined Object as an Object List entry with a leading greater-than sign. For example, the Object List command below invokes the MYOBJECT User Defined Object.

## 1.2 Who should read this Redbook?

The IBM ISPF Productivity Tool Redbooks publication addresses the questions of ISPF-PT end-users. However, anyone that uses ISPF can benefit from ISPF-PT. Frequent users can benefit the most.

In this Redbook, we also provide information for the product installation support team.

## 1.3 ISPF-PT seamless Integration with ISPF

The ISPF Productivity Tool works as *an extension of* ISPF, which means that you do not have to initiate a separate product within ISPF to use the ISPF Productivity Tool. They work together as one product, for example, you may use OLIST commands to locate files, use MSL commands to locate the correct PDS member, edit the member using ISPF, and use ISPF and ISPF-PT commands during the edit session.

### 1.3.1 Learning the ISPF Productivity Tool

You do not have to spend hours learning ISPF-PT to use it. ISPF functions are still available after the ISPF-PT product installation. As you learn more about ISPF-PT, you can begin to become more productive using the new shortcuts and tools that are available.

ISPF-PT provides many features that make it easy to learn:

- ▶ The “ISPF-PT?” command presents a list of ISPF-PT shortcuts.
- ▶ Action bars are available to guide you to the correct command.
- ▶ When using an Object List, the ASSIST (or A) command summarizes each Object List primary and line command that are available:
  - You can also display the command syntax and examples by entering ASSIST **olistcommand** (or A **olistcommand**), for example, A FINDTEXT provides the details of the FINDTEXT command.
- ▶ The Member Selection List ASSIST command summarizes each MSL primary and line-command syntax and examples of each MSL command:
  - You can also display the command syntax and examples by entering ASSIST **mslcommand** (or A **mslcommand**), for example, A WHERE provides the details of the WHERE command.



The ISPF Productivity Tool invocation uses an ISPF LOGON procedure that is designed for ISPF-PT. You can turn off ISPF-PT for the duration of your ISPF session, by entering `ISPF-PTOFF;=X` without leaving ISPF. To turn it back on, enter `ISPF-PTON;=X`.

### 1.3.2 Increased productivity

A development organization's time is valuable. Programmers often write shortcuts using CLIST or REXX programs to help with repetitive tasks and to share these tools with their peers. ISPF-PT standardizes the shortcuts and tools that you need to be productive with ISPF. Think of the time this saves everyone!

Have you ever seen the "No space in directory" message when you add members to a PDS? Think of the number of screens and keystrokes ISPF requires when it allocates a new PDS, copies the members, and renames the data set. With ISPF-PT, you see a confirmation window to expand the PDS directory. Press Enter, and the PDS directory is compressed, without leaving the edit session of your PDS member.

ISPF-PT uses Hotbars, Field-sensitive areas in the Object List, Member Selection Lists, and 47 shortcut commands to minimize the time that is required to get the job done.

Using a single command, invoked from an Object List, you can search multiple files for a member name, data within a member, or data in sequential files from an Object List. With ISPF-PT, you can make Global changes to PDS or PDSe members.

These are just a few of the time saving tips that we provide in this Redbook. ISPF-PT is packed full of shortcuts that make the most use of your time when you are working on a z/OS mainframe.

### 1.3.3 Integration with other products

The ISPF Productivity Tool enhances the standard TSO/ISPF to improve productivity and performance. ISPF-PT integrates seamlessly into the ISPF product and extends the functions that are provided by ISPF to other data set objects, including VSAM files, SCLM libraries, Librarian or Panvalet libraries, z/OS UNIX® System Services, PC Files, DB2 tables, and other object classes.

Using the ISPF-PT customization wizard, you can define the CLIST used by the various products. Then you can access the following files directly from an Object List:

- ▶ VSAM Files:
  - IBM File Manager for z/OS
  - Other third-party products
- ▶ Software Configuration Management Systems:
  - IBM Software Configuration and Library Manager
  - Other third-party products
- ▶ DB2 Tables
  - Other third-party products

### 1.3.4 Object Linking and Embedding (OLE)

With ISPF, you might need to navigate using many panels and products:

- ▶ To search for data and browse or edit data
- ▶ When the data is stored on DB2 tables, VSAM files, Panvalet or Librarian files, UNIX System Services files or PC Files.

ISPF-PT relates objects to applications similarly to the way a PC performs Object Linking and Embedding (OLE).

OLE is the process by which a specified object (such as a VSAM file) is registered to a method (for example, File Manager) that performs a task that the user requests. Usually, it is sufficient to select the object in order to take the appropriate action. ISPF-PT resolves the class of object, the desired action, and determines what program or product will perform the requested function.

### 1.3.5 The Object List

ISPF-PT introduces us to a new concept – an Object List. An Object List is a data store reference that you can use to organize your work into a single folder. With an Object List, you can organize all of the required files into a single project folder, which is called an OLIST.

Consider the following example. You are developing and testing an application. You created an Object List with 10 objects, as shown in Table 1-1.

Table 1-1 MYPROJ Object List

Object	Type of Object	Object Name
1	SCLM Hierarchy	<USBAPP.DEV1.SOURCE
2	JCL	'MYID.JCLLIB'
3	PROC	'MYID.PROCLIB'
4	DB2 Table	-USBAPP.CUSTORDER
5	SEQ File	'MYID.CUST.ORDER'
6	SEQ File	'MYID.CUST.ORDER.REPORT'
7	VSAM File	'MYID.CUST.MASTER.FILE'
8	USS File	)/U/MYID/Project Documentation
9	Imbedded Olist	'USBAPP*PROJECT'
10	Control Data	'MYID.CONTROL'

The Object List in Table 1-1 represents all of the files that you need to modify an application for a customer change request. Think of this Object List as a “Project File” for the customer change request. Using the SCLM Hierarchy, you can easily reference all of the levels in the SCLM tree and all SCLM library types. Files that contain the test JCL, PROC, and Control members are contained in the OLIST. Project documentation may reside in a UNIX System Services file. The application also uses VSAM and DB2 files, which you can easily reference. With this OLIST, you can access any of your project files by simply selecting the file you want. The Object Linking and Embedding process invokes the correct utility to reference the file.

An Object List is persistent. Therefore, changes that you make to the Object List are not lost when you log off the TSO session.

### 1.3.6 ISPF-PT functional overview

ISPF-PT offers the following functions:

- ▶ **Enhanced DSLIST ISPF (3.4)**  
Create and save lists of projects that are related datasets for quick access. ISPF-PT provides a list of LPA, Linklist, and TSO file allocations.
- ▶ **Point-and-Shoot**  
Provides Extensive Point-and-Shoot capabilities on data sets and member names, Column headings, Hotbars, and Action Fields.
- ▶ **TSO Command Shell enhancements**  
Create permanent TSO command lists and execute TSO commands from any ISPF panel.
- ▶ **Enhanced clipboards**  
Temporary and Permanent numbered or named clipboards.
- ▶ **Automatic PDS compression**  
Invokes the **compress** command when the PDS is full.
- ▶ **Automatic PDS directory expansion**  
Invokes the EXPDIR command when the PDS directory are full.
- ▶ **SCLM, Librarian, Panvalet support**  
Edit and browse files from your ISPF panels. There is no need to navigate to Library Management products.
- ▶ **Enhanced data set patterns**  
Data set patterns do not require the period, for example, SYS\*S is a valid data set name pattern request.
- ▶ **Provides extensive search capabilities**  
Locate volumes, data sets, members and text.
- ▶ **Automatic integrated access**  
Edit and browse VSAM files, DB2 tables, HFS files, PC files, catalog levels, and other objects. There is no need to navigate to other products.
- ▶ **Provides extremely fast results**  
Some ISPF-PT functions invoke a single utility to process a request with multiple files.

Archived

## The Object List

The ISPF Productivity Tool introduces us to a new concept – an Object List, as Jim Worker is experiencing in Figure 2-1. An Object List is a data store reference, used to organize your work into a single folder. With an Object List, you can organize all of the required files into a single project folder, which is called an OLIST.



Figure 2-1 Jim Worker works with the Object List

ISPF-PT is really useful in helping you explore the system using the system catalog. Imagine that you are a new user to a system or investigating in an unfamiliar area. The catalog is the obvious starting point. In this section, we examine the concepts of *Catalog navigation* and concentrate on some of the *Group execution* techniques.

## 2.1 Catalog navigation

In this section, we explore the new system to which we were assigned. We start by reviewing our environment, which includes all files that are currently allocated to our session.

### 2.1.1 Exploring the system

Let us start by creating a temporary OLIST, which we can do using the OLDDNAME command, which has a shortcut of OLDD. This can be supplied with a particular DDname (for example, OLDD SYSHELP, to see all of the TSO Help libraries); however, here we will omit the parameter to see all of the DDnames that are allocated, as shown in Figure 2-2.

```
Menu Utilities Compilers Options Status Help
z/OS Primary Option Menu
Option ==> OLDD
0 Settings      Terminal and user parameters      User ID . : SE16661
1 View          Display source data or listings    Time. . . : 08:57
2 Edit          Create or change source data      Terminal. : 3278
3 Utilities     Perform utility functions             Screen. . : 1
4 Foreground    Interactive language processing        Language. : ENGLISH
5 Batch         Submit job for language processing      Appl ID . : ISP
6 Command       Enter TSO or Workstation commands      TSO logon : SYSUSER
7 Dialog Test   Perform dialog testing                 TSO prefix: SE16661
P IBM Products  IBM program products                  System ID : ZT01
10 SCLM         SW Configuration Library Manager      MVS acct. : SYS0000
11 Workplace    ISPF Object/Action Workplace          Release . : ISPF 5.8
12 z/OS System  z/OS system programmer applications
13 z/OS User    z/OS user applications

Enter X to Terminate using log/list defaults
```

Figure 2-2 Issuing an OLDD command without a DDname parameter

Issuing an OLDD command without a DDname parameter might display something similar to Figure 2-3 on page 9.

File Edit Find Display Populate Settings Menu Util Test Help Exit				
----- ALLOCATION LIST ----- "A" will display assist				
-IPT- OLIST (B)	====>			SCROLL ==> CSR
Command				SET UPDATE
Hotbar: FLIP	CLRVL	FILLVOL	REFRESH	CUT
				*TEMPORARY LIST*
TSO PARMS ==>				
Command	Member	Numbr	Data Set Names / Objects	Volume
-STEPLIB		1	'IMS.V9R1.SDFSRESL'	Z00D01
		2	'COBOL.V3R4.SIGYCOMP'	Z00D01
-ADMP		3	'GDDM.SADMPF'	Z00S01
-ADMPROJ		4	'GDDM.SADMMAP'	Z00S04
-SYSLBC		5	'SYS1.BROADCAST'	Z00CAT
-SYSPRINT		6	'NULLFILE'	
-SYSTEM		7	'NULLFILE'	
-SYSIN		8	'NULLFILE'	
-ISPPROF		9	'SE16661.ISPF.ISPPROF'	SHAR02
-ADMIMG		10	'GDDM.SADMMAP'	Z00S04
-ADMGIMP		11	'GDDM.SADMMAP'	Z00S04
-ADMGGMAP		12	'GDDM.SADMMAP'	Z00S04
-SYSHELP		13	'SYS1.HELP'	Z00RES
		14	'SYS1.SEDGHLPI'	Z00RES
		15	'ISP.SISPHELP'	Z00RES
		16	'SYS1.HELPEP'	Z00RES
		17	'TCPIP.SEZAHLP'	Z00RES
-ISPLLIB		18	'IPT.V5R9.SIQLD'	Z00D14
		19	'CENTER.LOADLIB'	Z00CAT
		20	'SYS1.DFQLLIB'	Z00RES
		21	'SYS1.DGTLLIB'	Z00RES
		22	'DEBUG.V7R1.SEQAMOD'	Z00D10
		23	'CCCA.V2R1.SABJMOD1'	Z00D02

Figure 2-3 Displaying the Allocation List from an OLDD command without a DDname parameter- Part 1

Scroll down to view more, as shown in Figure 2-4.

File Edit Find Display Populate Settings Menu Util Test Help Exit				
----- ALLOCATION LIST ----- Row 24 to 46 of 174				
-IPT- OLIST (B)	====>			SCROLL ==> CSR
Command				SET UPDATE
Hotbar: FLIP	CLRVL	FILLVOL	REFRESH	CUT
				*TEMPORARY LIST*
TSO PARMS ==>				
Command	Member	Numbr	Data Set Names / Objects	Volume
		24	'QMF.V7R1M0.SDSQLOAD'	Z00D03
		25	'DB2.V8R1.SDSNLOAD'	Z00D06
-ADMGDF		26	'SYS1.SCBDHENU'	Z00RES
-SYSPROC		27	'GDDM.SADMMAP'	Z00S04
		28	'IPT.TEST.CLIB'	Z00D16
		29	'IPT.V5R9.SIQLCLIB'	Z00D17
		30	'CENTER.CLIST'	Z00CAT
		31	'SYS1.SBLSCLIO'	Z00RES
		32	'BOOKMAN.SEOYCLIB'	Z00S03
		33	'CBC.SCBCUTL'	Z00S04
		34	'SYS1.DGTCLIB'	Z00RES
		35	'DFSORT.SICECLIB'	Z00S04
		36	'FFST.SEPWCENU'	Z00S02
		37	'SYS1.SCBDCLST'	Z00RES
		38	'ISP.SISPCLIB'	Z00RES
		39	'RMF.SERBCLS'	Z00S04
		40	'SYS1.HRFCLST'	Z00RES
		41	'GIM.SGIMCLSO'	Z00RES
		42	'ICQ.ICQCCLIB'	Z00S02
		43	'CCCA.V2R1.SABJCLST'	Z00D09
		44	'DEBUG.V7R1.SEQAEXEC'	Z00D05
		45	'DB2CFG.DB2TOOLS.CLIST'	DB2G01
		46	'DB2.V8R1.SDSNCLST'	Z00D04

Figure 2-4 Issuing an OLDD command without a DDname parameter- Part 2

Continue to scroll down, as shown in Figure 2-5 on page 10.

File Edit Find Display Populate Settings Menu Util Test Help Exit									
-IPT- OLIST (B) ----- ALLOCATION LIST -----					Row 47 to 69 of 174				
Command ==>					SCROLL ==> CSR				
Hotbar: FLIP <b>CLRVOL</b> PALLVOL REFRESH UTIL CUT SET UPDATE					*TEMPORARY LIST*				
TSO PARMS ==>									
Command	Member	Numbr	Data Set Names / Objects		Volume				
-SYSEXEC		47	'SYS1.SBPXEXEC'		Z00RES				
		48	'BOOKMAN.SEOYCLIB'		Z000S3				
		49	'SYS1.SEDGE1'		Z00RES				
		50	'FFST.SEPWSRC1'		Z000S4				
		51	'ISP.SISPEXEC'		Z00RES				
		52	'ISF.SISFEXEC'		Z00RES				
		53	'DCE.SEUVEEXEC'		Z000S1				
		54	'DFS.SIOEEXEC'		Z00RES				
		55	'INFOPT.SAOPXEXEC'		Z000S1				
		56	'QMF.V7R1M0.SDSQEXCE'		Z00D01				
		57	'FILEMGR.V7R1.SFMNEXEC'		Z00D03				
		58	'LDAP.SGLDEXEC'		Z000S1				
		59	'GDDM.SADMSYM'		Z000S1				
-ADMSYMBL		60	'CENTER.PARMLIB'		Z00CAT				
-SYSTCPD	TCPDATA	61	'SE16661.ISPF.ISPPROF'		SHAR02				
-SMPTABL		62	'GIM.CIDTABL'		Z00RES				
-CIDTABL		63	'IPT.V5R9.SIQTILIB'		Z00D18				
-IPITLIB		64	'ISP.SISPSAMP'		Z00RES				
-ISPILIB		65	'SE16661.ISPF.ISPPROF'		SHAR02				
-ISPTABL		66	'NULLFILE'						
-SDSFDUMP		67	'ICQ.ICQAATAB'		Z000S1				
-ICQAATAB		68	'ICQ.ICQABTAB'		Z000S4				
-ICQABTAB		69	'ICQ.ICQANTAB'		Z000S3				
-ICQANTAB									

Figure 2-5 Issuing an OLDD command without a DDname parameter- Part 3

There are a lot of libraries allocated at this installation, so we stopped scrolling here. Yours will be different, that is a certainty.

Seeing all of the libraries that are allocated, we must focus on particular subsets, one at a time. We can start by looking at the first library in the STEPLIB concatenation, as shown in Figure 2-3 on page 9.

To focus on just the datasets that match, we could create a temporary OLIST using IMS™ as a pattern.

## Intuitive searching of generic patterns

This intuitive searching of generic patterns goes much further than the standard ISPF.

It is important to note that if ISPF-PT encounters a pattern that fits the standard ISPF pattern syntax, it uses ISPFs own generic search method. If, however, it does not meet this, but does meet the ISPF-PT generic pattern syntax, it will use its own method.

The patterns follow a common and well defined system, similar to products such as: ISPF, DFDSS, and VSAM but without some of their restrictions:

- ▶ \* any number of characters, 0+
- ▶ % any character (including dots)

Because the ISPF-PT system is so much less restrictive, it is preferable to force this type of search by ensuring that the pattern does not conform to the ISPF standard. Table 2-1 on page 11 shows some examples.



Table 2-1 Patterns

Generic pattern	Datasets found by intuitive search
SYS%**CLIB -or- SYS%.****CLIB	'SYS030.EPS.PROCLIB' 'SYS1.DGTCLIB' 'SYS1.MACLIB' 'SYS1.PROCLIB' 'SYS1.PROCLIB.ZT00PLEX.MOP' 'SYS1.SISTCLIB' 'SYS1.SPROCLIB' 'SYS1.SVCLIB' 'SYS2.PROCLIB' 'SYS2.PROCLIB.FLEXES' 'SYS2.PROCLIB.FO' 'SYS2.PROCLIB.ZT00PLEX.MOP'
SYS1**LIB*.%%%	'SYS1.PARMLIB.ZT00PLEX.MOP' 'SYS1.PROCLIB.ZT00PLEX.MOP'

Next, we look at data sets, beginning with IMS as shown in Figure 2-6.

```

Menu  Utilities  Compilers  Options  Status  Help

z/OS Primary Option Menu

Option ===> OL IMS*

0 Settings      Terminal and user parameters      User ID . : SE16661
1 View          Display source data or listings  Time. . . : 09:15
2 Edit          Create or change source data  Terminal. : 3278
3 Utilities      Perform utility functions          Screen. . : 1
4 Foreground    Interactive language processing  Language. : ENGLISH
5 Batch          Submit job for language processing  Appl ID . : ISP
6 Command        Enter TSO or Workstation commands  TSO logon : SYSUSER
7 Dialog Test    Perform dialog testing              TSO prefix: SE16661
P IBM Products  IBM program products                System ID : ZT01
10 SCLM          SW Configuration Library Manager    MVS acct. : SYS0000
11 Workplace     ISPF Object/Action Workplace        Release . : ISPF 5.8
12 z/OS System   z/OS system programmer applications
13 z/OS User      z/OS user applications

Enter X to Terminate using log/list defaults

```

Figure 2-6 Temporary Object List matching a pattern

When it has more than 1000 entries, ISPF-PT prompts you after each 1000 of the first 3000 as shown in Figure 2-7 on page 12, and another prompt is issued when just the 1596 rows are pending.

```

-IPT----- Object List -----
COMMAND    ==>

You have requested filling the object list with
the 'IMS*' level.
There are more than 1000 entries in the object list.

LEVEL: IMS*

Press ENTER to proceed or END key to stop now.

IQIP966 1000 Rows in OLIST. 1596 Rows pending.

```

Figure 2-7 Prompting for more than 1000 entries in the Object List

Finally the whole list is displayed, as shown in Figure 2-8.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (B) ----- LEVEL IMS* ----- "A" will display assist
Command ==> ----- SCROLL ==> CSR
Hotbar: FLIP CLRVOL FILLVOL REFRESH UTIL CUT SET UPDATE
*TEMPORARY LIST*

TSO PARMS ==>
Command Member Numbr Data Set Names / Objects Volume
-----
1 'IMS'
2 'IMS.APPC.ERRORLOG' MIGRAT
3 'IMS.APPC.JOBLOG' MIGRAT
4 'IMS.APPC.SYSUDUMP' MIGRAT
5 'IMS.CELDCUST.D01' MIGRAT
6 'IMS.CELDCUST.D01.DATA' MIGRAT
7 'IMS.CELDCUSX.D01' MIGRAT
8 'IMS.CELDCUSX.D01.DATA' MIGRAT
9 'IMS.CELDCUSX.D01.INDEX' MIGRAT
10 'IMS.CELDIAL.COPYLIB' Z00D08
11 'IMS.CELDIAL.SOURCE' Z00D10
12 'IMS.CELDPROD.D01' MIGRAT
13 'IMS.CELDPROD.D01.DATA' MIGRAT
14 'IMS.CELDPROX.D01' MIGRAT
15 'IMS.CELDPROX.D01.DATA' MIGRAT
16 'IMS.CELDPROX.D01.INDEX' MIGRAT
17 'IMS.JAVADEMO.CNTL' Z00D18
18 'IMS.JAVADEMO.JAVADB1'
19 'IMS.JAVADEMO.JAVADB1.DATA' Z00D02
20 'IMS.V9R1.ADFSLOAD' Z00D03
21 'IMS.V9R1.ADFSMACT' Z00D12
22 'IMS.V9R1.HFS' Z00D02
23 'IMS.V9R1.INSTALIB' Z00D01

```

Figure 2-8 Complete object is list displayed after all prompts are cleared

Now we are interested in Java™, so we want to see just the datasets in this list that contain the string 'JAVA', so we use the FILTER command, as shown Figure 2-9 on page 13.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-IPT- OLIST (B) ----- LEVEL IMS* ----- "A" will display assist
Command ==> FILT JAVA SCROLL ==> CSR
Hotbar: FLIP CLRVOL FILLVOL REFRESH UTIL CUT SET UPDATE
*TEMPORARY LIST*

TSO PARMS ==>
Command Member Numbr Data Set Names / Objects Volume
-----
1 'IMS'
2 'IMS.APPC.ERRORLOG' MIGRAT
3 'IMS.APPC.JOBLOG' MIGRAT
4 'IMS.APPC.SYSUDUMP' MIGRAT
5 'IMS.CELDCUST.D01' MIGRAT
6 'IMS.CELDCUST.D01.DATA' MIGRAT
7 'IMS.CELDCUSX.D01' MIGRAT
8 'IMS.CELDCUSX.D01.DATA' MIGRAT
9 'IMS.CELDCUSX.D01.INDEX' MIGRAT
10 'IMS.CELDIAL.COPYLIB' Z00D08
11 'IMS.CELDIAL.SOURCE' Z00D10
12 'IMS.CELDPROD.D01' MIGRAT
13 'IMS.CELDPROD.D01.DATA' MIGRAT
14 'IMS.CELDPROX.D01' MIGRAT
15 'IMS.CELDPROX.D01.DATA' MIGRAT
16 'IMS.CELDPROX.D01.INDEX' MIGRAT
17 'IMS.JAVADEMO.CNTL' Z00D18
18 'IMS.JAVADEMO.JAVADB1'
19 'IMS.JAVADEMO.JAVADB1.DATA' Z00D02
20 'IMS.V9R1.ADFSLOAD' Z00D03
21 'IMS.V9R1.ADFSMACT' Z00D12
22 'IMS.V9R1.HFS' Z00D02
23 'IMS.V9R1.INSTALIB' Z00D01

```

Figure 2-9 Using the Object List FILTER command to scope the results

Figure 2-10 shows the results of the FILTER command.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-IPT- OLIST (B) ----- LEVEL IMS* ----- Row 17 from 2596
Command ==> FILT JAVA SCROLL ==> CSR
Hotbar: FLIP CLRVOL FILLVOL REFRESH UTIL CUT SET UPDATE
*TEMPORARY LIST*

TSO PARMS ==>
Command Member Numbr Data Set Names / Objects *FILTER* Volume
-----
17 'IMS.JAVADEMO.CNTL' Z00D18
18 'IMS.JAVADEMO.JAVADB1' Z00D02
19 'IMS.JAVADEMO.JAVADB1.DATA' Z00D02
----- END OF LIST -----

```

Figure 2-10 Object list FILTERing only datasets containing the string 'JAVA'

Figure 2-11 on page 14 shows us all of the ones that we just excluded using the FLIP command.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (B) ----- LEVEL IMS* ----- Row 17 from 2596
Command ==> FLIP SCROLL ==> CSR
Hotbar: FLIP CLRVOL FILLVOL REFRESH UTIL CUT SET UPDATE
*TEMPORARY LIST*

TSO PARMS ==>
Command Member Numbr Data Set Names / Objects *FILTER* Volume
-----
17 'IMS.JAVADEMO.CNTL' 200D18
18 'IMS.JAVADEMO.JAVADB1' 200D02
19 'IMS.JAVADEMO.JAVADB1.DATA' 200D02
----- END OF LIST -----

```

Figure 2-11 Reversing the list with the FLIP command

Figure 2-12 displays all of the **IMS\*** datasets except the ones that contain the string JAVA.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (B) ----- LEVEL IMS* ----- Row 1 from 2596
Command ==> CLRVOL FILLVOL REFRESH UTIL CUT SET UPDATE
Hotbar: FLIP CLRVOL FILLVOL REFRESH UTIL CUT SET UPDATE
*TEMPORARY LIST*

TSO PARMS ==>
Command Member Numbr Data Set Names / Objects *EXCLUDE* Volume
-----
1 'IMS' MIGRAT
2 'IMS.APPC.ERRORLOG' MIGRAT
3 'IMS.APPC.JOBLOG' MIGRAT
4 'IMS.APPC.SYSUDUMP' MIGRAT
5 'IMS.CELDCUST.D01' MIGRAT
6 'IMS.CELDCUST.D01.DATA' MIGRAT
7 'IMS.CELDCUSX.D01' MIGRAT
8 'IMS.CELDCUSX.D01.DATA' MIGRAT
9 'IMS.CELDCUSX.D01.INDEX' MIGRAT
10 'IMS.CELDIAL.COPYLIB' 200D08
11 'IMS.CELDIAL.SOURCE' 200D10
12 'IMS.CELDPROD.D01' MIGRAT
13 'IMS.CELDPROD.D01.DATA' MIGRAT
14 'IMS.CELDPROX.D01' MIGRAT
15 'IMS.CELDPROX.D01.DATA' MIGRAT
16 'IMS.CELDPROX.D01.INDEX' MIGRAT
20 'IMS.V9R1.ADFSLOAD' 200D03
21 'IMS.V9R1.ADFSLOAD' 200D12
22 'IMS.V9R1.HFS' 200D02
26 'IMS.V9R1.MATRIX' 200D01
27 'IMS.V9R1.MATRIX' 200D10
28 'IMS.V9R1.MATRIX' 200D07
29 'IMS.V9R1.MATRIX' 200D05

IQIP1012 Visible rows excluded. Invisible rows revealed.

```

Figure 2-12 FLIPped Object List showing only datasets not containing the string 'JAVA'

**Note:** The **\*EXCLUDE\*** or (as in the panel display before) the **\*FILTER\*** are 'hot'. At any time you can return to the base IMS\* list by placing your cursor on the **\*EXCLUDE\*** column heading and pressing Enter. Similarly with **\*FILTER\*** from the panel display before.

At this point, as we are on the subject of 'hot' buttons. Let us take a little diversion and see what happens when we place the cursor on the Volume column heading, and press Enter. Figure 2-13 on page 15 shows the results.

File	Edit	Find	Display	Populate	Settings	Menu	Util	Test	Help	Exit
-IPT- OLIST (B) ----- LEVEL IMS* -----										Row 1 from 2596
Command ==>										SCROLL ==> CSR
Hotbar: FLIP										SET UPDATE
										*TEMPORARY LIST*
TSO PARMS ==>										
Command	Member	Numbr	Data	Set Names / Objects	*EXCLUDE*	Class				
		1	'IMS'			ALIAS				
		2	'IMS.APPC.ERRORLOG'			*MIGR*				
		3	'IMS.APPC.JOBLOG'			*MIGR*				
		4	'IMS.APPC.SYSUDUMP'			*MIGR*				
		5	'IMS.CELDCUST.D01'			*MIGR*				
		6	'IMS.CELDCUST.D01.DATA'			*MIGR*				
		7	'IMS.CELDCUSX.D01'			*MIGR*				
		8	'IMS.CELDCUSX.D01.DATA'			*MIGR*				
		9	'IMS.CELDCUSX.D01.INDEX'			*MIGR*				
		10	'IMS.CELDIAL.COPYLIB'							
		11	'IMS.CELDIAL.SOURCE'							
		12	'IMS.CELDPROD.D01'			*MIGR*				
		13	'IMS.CELDPROD.D01.DATA'			*MIGR*				
		14	'IMS.CELDPROX.D01'			*MIGR*				
		15	'IMS.CELDPROX.D01.DATA'			*MIGR*				
		16	'IMS.CELDPROX.D01.INDEX'			*MIGR*				
		20	'IMS.V9R1.ADFSLOAD'							
		21	'IMS.V9R1.ADFSMAC'							
		22	'IMS.V9R1.HFS'			HFS				
		23	'IMS.V9R1.INSTALIB'							
		24	'IMS.V9R1.INSTALL'							
		25	'IMS.V9R1.INSTATBL'							
		26	'IMS.V9R1.MATRIX'							

Figure 2-13 Toggling from Volume to Class column display

The result is that the Volume column is toggled to display the data set 'Class'. PF6 achieves the same result, as does the commands: CLASS or SHOWTYPE. To change back, toggle back by placing the cursor on the Class column and pressing Enter or issue the commands: VOLUME or SHOWVOL.

## 2.1.2 Further exploration of system datasets

Let us look at a different way to explore the system allocations. We can use the **OL/** command, as shown in Figure 2-14.

Menu	Utilities	Compilers	Options	Status	Help
z/OS Primary Option Menu					
Option ==> OL/					
0	Settings	Terminal and user parameters		User ID . . :	SE16661
1	View	Display source data or listings		Time. . . :	09:15
2	Edit	Create or change source data		Terminal. . :	3278
3	Utilities	Perform utility functions		Screen. . . :	1

Figure 2-14 Using the OL/ command to display the system files

The **OL/** command displays a pop-up panel, as shown in Figure 2-15 on page 16, where you can choose several alternative ways to populate the generated OLIST. At the top we see 1 – Allocations, which is basically what we did before when we looked into OLDD.

Menu Utilities Compilers Options Status Help	
Populate into OLIST	
Option ==>	-IPT-
0	Select one of the following options:
0	1 - Allocations...
1	2 - Catalog...
2	3 - VTOC ...
3	4 - Multiple Levels...
4	5 - History
5	6 - Migrated files...
6	7 - SYSTEM files...
7	8 - GDG (Generation Data-Groups)...
P	9 - GDS (Generation Data-Sets)...
1	10 - TAPE files...
1	11 - VSAM clusters...
1	12 - PAGE files...
1	13 - Paste (from clipboard)
1	Press Enter to process or END to cancel

User ID . : SE16661  
 Time . . : 09:15  
 Terminal . : 3278  
 Screen . . : 1  
 Language . : ENGLISH  
 Appl ID . : ISP  
 TSO logon : SYSUSER  
 TSO prefix: SE16661  
 System ID : ZT01  
 MVS acct. : SYS0000  
 Release . : ISPF 5.8

Figure 2-15 Pop-up selection panel to select Object List population criteria

Let us investigate option 7 – SYSTEM files. Either enter a 7 in the Option field or use the cursor to point-and-shoot at the line ‘system’ libraries by category, and Figure 2-16 is displayed. You might also want to refer to Appendix D, “Customizing the IBM Personal Communications” on page 305.

-IPT-

COMMAND ==>

Enter the list type or select it with the cursor.

List type ==> 5

1	- LINKLIST libraries
2	- LPALIB libraries
3	- APF authorized libraries
4	- PARMLIB libraries
5	- All of the above

Press Enter to process or the END key to cancel.

Figure 2-16 Select the type of list to display

In Figure 2-16, option **5 - All of the above** was already selected. Figure 2-17 on page 17 shows the results.

File	Edit	Find	Display	Populate	Settings	Menu	Util	Test	Help	Exit
-----										
-IPT- OLIST (B)			OBJECTS LIST					"A" will display assist		
Command ==>								SCROLL ==> CSR		
Hotbar: FLIP			CLRVOL	FILLVOL	REFRESH	UTIL	CUT	SET	UPDATE	
								*TEMPORARY LIST*		
-----										
TSO PARMS ==>										
Command	Member	Numb	Data Set Names / Objects					Volume		
-----										
-LPALIB	*****	1	!----- Link Pack Area							
		2	'SYS1.LPALIB'					Z00RES		
		3	'LPALST.IPT.V5R9.SIQILPA'					Z00CAT		
		4	'ISP.SISPLPA'					Z00RES		
		5	'ISF.SISFLPA'					Z00RES		
		6	'CEE.SCEELPA'					Z00RES		
		7	'RMF.SERBLPA'					Z00S4		
		8	'TCP/IP.SEZALPA'					Z00RES		
		9	'DFSORT.SICELPA'					Z00S4		
		10	'DFSORT.SORTLPA'					Z00S4		
		11	'BOOKMAN.SEOYLPALPA'					Z00S1		
		12	'DCE.SEUVLPA'					Z00S3		
		13	'CENTER.LPALIB'					Z00CAT		
		14	'LPALST.CICSTS.V3R2.CICS.SDFHLPALPA'					Z00CAT		
		15	'LPALST.CICSTS.V3R2.CPSM.SEYULPA'					Z00CAT		
		16	'LPALST.CICSVT.V1R2.SVIDLPALPA'					Z00CAT		
		17	'LPALST.CICSVR.V4R1.SDWLPA'					Z00CAT		
		18	'SYS1.SDWLPA'					Z00RES		
		19	'LPALST.WMQ.V6R0.SCSQLINK'					Z00CAT		
		20	'LPALST.IMS.V9R1.LPALIB'					Z00CAT		
		21	'LPALST.IMSDPROP.V3R1.LPALIB'					Z00CAT		
		22	'LPALST.REXX.V1R4.SEAGLPALPA'					Z00CAT		
		23	'LPALST.FAULTANL.V7R1.SIDIALPA'					Z00CAT		

Figure 2-17 All of LINKLIST, LPALIB, APF and PARMLIB libraries selected

The list is long. Each category is separated by a comment line that describes the libraries that follow, which we can show using the FILTER command as shown in Figure 2-18.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (B) ----- OBJECTS LIST ----- Row 1 to 23 of 412
Command ==> FILT ! SCROLL ==> CSR
Hotbar: FLIP CLRVOL FILLVOL REFRESH UTIL CUT SET UPDATE
*TEMPORARY LIST*

TSO PARMS ==>
Command Member Numbr Data Set Names / Objects Volume
-----
-LPALIB ***** 1 !----- Link Pack Area
2 'SYS1.LPALIB' Z00RES
3 'LPALST.IPT.V5R9.SIQILPA' Z00CAT
4 'ISP.SISPLPA' Z00RES

```

Figure 2-18 Using the FILTER command to show only the category separator comment lines - Part-1

Figure 2-19 on page 18 shows only the lines that have exclamation marks (!) separating the comments.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (B) ----- OBJECTS LIST ----- 0407 LINES FILTERED OUT
Command ==> SCROLL ==> CSR
Hotbar: FLIP CLRVOL FILLVOL REFRESH UTIL CUT SET UPDATE
*TEMPORARY LIST*

TSO PARMS ==>
Command Member Numbr Data Set Names / Objects *FILTER* Volume
-----
-LPALIB ***** 1 !----- Link Pack Area
-LINKLIST *ACTIVE* 26 !NAME=LNKLST5 Link List (current)
-APFLIST ***** 93 !----- APF List (dynamic)
-PARMLIB ***** 408 !----- PARMLIB List
-***** ***** 412 !----- End of @LISTSYS output
----- END OF LIST -----

```

Figure 2-19 Using the FILTER command to show only the category separator comment lines - Part-2.

To show just the libraries, we can use the FLIP command as shown in Figure 2-20.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (B) ----- OBJECTS LIST ----- 0407 LINES FILTERED OUT
Command ==> FLIP SCROLL ==> CSR
Hotbar: FLIP CLRVOL FILLVOL REFRESH UTIL CUT SET UPDATE
*TEMPORARY LIST*

TSO PARMS ==>
Command Member Numbr Data Set Names / Objects *FILTER* Volume
-----
-LPALIB ***** 1 !----- Link Pack Area
-LINKLIST *ACTIVE* 26 !NAME=LNKLST5 Link List (current)
-APFLIST ***** 93 !----- APF List (dynamic)
-PARMLIB ***** 408 !----- PARMLIB List
-***** ***** 412 !----- End of @LISTSYS output
----- END OF LIST -----

```

Figure 2-20 Using the FLIP command to reverse the display

Figure 2-21 on page 19 shows the result from entering the FLIP command.



```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (B) ----- OBJECTS LIST ----- Row 2 from 412
Command ===>          SCROLL ===> CSR
Hotbar: FLIP          CLRVOL FILLVOL REFRESH UTIL CUT SET UPDATE
                        *TEMPORARY LIST*

TSO PARMS ===>
Command Member Numbr Data Set Names / Objects *EXCLUDE* Volume
-----
2 'SYS1.LPALIB' 200RES
3 'LPALST.IPT.V5R9.SIQILPA' 200CAT
4 'ISP.SISPLPA' 200RES
5 'ISF.SISFLPA' 200RES
6 'CEE.SCEELPA' 200RES
7 'RMF.SERBLPA' 200S4
8 'TCPIP.SEZALPA' 200RES
9 'DFSORT.SICELPA' 200S4
10 'DFSORT.SORTLPA' 200S4
11 'BOOKMAN.SEOYLPA' 200S1
12 'DCE.SEUVLPA' 200S3
13 'CENTER.LPALIB' 200CAT
14 'LPALST.CICSTS.V3R2.CICS.SDFHLP' 200CAT
15 'LPALST.CICSTS.V3R2.CPSM.SEYULPA' 200CAT
16 'LPALST.CICSVT.V1R2.SVIDLPA' 200CAT
17 'LPALST.CICSVR.V4R1.SDWWLPA' 200CAT
18 'SYS1.SDWWDLPA' 200RES
19 'LPALST.WMQ.V6R0.SCSQLINK' 200CAT
20 'LPALST.IMS.V9R1.LPALIB' 200CAT

IQUIP1012 Visible rows excluded. Invisible rows revealed.

24 'LPALST.FAULTANL.V7R1.SIDILPA1' 200CAT

```

Figure 2-21 The FLIPped Object List shows all but the category separator comment lines

Now we can analyze what is on each volume, perhaps to see which libraries are on the Z00RES pack, which we can do using the SORT command, as shown in Figure 2-22.

**Note:** Remember that if at any time within an OLIST (or an MSL) you are unsure of the syntax of a command or of what commands are available, you can use the ASSIST command.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (B) ----- OBJECTS LIST ----- "A" will display assist
Command ==> A SORT SCROLL ==> CSR
Hotbar: FLIP CLRVOL FILLVOL REFRESH UTIL CUT SET UPDATE
                        *TEMPORARY LIST*

TSO PARMS ===>
Command Member Numbr Data Set Names / Objects *EXCLUDE* Volume
-----
2 'SYS1.LPALIB' 200RES

```

Figure 2-22 Checking on the command syntax using the ASSIST command

Figure 2-23 shows another SORT command.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (B) ----- OBJECTS LIST ----- Row 2 from 412
Command ==> SORT VOL D DSN A SCROLL ==> CSR
Hotbar: FLIP CLRVOL FILLVOL REFRESH UTIL CUT SET UPDATE
*TEMPORARY LIST*

TSO PARMS ==>
Command Member Numbr Data Set Names / Objects *EXCLUDE* Volume
-----
2 'SYS1.LPALIB' Z00RES
3 'LPALST.IPT.V5R9.SIQILPA' Z00CAT
4 'ISP.SISPLPA' Z00RES

```

Figure 2-23 Issuing a SORT command to order the Object List by volume

Figure 2-24 is the result of the SORT command from Figure 2-23.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (B) ----- OBJECTS LIST ----- Row 2 from 412
Command ==> SCROLL ==> CSR
Hotbar: FLIP CLRVOL FILLVOL REFRESH UTIL CUT SET UPDATE
*TEMPORARY LIST*

TSO PARMS ==>
Command Member Numbr Data Set Names / Objects *EXCLUDE* Volume
-----
2 'CEE.SCEELPA' Z00RES
3 'ISF.SISFLPA' Z00RES
4 'ISP.SISPLPA' Z00RES
5 'SYS1.LPALIB' Z00RES
6 'SYS1.SDWWDLPA' Z00RES
7 'TCPIP.SEZALPA' Z00RES
8 'DFSORT.SICELPA' Z000S4
9 'DFSORT.SORTLPA' Z000S4
10 'RMF.SERBLPA' Z000S4
11 'DCE.SEUVLPA' Z000S3
12 'BOOKMAN.SEOYLPA' Z000S1
13 'CENTER.LPALIB' Z00CAT
14 'LPALST.CICSTS.V3R2.CICS.SDFHLP' Z00CAT
15 'LPALST.CICSTS.V3R2.CPSM.SEYULPA' Z00CAT
16 'LPALST.CICSVR.V4R1.SDWWLPA' Z00CAT
17 'LPALST.CICSVT.V1R2.SVIDLPA' Z00CAT
18 'LPALST.DEBUG.V7R1.SEQALPA' Z00CAT
19 'LPALST.FAULTANL.V7R1.SIDIALPA' Z00CAT
20 'LPALST.FAULTANL.V7R1.SIDILPA1' Z00CAT
21 'LPALST.FAULTANL.V7R1.SIDILPA1' Z00CAT
22 'LPALST.FAULTANL.V7R1.SIDILPA1' Z00CAT
23 'LPALST.FAULTANL.V7R1.SIDILPA1' Z00CAT
24 'LPALST.REXX.V1R4.SEAGLPA' Z00CAT

IQIP880 Note: list contains multiple sorted groups.

```

Figure 2-24 Object list of system libraries sorted by Volume

In Figure 2-24, we can see that the top six libraries are located on pack Z00RES.

ISPF-PT is versatile though. You can obtain all of the datasets on the same pack by listing by VTOC instead of using the catalogue as shown in Figure 2-25.

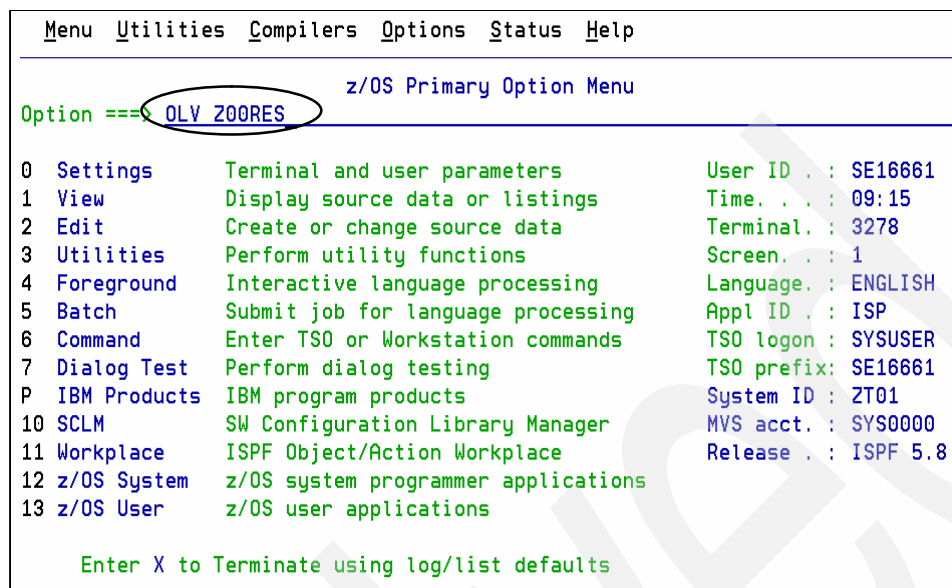


Figure 2-25 Displaying an OLIST of the VTOC of a disk volume

Figure 2-26 lists the VTOC for the pack Z00RES.

**Note:** Almost subliminally, in this case, is displayed the status panel. If you had a slow response or the volume-serial was generic (say we requested a volume-serial pattern of Z00\*), this might have been a less fleeting display.

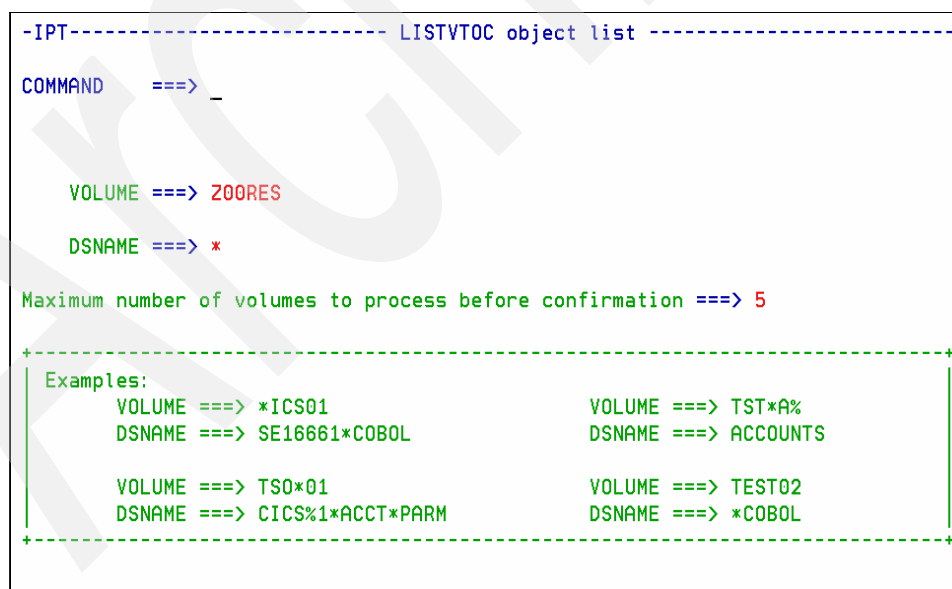


Figure 2-26 Status panel displayed while the LISTVTOC Object List is built

Figure 2-27 lists the VTOC display.

File Edit Find Display Populate Settings Menu Util Test Help Exit			
-----			
-IPT- OLIST (B) ----- SELECTED DATASETS ON 200R "A" will display assist			
Command ==> -----			
Hotbar: FLIP ----- CLRVL FILLVOL REFRESH UTIL CUT SET UPDATE			
TSO PARMS ==> -----			
Command Member Numbr Data Set Names / Objects Volume			
-----			
		1 'CEE.SAFHFORT'	200RES
		2 'CEE.SCEEBIND'	200RES
		3 'CEE.SCEEEND2'	200RES
		4 'CEE.SCEEICIS'	200RES
		5 'CEE.SCEECLST'	200RES
		6 'CEE.SCEECPMAP'	200RES
		7 'CEE.SCEECP'	200RES
		8 'CEE.SCEEGLT'	200RES
		9 'CEE.SCEEH.ARPA.H'	200RES
		10 'CEE.SCEEH.H'	200RES
		11 'CEE.SCEEH.NET.H'	200RES
		12 'CEE.SCEEH.NETINET.H'	200RES
		13 'CEE.SCEEH.SYS.H'	200RES
		14 'CEE.SCEEH.T'	200RES
		15 'CEE.SCEEH'	200RES
		16 'CEE.SCEEH.LIB'	200RES
		17 'CEE.SCEEKED'	200RES
		18 'CEE.SCEEKEX'	200RES
		19 'CEE.SCEELOCL'	200RES
		20 'CEE.SCEELOCLX'	200RES
		21 'CEE.SCEELOPA'	200RES
		22 'CEE.SCEELOMAC'	200RES
		23 'CEE.SCEELOSGP'	200RES

Figure 2-27 Displaying the VTOC of a disk volume

If you had a slow response or the volume-serial was generic, this might have been less than a fleeting display.

Remember that a complete list of available commands is available at any time through the **IPT?** command, as shown in Figure 2-28.

-IPT-		COMMAND SHORTCUTS	Row 1 to 25 of 47
COMMAND ==>			SCROLL ==> PAGE
Commands: SORT			
SHORTCUT	COMMAND	DESCRIPTION	
-----			
IPT	IPTHelp	IPT HELP	
ISet		IPT SET OPTIONS	
IMaint		IPT MAINTENANCE REPORT	
IVer	IVersion	IPT VERSION REPORT	
BR	BROWSE	'<DSNpat>(<MEMpat>)' - BROWSE	
ED	EDIT	'<DSNpat>(<MEMpat>)' - EDIT	
VI	VIEW	'<DSNpat>(<MEMpat>)' - VIEW	
BF	BFILE	'<DSNpat>(<MEMpat>)' - BROWSE VSAM	
EF	EFILE	'<DSNpat>(<MEMpat>)' - EDIT VSAM	
VF	VFILE	'<DSNpat>(<MEMpat>)' - VIEW VSAM	
O*		SHOW ALL OBJECT-LISTS	
O/		POPULATE NEW OBJECT-LIST	
OH	OHIST	HISTORY LIST OF ACCESSED DATASETS	
OLIS	OLIST	<parm1>,<parm2> - OBJECT-LIST	
OL		<parm1>,<parm2> - OBJECT-LIST	
OL*		SHOW ALL OBJECT-LISTS	
OL/		POPULATE NEW OBJECT-LIST	
OLDD	OLDDNAME	<DDname> - LIST ALLOCATED	
OLB	OLBASE	<DSNpat> - LISTCAT BASE OF GENERATION	
OLC	OLCAT	<DSNpat> - LISTCAT MULTIPLE PATTERNS	
OLE		<DSNpat> - LISTCAT PDSE LIBRARIES	
OLPDSE		<DSNpat> - LISTCAT PDSE LIBRARIES	
OLG	OLGDG	<DSNpat> - LISTCAT GDG DATASETS	
OLH	OLHIST	HISTORY LIST OF ACCESSED DATASETS	
OLI	OLINFO	<DSNpat>,<VOLpat> - LISTCAT + INFO	

Figure 2-28 The available ISPF-PT commands with the IPT? command

Scroll down to see the remaining commands.

Validating APF libraries is something that is an especially useful attribute of ISPF-PT. The APF libraries have a problem in development shops because they are often out-of-date as changes are made and the APF list is not maintained quite so energetically with deletions as it might be with additions. Going back to the O/ list, select option **7**, Figure 2-15 on page 16, for System Libraries, and then from this panel, select option **5** to generate an OLIST for the APF libraries, as shown in Figure 2-29.

Figure 2-29 OLIST for APF libraries

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (B) ----- OBJECTS LIST ----- "A" will display assist
Command ==> SCROLL ==> CSR
Hotbar: FLIP - CLRVOL FILLVOL REFRESH UTIL CUT SET UPDATE
*TEMPORARY LIST

TSO PARMS ==>
Command Member Number Data Set Names / Objects Volume
-----
-APFLIST ***** 1 |----- APF List (dynamic)
2 'SYS1.LINKLIB' 200RES
3 'SYS1.SVCLIB' 200RES
4 'SYS1.CMDLIB' 200RES
5 'SYS1.MIGLIB' 200RES
6 'SYS1.VTAMLIB' 200RES
7 'SYS1.DFQLLIB' 200RES
8 'SYS1.DGTLLIB' 200RES
9 'SYS1.CSSLIB' 200RES
10 'SYS1.SAPPMOD1' 200RES
11 'SYS1.SHASLNKE' 200RES
12 'SYS1.SHASMIG' 200RES
13 'SYS1.SIEAMIGE' 200RES
14 'SYS1.NFSLIBE' 200RES
15 'SYS1.SCUNIMG' 200RES
16 'SYS1.SIEALNKE' 200RES
17 'CEE.SCEERUN' 200RES
18 'CEE.SCEERUN2' 200RES
19 'SYS1.SCBDHENU' 200RES
20 'SYS1.SISTCLIB' 200RES
21 'GIM.SGIMLMD0' 200RES
22 'ISF.SISFLPA' 200RES
23 'ISF.SISFLOAD' 200RES

```

If you place the cursor on the **Volume** column heading and press enter, the column toggles to display the **Class**, as shown in Figure 2-31 on page 24.

File Edit Find Display Populate Settings Menu Util Test Help Exit									
-----									
-IPT- OLIST (B)				OBJECTS LIST				Row 1 to 23 of 312	
Command ==>								SCROLL ==> CSR	
Hotbar: FLIP				CLRVOL	FILLVOL	REFRESH	UTIL	CUT	SET UPDATE
								*TEMPORARY LIST*	
-----									
TSO PARMS ==>									
Command	Member	Numbr	Data Set Names / Objects						Class
-----									
-APFLIST	*****	1	APF List (dynamic)						
		2	'SYS1.LINKLIB'						
		3	'SYS1.SVCLIB'						
		4	'SYS1.CMDLIB'						
		5	'SYS1.MIGLIB'						
		6	'SYS1.VTAMLIB'						
		7	'SYS1.DFQLLIB'						
		8	'SYS1.DGTLLIB'						
		9	'SYS1.CSSLIB'						
		10	'SYS1.SAPPMOD1'						
		11	'SYS1.SHASLNKE'						
		12	'SYS1.SHASMIG'						
		13	'SYS1.SIEAMIGE'						
		14	'SYS1.NFSLIBE'						
		15	'SYS1.SCUNIMG'						
		16	'SYS1.SIEALNKE'						
		17	'CEE.SCEERUN'						
		18	'CEE.SCEERUN2'						
		19	'SYS1.SCBDHENU'						
		20	'SYS1.SISTCLIB'						
		21	'GIM.SGIMLMD0'						
		22	'ISF.SISFLPA'						
		23	'ISF.SISFLOAD'						

Figure 2-31 Class display

Notice that the library Class fields are empty in Figure 2-31. To check the validity of the entries, we need to issue a VALIDATE command.

Use the ASSIST command to see exactly what the VALIDATE command does, as shown in Figure 2-32.

-IPT-	OLIST Command Assistance
Cmd ==>	-
	Validate existence of OLIST dataset objects
Type:	Main command.
Syntax:	VALIDate
Examples:	VALID - for dataset objects with a blank VOLSER, check the catalog and if cataloged, their corresponding VTOCs. for dataset objects with a valid VOLSER, check their corresponding VTOCs.
Notes:	OLIST entries of datasets that do not exist, are marked with a feedback message -NOT FND- in the Command field. OLIST RESET DEL command can be used to remove all entries marked -NOT FND- .

Figure 2-32 VALIDATE command

Figure 2-33 shows the result of the VALIDATE command.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (B) ----- OBJECTS LIST ----- "A" will display assist
Command ==> validate SCROLL ==> CSR
Hotbar: FLIP CLRVOL FILLVOL REFRESH UTIL CUT SET UPDATE
*TEMPORARY LIST*

TSO PARMS ==>
Command Member Numbr Data Set Names / Objects Class
-----
-APFLIST ***** 1 |----- APF List (dynamic)
                  2 'SYS1.LINKLIB'
                  3 'SYS1.SVCLIB'
                  4 'SYS1.CMDLIB'

```

Figure 2-33 Result of VALIDATE command

A status message is displayed on the panel as it progresses, and there can be a large number of APF libraries depending on the installation, as shown in Figure 2-34.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (B) ----- OBJECTS LIST -----
Command ==> VALIDATE SCROLL ==> CSR
Hotbar: FLIP CLRVOL FILLVOL REFRESH UTIL CUT SET UPDATE
*TEMPORARY LIST*

TSO PARMS ==>
Command Member Numbr Data Set Names / Objects Class
-----
-LPALIB ***** 1 |----- Link Pack Area
                  2 'SYS1.LPALIB'
                  3 'LPALST.IPT.V5R9.SIQLPA'
                  4 'ISP.SISPLPA'
                  5 'ISF.SISFLPA'
                  6 'CEE.SCEELPA'
                  7 'RMF.SERBLPA'
                  8 'TCP/IP.SEZALPA'
                  9 'DFSORT.SICELPA'
                  10 'DFSORT.SORTLPA'
                  11 'BOOKMAN.SEOYLPA'
                  12 'DCE.SEUVLPA'
                  13 'CENTER.LPALIB'
                  14 'LPALST.CICSTS.V3R2.CICS.SDFHLPA'
                  15 'LPALST.CICSTS.V3R2.CPSM.SEYULPA'
                  16 'LPALST.CICSVT.V1R2.SVIDLPA'
                  17 'LPALST.CICSVR.V4R1.SDWLPA'
                  18 'SYS1.SDWDLPA'
                  19 'LPALST.WMQ.V6R0.SCSQLINK'

IQIP904 Validating data sets (16 were processed)

23 'LPALST.FAULTANL.V7R1.SIDIALPA'

```

Figure 2-34 APF libraries listing

After validation, another completion status message is displayed that shows the number of APF libraries that are no longer valid, as shown in Figure 2-35 on page 26.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (B) ----- OBJECTS LIST ----- Row 1 to 18 of 407
Command ===> _ Scroll ===> CSR

Hotbar: FLIP CLRVOL FILLVOL REFRESH UTIL CUT SET UPDATE
+-----+
| NOTE: Enter the "RESET DEL" command to remove items marked as deleted |
| (Those with "-NOT FND-" in the COMMAND column) |
+-----+

*TEMPORARY LIST*

TSO PARMS ===>
Command Member NUM Data Set Names / Objects Class
-----
-LPALIB ***** 1 |----- Link Pack Area
2 'SYS1.LPALIB' PDS
3 'LPALST.IPT.V5R9.SIQLPA' PDS
4 'ISP.SISPLPA' PDS
5 'ISF.SISFLPA' PDS
6 'CEE.SCEELPA' PDS
7 'RMF.SERBLPA' PDS
8 'TCPIP.SEZALPA' PDS
9 'DFSORT.SICELPA' PDS
10 'DFSORT.SORTLPA' PDS
11 'BOOKMAN.SEOYLPA' PDS
12 'DCE.SEUVLPA' PDS
13 'CENTER.LPALIB' PDS
14 'LPALST.CICSTS.V3R2.CICS.SDFHLP' PDS
15 'IQIP908 19 item(s) not found' PDS
16 'SYS1.SDWWDLPA' PDS
17
18 'SYS1.SDWWDLPA' PDS

```

Figure 2-35 Invalid APF libraries

Scroll down more to find one of these **-NOT FND-** libraries, as shown in Figure 2-36.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (B) ----- OBJECTS LIST ----- Row 148 to 165 of 407
Command ===> _ Scroll ===> CSR

Hotbar: FLIP CLRVOL FILLVOL REFRESH UTIL CUT SET UPDATE
+-----+
| NOTE: Enter the "RESET DEL" command to remove items marked as deleted |
| (Those with "-NOT FND-" in the COMMAND column) |
+-----+

*TEMPORARY LIST*

TSO PARMS ===>
Command Member NUM Data Set Names / Objects Class
-----
- NOT FND- 148 'LDAP.SGLDLNK' PDS
149 'SSL.SGSKLOAD' PDS
150 'FIREWALL.SICALMOD' *DEL*
151 'NETRSLTS.NTR35.MOD' PDS
152 'DFSORT.SICELINK' PDS
153 'DFSORT.SICELPA' PDS
154 'CBC.SCBCCMP' PDS
155 'CBC.SCLBDLL' PDS
156 'GDDM.SADMMOD' PDS
157 'BOOKMAN.SEOYLPA' PDS
158 'BOOKMAN.SEOYLOAD' PDS
159 'FFST.SEPWMOD1' PDS
160 'FFST.SEPWMOD2' PDS
161 'TIVSM.V5R3.LINKLIB' PDS
162 'TIVOM.V2R1.SBJTLOAD' PDS
163 'TIVAO.V1R1.SGLOLOAD' PDS
164 'DCF.DCFLOAD' PDS
- NOT FND- 165 'PASCAL.V1R2M0.SAMPRUN1' *DEL*

```

Figure 2-36 Scroll to NOT FIND

An on-panel message, just under the Hotbar, says that to remove the **-NOT FND-** libraries from the list, issue the RESET DEL command, as shown in Figure 2-37 on page 27.



```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (B) ----- OBJECTS LIST ----- Row 148 to 165 of 407
Command ==> reset del Scroll ==> CSR

Hotbar: FLIP CLRVOL FILLVOL REFRESH UTIL CUT SET UPDATE
+----- Assist -+
| NOTE: Enter the "RESET DEL" command to remove items marked as deleted |
| (Those with "-NOT FND-" in the COMMAND column) |
+-----+
*TEMPORARY LIST*

TSO PARMS ==>
Command Member NUM Data Set Names / Objects Class
-----
148 'LDAP.SGLDLNK' PDS
149 'SSL.SGSKLOAD' PDS
- NOT FND- 150 'FIREWALL.SICALMOD' *DEL*
151 'NETRSLTS.NTR35.MOD' PDS

```

Figure 2-37 RESET DEL command

Figure 2-38 shows the results of the RESET DEL command.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (B) ----- OBJECTS LIST ----- Row 148 to 170 of 388
Command ==> SCROLL ==> CSR
Hotbar: FLIP CLRVOL FILLVOL REFRESH UTIL CUT SET UPDATE
*TEMPORARY LIST*

TSO PARMS ==>
Command Member Numbr Data Set Names / Objects Class
-----
148 'LDAP.SGLDLNK' PDS
149 'SSL.SGSKLOAD' PDS
150 'NETRSLTS.NTR35.MOD' PDS
151 'DFSORT.SICELINK' PDS
152 'DFSORT.SICELPA' PDS
153 'CBC.SCBCCMP' PDS
154 'CBC.SCLBDLL' PDS
155 'GDDM.SADMMOD' PDS
156 'BOOKMAN.SEOYLPA' PDS
157 'BOOKMAN.SEOYLOAD' PDS
158 'FFST.SEPWMOD1' PDS
159 'FFST.SEPWMOD2' PDS
160 'TIVSM.V5R3.LINKLIB' PDS
161 'TIVOM.V2R1.SBJTLOAD' PDS
162 'TIVAO.V1R1.SGLOLOAD' PDS
163 'DCF.DCFLOAD' PDS
164 'VACOBOL.V3R0M7.SFBNLOAD' PDS
165 'WSED.V5R1M2.SCCULOAD' PDS
166 'CICSTS.V3R1.CICS.SDFHAUTH' PDS
167 'CICSTS.V3R1.CICS.SDFJAUTH' PDSE
168 'CICSTS.V3R1.CICS.SDFHEXCI' PDS
169 'CICSTS.V3R1.CPSM.SEYUAUTH' PDS
170 'CICSTS.V3R1.CICS.SDFHLOAD' PDS

```

Figure 2-38 Result of removing RESET DEL command

**Note:** You can also use the VALIDATE command for older OLISTs, for example, permanent OLISTs that might be out of date.

## 2.2 Group execution

Let us start again by looking at IMS\* OLIST. We first exclude the catalog entry, number 1 in the list, which we can do with an **/X** line command, as shown in Figure 2-39.

```
File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (B) ----- LEVEL IMS* ----- Row 1 to 23 of 2,613
Command   ==>
Hotbar: FLIP  CLRVOL  FILLVOL  REFRESH  UTIL  CUT  SET  UPDATE
                                         *TEMPORARY LIST*

TSO PARMS ==>
Command  Member  Numbr Data Set Names / Objects  Volume
-----
/X      1 'IMS'
        2 'IMS.APPC.ERRORLOG'          Z00D16
        3 'IMS.APPC.JOBLOG'            Z00D17
        4 'IMS.APPC.SYSUDUMP'          Z00D08
```

Figure 2-39 Excluding a catalogue entry from an Object List – by line command

The first line of the list is excluded.

### 2.2.1 Group execution by number reference

To do the same **/X** line command for a single line by number, enter **1 x**, as shown in Figure 2-40.

```
File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (B) ----- LEVEL IMS* ----- Row 1 to 23 of 2,553
Command   ==> 1 x
Hotbar: FLIP  CLRVOL  FILLVOL  REFRESH  UTIL  CUT  SET  UPDATE
                                         *TEMPORARY LIST*

TSO PARMS ==>
Command  Member  Numbr Data Set Names / Objects  Volume
-----
        1 'IMS'
        2 'IMS.APPC.ERRORLOG'          Z00D16
        3 'IMS.APPC.JOBLOG'            Z00D17
        4 'IMS.APPC.SYSUDUMP'          Z00D08
```

Figure 2-40 Excluding a catalogue entry from an Object List – by number reference

The first line of the list is excluded. The list now looks similar to Figure 2-41 on page 29.

File	Edit	Find	Display	Populate	Settings	Menu	Util	Test	Help	Exit
-----										
-IPT- OLIST (B) ----- LEVEL IMS* -----								Row 2 from 2553		
Command ==>								SCROLL ==> CSR		
Hotbar: FLIP    CLRVOL    FILLVOL    REFRESH    UTIL    CUT    SET    UPDATE								*TEMPORARY LIST*		
TSO PARMS ==>										
Command	Member	Numbr	Data Set Names / Objects						*EXCLUDE*	Volume
-----										
		2	'IMS.APPC.ERRORLOG'							MIGRAT
		3	'IMS.APPC.JOBLOG'							MIGRAT
		4	'IMS.APPC.SYSUDUMP'							MIGRAT
		5	'IMS.CELDCUST.D01'							MIGRAT
		6	'IMS.CELDCUST.D01.DATA'							MIGRAT
		7	'IMS.CELDCUSX.D01'							MIGRAT
		8	'IMS.CELDCUSX.D01.DATA'							MIGRAT
		9	'IMS.CELDCUSX.D01.INDEX'							MIGRAT
		10	'IMS.CELDIAL.COPYLIB'							Z00D08
		11	'IMS.CELDIAL.SOURCE'							Z00D10
		12	'IMS.CELDPROD.D01'							MIGRAT
		13	'IMS.CELDPROD.D01.DATA'							MIGRAT
		14	'IMS.CELDPROX.D01'							MIGRAT
		15	'IMS.CELDPROX.D01.DATA'							MIGRAT
		16	'IMS.CELDPROX.D01.INDEX'							MIGRAT
		17	'IMS.JAVADEMO.CNTL'							Z00D18
		18	'IMS.JAVADEMO.JAVADB1'							
		19	'IMS.JAVADEMO.JAVADB1.DATA'							Z00D02
		20	'IMS.V9R1.ADFSLOAD'							Z00D03
		21	'IMS.V9R1.ADFSMACT'							Z00D12
		22	'IMS.V9R1.HFS'							Z00D02
		23	'IMS.V9R1.INSTALIB'							Z00D01
		24	'IMS.V9R1.INSTALL'							Z00D10

Figure 2-41 Excluded list

Many of these data sets that we can see from the Volume column are migrated (MIGRAT). From the Class view, we would have seen the value \*MIGR\*.

If we needed to recall all of the 'IMS.APPC.\*' data sets, we could achieve it, as shown in Figure 2-42.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (B) ----- LEVEL IMS* ----- Row 2 from 2553
Command ==> 2-4 hreca SCROLL ==> CSR
Hotbar: FLIP CLRVOL FILLVOL REFRESH UTIL CUT SET UPDATE
                                         *TEMPORARY LIST*

TSO PARMS ==>
Command Member Numbr Data Set Names / Objects *EXCLUDE* Volume
-----
2 'IMS.APPC.ERRORLOG' MIGRAT
3 'IMS.APPC.JOBLOG' MIGRAT
4 'IMS.APPC.SYSUDUMP' MIGRAT
5 'IMS.CELDCUST.D01' MIGRAT
6 'IMS.CELDCUST.D01.DATA' MIGRAT
7 'IMS.CELDCUSX.D01' MIGRAT
8 'IMS.CELDCUSX.D01.DATA' MIGRAT
9 'IMS.CELDCUSX.D01.INDEX' MIGRAT
10 'IMS.CELDIAL.COPYLIB' Z00D08

```

Figure 2-42 Recalling a group of datasets in one IPT command, by number reference

As shown in Figure 2-42, recalled datasets with 2-4 in their **Numbr** column result in Figure 2-43 on page 30.

```

ARC1000I IMS.APPC.ERRORLOG RECALL PROCESSING ENDED
ARC1000I IMS.APPC.SYSUDUMP RECALL PROCESSING ENDED
ARC1007I RECALL REQUEST 00003175 SENT TO DFSMSHSM
ARC1007I RECALL REQUEST 00003176 SENT TO DFSMSHSM
ARC1007I RECALL REQUEST 00003177 SENT TO DFSMSHSM
***

```

Figure 2-43 Recall in progress for a group of migrated datasets.

To view a subset of datasets, enter the command, as shown in Figure 2-44.

File Edit Find Display Populate Settings Menu Util Test Help Exit			
-----			
-IPT- OLIST (B)	-----	LEVEL IMS*	-----
Command ==>	9-10 v		Row 2 from 2553
Hotbar: FLIP	CLRVL	FILLVOL	REFRESH UTIL CUT
			SET UPDATE
			*TEMPORARY LIST*
TSO PARMS ==>			
Command	Member	Numbr Data Set Names / Objects	*EXCLUDE* Volume
-----			
		2 'IMS.APPC.ERRORLOG'	Z00D05
		3 'IMS.APPC.JOBLOG'	Z00D02
		4 'IMS.APPC.SYSUDUMP'	Z00D07
		5 'IMS.CELDCUST.D01'	MIGRAT
		6 'IMS.CELDCUST.D01.DATA'	MIGRAT
		7 'IMS.CELDCUSX.D01'	MIGRAT
		8 'IMS.CELDCUSX.D01.DATA'	MIGRAT
		9 'IMS.CELDCUSX.D01.INDEX'	MIGRAT
		10 'IMS.CELDIAL.COPYLIB'	Z00D08
		11 'IMS.CELDIAL.SOURCE'	Z00D10

Figure 2-44 Requesting to View a group of data sets

In Figure 2-44, we inadvertently selected 9-10 instead of 10-11, and the first of these is a migrated data set (quite apart from the fact that it is the Index of a VSAM data set!). ISPF-PT knows that the data set is migrated and that we cannot wish for the standard TSO response, to WAIT or hit attention; therefore, ISPF-PT provides certain alternative options to use, as shown in Figure 2-45 on page 31.

```

-IPT- PLIST ----- HSM MIGRATED DATA SET RECALL PROMPT PANEL -----
OPTION ==> X

You have requested the following HSM migrated data set:
IMS.CELDCUSX.D01.INDEX

Select one of the following options (option 2 is the default):

1 INFORMATION      - Display migrated data set information.
2 RECALL and WAIT  - Recall the data set. Wait until data set is recalled.
3 RECALL offline   - Recall the data set. Do not wait for it to be recalled.
X EXIT             - Do not recall the data set

Make your selection and press ENTER. (The END key will exit)

```

Figure 2-45 Migrated data set recall prompt

As you can see in Figure 2-45, we chose option **X** to ignore our previous action, and go on to VIEW the next data set, as shown in Figure 2-46.

```

File Display Library Settings Menu Utilities Test Help Exit
-----
-IPT--VIEW L1----- IMS.CELDIAL.COPYLIB ----- "A" will display assist
COMMAND ==> _ SCROLL ==> PAGE
HOTBAR?

                                ON VOLUME 200D08
NAME  RENAME  LIB VV.MM CREATED   CHANGED   SIZE  INIT  MOD USERID
CELDUST      1 01.00 94/06/05 94/06/05 09:51   36   36   0 HAMEL
CELDPROD     1 01.01 94/06/05 94/06/05 09:52   19   74   0 HAMEL
CUSTOMER     1 01.01 94/06/04 94/06/04 10:36   14   72   0 HAMEL
CUSTSHIP     1 01.00 94/06/04 94/06/04 10:34   12   12   0 HAMEL
ORDRDET      1 01.00 94/06/04 94/06/04 10:35    5    5   0 HAMEL
ORDRSUMM     1 01.00 94/06/04 94/06/04 10:35    7    7   0 HAMEL
PRODSHOW     1 01.00 94/06/05 94/06/05 09:52    5    5   0 HAMEL
PRODUCT      1 01.00 94/06/05 94/06/05 09:53    7    7   0 HAMEL
--END--

```

Figure 2-46 Viewing a COPYLIB library's directory

We can now go back and correctly View 10-11 from the beginning.

## 2.2.2 Group execution by generic reference

We can act on a whole range of datasets by issuing a generic reference.

\* DELETE, for instance, as shown in Figure 2-47 on page 32, might send cold shivers up your spine until you realize that ISPF-PT makes certain that you really want to delete. ISPF-PT gives you the option to cancel one or all actions.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (B) ----- LEVEL IMS* ----- "A" will display assist
Command ==> * DELETE                                SCROLL ==> CSR
Hotbar: FLIP      CLRVOL  FILLVOL  REFRESH  UTIL      CUT      SET      UPDATE
                                           *TEMPORARY LIST*

TSO PARMS ==>
Command  Member  Numbr Data Set Names / Objects  *FILTER* *EXCLUDE* Volume
-----
          2  'IMS.APPC.ERRORLOG'                                Z00D05
          3  'IMS.APPC.JOBLOG'                                  Z00D02
          4  'IMS.APPC.SYSUDUMP'                                Z00D07
          ----- END OF LIST -----

```

Figure 2-47 Generically deleting data sets

Figure 2-48 gives you the option to cancel one or all actions.

```

-IPT----- CONFIRM DELETE REQUEST -----
COMMAND ==> _

You have requested the deletion of the following data set:
DSNAME: IMS.APPC.ERRORLOG
VOLSER: Z00D05
DSORG:  PS
LRECL:  80

Press ENTER to delete the data set, or END key to cancel.

IQIP730 Enter QUIT to abort executing "DELETE" before end of range.

```

Figure 2-48 Cancelling a request to delete a data set

In Figure 2-48, we press the **END** key to cancel and return to Figure 2-49 on page 33.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (B) ----- LEVEL IMS* ----- Row 2 from 2553
Command ===> SCROLL ===> CSR
Hotbar: FLIP - CLRVOL FILLVOL REFRESH UTIL CUT SET UPDATE
*TEMPORARY LIST*

TSO PARMS ===>
Command Member Numbr Data Set Names / Objects *FILTER* *EXCLUDE* Volume
-----
-NO DEL          2 'IMS.APPC.ERRORLOG'          Z00D05
                  3 'IMS.APPC.JOBLOG'          Z00D02
                  4 'IMS.APPC.SYSUDUMP'         Z00D07
                  ----- END OF LIST -----

```

IQIP730 Enter QUIT to abort executing "DELETE" before end of range.

Figure 2-49 QUITing to terminate the deletion process

Type QUIT on the command line, and press Enter. ISPF-PT checks if we want to quit for our whole request, for example, all pending DELETES, as shown in Figure 2-50.

IQIQ002 Pending commands will not be executed. Enter "QUIT STOP" to resume.

Figure 2-50 Message to indicate that commands are pending

Type QUIT STOP on the command line, and press Enter, which returns to Figure 2-51.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (B) ----- LEVEL IMS* ----- QUIT stopped
Command ===> SCROLL ===> CSR
Hotbar: FLIP - CLRVOL FILLVOL REFRESH UTIL CUT SET UPDATE
*TEMPORARY LIST*

TSO PARMS ===>
Command Member Numbr Data Set Names / Objects *FILTER* *EXCLUDE* Volume
-----
-NO DEL          2 'IMS.APPC.ERRORLOG'          Z00D05
                  3 'IMS.APPC.JOBLOG'          Z00D02
                  4 'IMS.APPC.SYSUDUMP'         Z00D07
                  ----- END OF LIST -----

```

Figure 2-51 Issuing QUIT STOP to terminate a whole group action

The group deletion is terminated.

## 2.3 Permanent Object List

Creating a temporary view of data objects is useful, but sometimes you might want to save a certain set of objects that you can use frequently in a particular operation. It can be something as simple as the set libraries that you use the most often or the set of databases that you monitor or maintain.

### 2.3.1 Permanent Object List for projects

To help explain the “Permanent OLIST”, in this section we describe a real-life scenario (project). This project is not large, which helps to keep things understandable. This project does not involve all of the ISPF-PT features.

This scenario involves one activity that frequently occurs for a particular development-center programmer who was called on to assist some senior auditors in Asset Management in some complicated cross-matching of data. As is often the case, this becomes a periodic on-demand task for this professional.

The asset manager has several massive Excel® spreadsheets that have various details, such as assets, contacts, orders, shipments, deliveries, and so on that they update and manipulate, attempting reconciliations. Apparently Excel is not so good for manipulating very large volumes, using LOOKUPs, and using VLOOKUPs. So they call upon the assistance of the development-center programmer.

The development-center programmer can, of course, resolve the problem in several ways. DB2 tables might be a solution, if they were consistent and precise in their use of the spreadsheets, but they are not. This particular programmer is adept in REXX and has a large processor at their disposal. He merely saves and uploads the spreadsheets as CSV files and processes them using compiled REXX programs.

Although the file formats and content and requirements change each time, the activity is similar each time it is requested. The spreadsheets to be matched—two, three, sometimes up to six—are sent to the programmer who uploads them to the host. A new REXX exec is then tailored based on previous examples, compiled and tested, and any required modifications applied. The results from the match are downloaded to the PC and then e-mailed to the auditors.

The following list summarizes the tasks that are completed on the host. Receiving and sending files is actually completed, in this case, using Lotus® Notes® attachments:

- ▶ Upload CSV files to host.
- ▶ Modify run JCL to accept new input and output filenames.
- ▶ Modify the compile JCL to compile the new exec.
- ▶ Create a new REXX exec, and compile the REXX exec to a load module.
- ▶ Run the REXX program.
- ▶ Review the output of the CSV file.
- ▶ Download the output CSV to the PC.

### 2.3.2 Creating project ‘ELUX’ OLIST

Before we look at the final OLIST, which has been around for a while, let us look at how the first ELUX project OLIST was created from scratch. It does not map exactly to the order in the previous bulleted list because there are a few additional stages in the initial setup.



Set the defaults that you want for OLISTs, in general, with the ISET command, as shown in Figure 2-52.

```

Menu  Utilities  Compilers  Options  Status  Help
-----
ISR@390                               z/OS Primary Option Menu          IBM IPT VERSION 5.9
Option ==> ISET

0 Settings      Terminal and user parameters      User ID   : SE16661
1 View          Display source data or listings      Time     : 07:10
2 Edit          Create or change source data      Terminal : 3278
3 Utilities      Perform utility functions          Screen   : 1
4 Foreground     Interactive language processing      Language : ENGLISH
5 Batch          Submit job for language processing      Appl ID  : ISP
6 Command        Enter TSO or Workstation commands      TSO logon: SYSUSER
7 Dialog Test    Perform dialog testing          TSO prefix: SE16661
P IBM Products   IBM program products          System ID : ZT01
10 SCLM          SW Configuration Library Manager      MVS acct. : SYS0000
11 Workplace     ISPF Object/Action Workplace          Release  : ISPF 5.8
12 z/OS System   z/OS system programmer applications
13 z/OS User      z/OS user applications

Enter X to Terminate using log/list defaults

```

Figure 2-52 Setting OLIST defaults using ISET

Chose option **P** to select your defaults, as shown in Figure 2-53.

```

-IPT- -----Setting IBMIPT Defaults-----
COMMAND ==> P
Select options by number, name, with cursor selection, or with line commands:
  IBMIPT is running under ISPF version 5.8

- A - ALL       - Select all the below displayed options
- M - MSL       - Member Selection List options
- P - OLIST     - Object list options
- G - GLOBAL    - Global edit and Findtext options
- R - PRINT     - Print options
- D - DSLIST    - DSLIST options
- T - TSO       - TSO shell options
- E - EDIT      - Edit, Browse and View options
- I - INTERFACE - Specify user interface options
- N - DIAGNOSE  - Diagnose ISPF errors
- L - LIBRARY   - Persistent table library options

Make your selection and press the ENTER key or press the END key to exit

```

Figure 2-53 Setting OLIST defaults using ISET option P

Select option **P**, and press Enter. Figure 2-54 on page 36 is displayed.

```

-IPT- -----OLIST - Object List Options-----
COMMAND ==>

Default command (when selecting an item) ==> B (B=Browse, E=Edit, V=View)

Show VOLSER of cataloged data-sets ==> N (Y=Yes, N=No)

Check edit recovery when the list opens ==> N (Y=Yes, N=No)

Provide a field for TSO command parms ==> Y (Y=Yes, N=No)

Display mode (right column shows) ==> C (C=Class, V=Volume)

Default cursor position ==> M (M=Main command, L=Line cmd)

Press ENTER for options menu, END to exit, CANCEL for installation defaults.

```

Figure 2-54 Setting OLIST defaults – Object List Options panel

In Figure 2-54, we chose **N** for the option **Show VOLSER of cataloged data-sets** and **C** for the option **Display mode (right column shows)** because these days using uncataloged datasets is rare for most users. The actual location of the data is unimportant when just about everywhere uses system managed storage.

Choose your own preferences for the default actions. You can always come back at any time and amend them.

Return directly to the main menu by pressing END.

Create an empty Permanent OLIST as shown in Figure 2-55.

```

Menu Utilities Compilers Options Status Help
-----
z/OS Primary Option Menu          IBM IPT VERSION 5.9

Option ==> OL_ELUX

0 Settings      Terminal and user parameters      User ID . . : SE16661
1 View          Display source data or listings      Time. . . . : 06:37
2 Edit          Create or change source data      Terminal. . : 3278
3 Utilities     Perform utility functions      Screen. . . : 1
4 Foreground    Interactive language processing      Language. . : ENGLISH
5 Batch         Submit job for language processing      Appl ID . . : ISP
6 Command       Enter TSO or Workstation commands      TSO logon . : SYSUSER
7 Dialog Test   Perform dialog testing      TSO prefix: SE16661
P IBM Products  IBM program products      System ID . : ZT01
10 SCLM         SW Configuration Library Manager      MVS acct. . : SYS0000
11 Workplace    ISPF Object/Action Workplace      Release . . : ISPF 5.8
12 z/OS System  z/OS system programmer applications
13 z/OS User    z/OS user applications

Enter X to Terminate using log/list defaults

```

Figure 2-55 Creating a Permanent OLIST

This creates an empty OLIST named ELUX, as shown in Figure 2-56 on page 37.

```

-IPT----- Object List -----
COMMAND ==>

Specify the DSNNAME pattern you would like to populate into the OLIST:

DSNAME ==> ELUX

Optionally specify a VOLUME pattern:

VOLUME ==> (To filter data sets catalogued to this volume pattern)

-----
Examples:
DSNAME ==> SE16661*PAY%*%*DATA      DSNNAME ==> SE16661*COBOL
VOLUME ==>                          VOLUME ==> PROD*
-----

Press ENTER to proceed or END key to cancel the catalog search command.

IQIP972  "ELUX" not in catalog

```

Figure 2-56 ELUX Object List

Press **END** to continue. We added FTP control member(s).

We need to upload the files from the PC to the host and back. We do not have the workstation connections set up here because the programmers are all familiar with, and happy to use, FTP for this task. The filenames and password in the FTP commands might change, but the basic process is the same. After you set up the FTP command, the FTP command does not change for this particular piece of work, which we show in Figure 2-57.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (B) ----- Objects List ----- "A" will display assist
Command ==> SCROLL ==> CSR
Hotbar: FLIP CLRVOL FILLVOL REFRESH UTIL CUT SET UPDATE
Open list ==> ELUX (or BLANK for reference list)
TSO PARMS ==>
Command Member Numbr Data Set Names / Objects Class
-----
1 -
2 -
3 -
4 =
5 -
6 -
7 -
8 -
9 -
----- END OF LIST -----

```

Figure 2-57 Setting up the FTP command to start to populate the Permanent OLIST

Place the cursor under the first Data Set Names / Objects field, and type in the library in which control elements for this are kept: SE16661.ELUX.PCTL, as shown in Figure 2-58 on page 38. In the member name field, we could type the member name where this particular control element will be, FTPUPLD; however, there needs to be a similar member to download the results afterwards, which will be called FTPDNLD. So we will use a member mask of

FTP%%LD here. In the Command field, for this line, enter **E** to Edit the new members. Type **SAVE** in the Command field, and press Enter to save our new OLIST.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (B) ----- Objects List ----- "A" will display assist
Command ==>
Hotbar: FLIP CLRVOL FILLVOL REFRESH UTIL CUT SET UPDATE
Open list ==> ELUX (or BLANK for reference list)
TSO PARMS ==>
Command Member Numbr Data Set Names / Objects Class
-----
FTP%%LD 1 'SE16661.ELUX.PCTL' _
2 _
3 _
4 _
5 _
6 _
7 _
8 _
9 _
----- END OF LIST -----

```

Figure 2-58 Populating the Object List

But let us also give the OLIST a meaningful description in the title line. At the moment we see that the description (center-padded by dashes) is “Objects List”. Place the cursor in this area, and press Enter. A pop-up panel is displayed, where you can type the description of your choice, as shown in Figure 2-59.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLI ----- DEFINE OLIST DESCRIPTION ----- -IPT- of 9
Command COMMAND ==> CSR
Hotbar: F ATE
Open list OLIST name . . . . . ELUX
TSO PARMS OLIST description ==> _
Command
-----
Note: The description is preserved in the reference list.
Press ENTER to accept or the END key to cancel.
4 _
5 _
6 _
7 _
8 _
9 _
----- END OF LIST -----

```

Figure 2-59 Adding an OLIST description – Part-1

Next, we add the title as Asset Management, as shown in Figure 2-60.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLI                                     -IPT- of 9
Command COMMAND ==>                               CSR
Hotbar: F                                         ATE
Open list OLIST name . . . . . ELUX
TSO PARMS OLIST description ==> Asset Management
Command ass
-----
Note: The description is preserved in the reference list.

Press ENTER to accept or the END key to cancel.

4 _
5 _
6 _
7 _
8 _
9 _
----- END OF LIST -----

```

Figure 2-60 Adding an OLIST description – Part-2

Press Enter to accept the name, and Figure 2-61 is displayed. If at anytime you want to change the name, repeat this procedure.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (B) ----- Asset Management ----- Row 1 to 9 of 9
Command ==> save SCROLL ==> CSR
Hotbar: FLIP CLRVOL FILLVOL REFRESH UTIL CUT SET UPDATE
Open list ==> ELUX (or BLANK for reference list)
TSO PARMS ==>
Command Member Numbr Data Set Names / Objects Class
-----
FTP%LD 1 'SE16661.ELUX.PCTL'
2 _
3 _
4 _
5 _
6 _
7 _
8 _
9 _
----- END OF LIST -----

```

Figure 2-61 Adding an OLIST description – Part-3

Save it to keep what we did so far. Press Enter, and the OLIST is saved with this single entry, as shown in Figure 2-62 on page 40.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (B) ----- Asset Management ----- ELUX saved
Command ==> SCROLL ==> CSR
Hotbar: FLIP CLRVOL FILLVOL REFRESH UTIL CUT SET UPDATE
Open list ==> ELUX (or BLANK for reference list)
TSO PARMS ==>
Command Member Numbr Data Set Names / Objects Class
-----
FTP%LD 1 'SE16661.ELUX.PCTL'
----- END OF LIST -----

```

Figure 2-62 Adding an OLIST description – Part-4

To save time, we set up the two members with the FTP up and download commands already.

Type E in the command field for line 1, and press Enter to edit the members that match this mask, as shown in Figure 2-63.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (B) ----- Asset Management ----- ELUX saved
Command ==> SCROLL ==> CSR
Hotbar: FLIP CLRVOL FILLVOL REFRESH UTIL CUT SET UPDATE
Open list ==> ELUX (or BLANK for reference list)
TSO PARMS ==>
Command Member Numbr Data Set Names / Objects Class
-----
E FTP%LD 1 'SE16661.ELUX.PCTL'
----- END OF LIST -----

```

Figure 2-63 EDIT the member mask of the library

Figure 2-64 shows us the two members that match that pattern.

```

File Display Library Settings Menu Utilities Test Help Exit
-----
-IPT--EDIT L1----- SE16661.ELUX.PCTL ----- "A" will display assist
COMMAND ==> SCROLL ==> PAGE
HOTBAR: REFRESH SORT CHA SORT LIB
ON VOLUME SHAR05
NAME RENAME LIB VV.MM CREATED CHANGED SIZE INIT MOD USERID
FTPDNLD 1 01.01 07/09/20 07/10/24 07:14 7 7 0 SE16661
S FTPUPLD 1 01.01 07/09/20 07/09/20 12:14 9 8 4 SE16661
--END--

```

Figure 2-64 Memberlist of members matching the mask

Select the member FTPUPLD. Place an S for Select against the member name, and press Enter, as shown in Figure 2-64. Figure 2-65 on page 41 is displayed.

```

File Edit Edit_Settings Menu Utilities Compilers Test Help
-IPT- EDIT SE16661.ELUX.PCTL (FTPUPLD) - 01.01 Columns 00001 00072
Command ==> Scroll ==> CSR
***** Top of Data *****
000100 CLS
000300 FTP
000400 OPEN 9.212.143.123
000500 SE16661
000610 LLEWEDIH
000611 QUOTE SITE LRECL=4096 RECFM=VB BLKSIZE=32760
000620 PUT C:/TEMP/ASSETS.CSV ELUX.ASSETS.CSV
000630 PUT C:/TEMP/CONTACTS.CSV ELUX.CONTACTS.CSV
000800 QUIT
***** Bottom of Data *****

```

Figure 2-65 Editing the FTP upload member

Press END to save. Figure 2-66 is displayed.

```

File Display Library Settings Menu Utilities Test Help Exit
-----
-IPT--EDIT L1----- SE16661.ELUX.PCTL -----ROW 00002 OF 00002
COMMAND ==> SCROLL ==> PAGE
HOTBAR?
ON VOLUME Z00P02
NAME RENAME LIB VV.MM CREATED CHANGED SIZE INIT MOD USERID
FTPUPLD -NO SAVE 1 01.01 07/09/20 07/09/20 12:14 9 8 4 SE16661
--END--

```

Figure 2-66 Edited member shows that changes were made and these have been saved

Because we made no changes, -NO SAVE is displayed, as shown in Figure 2-66.

Press END again, and you are returned to the OLIST, as shown in Figure 2-67 on page 42.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (B) ----- Asset Management ----- Row 1 to 1 of 1
Command ==> _ SCROLL ==> CSR
Hotbar: FLIP CLRVOL FILLVOL REFRESH UTIL CUT SET UPDATE
Open list ==> ELUX (or BLANK for reference list)
TSO PARMS ==>
Command Member Numbr Data Set Names / Objects Class
-----
-E FTP%LD 1 'SE16661.ELUX.PCTL' PDSE
----- END OF LIST -----

```

Figure 2-67 Return to the OLIST after editing

Next, we add entries for the CSV files that are referenced here and the one we decided will be the output file. The easiest way to do this, because they do not yet exist on the system, is to use the UPDATE command (UPD for short), as shown in Figure 2-68. This is a great feature of OLISTs that gives you complete control over the order of entries and the appearance of the list.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (B) ----- Asset Management ----- Row 1 to 1 of 1
Command ==> UPD SCROLL ==> CSR
Hotbar: FLIP CLRVOL FILLVOL REFRESH UTIL CUT SET UPDATE
Open list ==> ELUX (or BLANK for reference list)
TSO PARMS ==>
Command Member Numbr Data Set Names / Objects Class
-----
-E FTP%LD 1 'SE16661.ELUX.PCTL' PDSE
----- END OF LIST -----

```

Figure 2-68 Using the UPDATE command to edit the OLIST content

Press Enter, and Figure 2-69 on page 43 is displayed.



```

-IPT- ----- ELUX OLIST UPDATE SCREEN -----
COMMAND ==>
                                SCROLL ==> CSR
Use the standard EDIT commands to edit the OLIST entries.
Press the END key to change the OLIST or enter CANCEL to cancel.

ENTRY  MEMBER  DATA SET NAME                                VOLUME
*****
***** Top of Data *****
I 0001 FTP%LD 'SE1661.ELUX.PCTL'
*****
***** Bottom of Data *****

```

Figure 2-69 The OLIST Update panel – Customized Edit panel

We are now in a customized EDIT panel, as shown in Figure 2-70. The panel has fixed tab positions directly before the DATA SET NAME and VOLUME columns.

After the first ENTRY line, insert a line. Here we can see the I for insert is already typed.

We are going to add a generic entry for all of the Excel spreadsheets held for this asset management.

```

-IPT- ----- ELUX OLIST UPDATE SCREEN -----
COMMAND ==>
                                SCROLL ==> CSR
Use the standard EDIT commands to edit the OLIST entries.
Press the END key to change the OLIST or enter CANCEL to cancel.

ENTRY  MEMBER  DATA SET NAME                                VOLUME
*****
***** Top of Data *****
000001 FTP%LD 'SE1661.ELUX.PCTL'
000002      'SE1661*ELUX*CSV'
      -
*****
***** Bottom of Data *****

```

Figure 2-70 The OLIST Update panel – adding a new 'generic' entry

We could press END to save the changes. But, better, let us add some comment lines to make it clear on what activity, or phase within the project, we are working. This is a very useful feature that can help to make each permanent OLIST like your private desktop.

We insert a line above each of lines one and two and add descriptive comments, as shown in Figure 2-71.

```

-IPT- ----- ELUX OLIST UPDATE SCREEN -----
COMMAND ==>
                                SCROLL ==> CSR
Use the standard EDIT commands to edit the OLIST entries.
Press the END key to change the OLIST or enter CANCEL to cancel.

ENTRY  MEMBER  DATA SET NAME                                VOLUME
*****
***** Top of Data *****
000001      !----- FTP Up/Download -----
000002 FTP%LD 'SE1661.ELUX.PCTL'
000003      !----- Spreadsheets -----
000004      'SE1661*ELUX*CSV'
*****
***** Bottom of Data *****

```

Figure 2-71 The OLIST Update panel – adding comments for clarity

Again, we could **Press the END key to change the OLIST**, and save the changes, but we will continue to Edit our OLIST.

The REXX exec will match the two Excel spreadsheet's CSV files and CONTACTS and ASSETS, to create a match file CONASS. To be able to create and edit the REXX exec, we need to add an entry. At the same time, we similarly need to add entries for the JCL job to compile the exec and the JCL job to run the compiled exec. Let us do these all at once under a new section named Program elements, as shown in Figure 2-72.

```
-IPT- ----- ELUX OLIST UPDATE SCREEN -----
COMMAND ==>                                SCROLL ==> CSR
Use the standard EDIT commands to edit the OLIST entries.
Press the END key to change the OLIST or enter CANCEL to cancel.

ENTRY  MEMBER  DATA SET NAME                                VOLUME
***** ***** Top of Data *****
000001      !----- FTP Up/Download -----
000002 FTP%LD  'SE16661.ELUX.PCTL'
000003      !----- Spreadsheets -----
000004      'SE16661*ELUX*CSV'
*****      !----- Program elements -----
000005 CONASS  'SE16661.ELUX.EXEC'
000006 REXXCL  'SE16661.ELUX.JCL'
000007 CONASS  'SE16661.ELUX.JCL'
***** ***** Bottom of Data *****
```

Figure 2-72 The OLIST Update panel – adding entries for the program elements

We want to check the status of the job outputs along the way. Here we could use a User Defined Object, or we could just save a TSO command to check the job output status.

Let us do both.

We carry on editing and adding a new section of 'Job status' and the two objects, as shown in Figure 2-73 on page 45.

```

-IPT- ----- ELUX OLIST UPDATE SCREEN -----
COMMAND ==>                                     SCROLL ==> CSR
Use the standard EDIT commands to edit the OLIST entries.
Press the END key to change the OLIST or enter CANCEL to cancel.

ENTRY  MEMBER  DATA SET NAME                                VOLUME
*****
***** Top of Data *****
000001      !----- FTP Up/Download -----
000002 FTP%LD  'SE16661.ELUX.PCTL'
000003      !----- Spreadsheets -----
000004      'SE16661*ELUX*CSV'
000005      !----- Program elements -----
000006 CONASS  'SE16661.ELUX.EXEC'
000007 REXXCL  'SE16661.ELUX.JCL'
000008 CONASS  'SE16661.ELUX.JCL'
000009      !----- Job status -----
      >OUT SE16661C
      +SDSF ST SE16661C
*****
***** Bottom of Data *****

```

Figure 2-73 The OLIST Update panel – adding job status objects for the OUT UDO and an SDSF command

Finally, plan to use a REXX exec, which runs under UNIX (OMVS) to perform the download and to send the file directly to the contact in asset management. It is still being written, but we want to accommodate it in advance. We have this section that contains the exec in the list so that we can browse the status of it, as shown in Figure 2-74.

```

-IPT- ----- ELUX OLIST UPDATE SCREEN -----
COMMAND ==>                                     SCROLL ==> CSR
Use the standard EDIT commands to edit the OLIST entries.
Press the END key to change the OLIST or enter CANCEL to cancel.

ENTRY  MEMBER  DATA SET NAME                                VOLUME
*****
***** Top of Data *****
000001      !----- FTP Up/Download -----
000002 FTP%LD  'SE16661.ELUX.PCTL'
000003      !----- Spreadsheets -----
000004      'SE16661*ELUX*CSV'
000005      !----- Program elements -----
000006 CONASS  'SE16661.ELUX.EXEC'
000007 REXXCL  'SE16661.ELUX.JCL'
000008 CONASS  'SE16661.ELUX.JCL'
000009      !----- Job status -----
000010      >OUT SE16661C
000011      +SDSF ST SE16661C
      !----- FTP transfer in OMVS -----
      )/U/SE16661/FTPDNLD.REX_
*****
***** Bottom of Data *****

```

Figure 2-74 The OLIST Update panel – adding an OMVS object

Press the END key to SAVE and display the final OLIST, as shown in Figure 2-75.

File	Edit	Find	Display	Populate	Settings	Menu	Util	Test	Help	Exit
-----										
-IPT- OLIST (B)			Asset Management					Row 1 to 13 of 13		
Command ==>								SCROLL ==> CSR		
Hotbar: FLIP			CLRVL	FILLVOL	REFRESH	UTIL	CUT	SET	UPDATE	
Open list ==>			ELUX (or BLANK for reference list)							
TSO PARMS ==>										
Command	Member	Numbr	Data Set Names / Objects						Class	
-----										
		1	!----- FTP Up/Download -----							
	FTP%LD	2	'SE16661.ELUX.PCTL'							
		3	!----- Spreadsheets -----							
		4	'SE16661*ELUX*CSV'						LIST	
		5	!----- Program elements -----							
	CONASS	6	'SE16661.ELUX.EXEC'							
	REXXCL	7	'SE16661.ELUX.JCL'							
	CONASS	8	'SE16661.ELUX.JCL'							
		9	!----- Job status -----							
		10	>OUT SE16661C						USER	
		11	+SDSF ST SE16661C						CMD	
		12	!----- FTP transfer in OMVS -----							
		13	)/U/SE16661/FTPDNLD.REX						OE	
----- END OF LIST -----										

Figure 2-75 The final OLIST displayed

**Note:** We still have to go through and tailor the exec and the JCLs to compile, and run the job, but we will run the job when those elements are complete.

Notice that the object Class is missing from several of the objects on our OLIST. We can neaten it up by either entering the command FILLVOL or placing the cursor on the FILLVOL command (which was previously loaded into the Hotbar) and pressing Enter. Figure 2-76 is displayed.

File	Edit	Find	Display	Populate	Settings	Menu	Util	Test	Help	Exit
-----										
-IPT- OLIST (B) ----- Asset Management ----- "A" will display assist										
Command ==> SCROLL ==> CSR										
Hotbar: FLIP CLRVOL FILLVOL REFRESH UTIL CUT SET UPDATE										
Open list ==> ELUX (or BLANK for reference list)										
TSO PARMS ==>										
Command Member Numbr Data Set Names / Objects Class										

Figure 2-76 Using the FILLVOL command - populating the Class column – Part 1

Press Enter to populate all of the values in the Class column, showing the object types, as displayed in Figure 2-77.

File Edit Find Display Populate Settings Menu Util Test Help Exit				
-----				
-IPT- OLIST (B) ----- Asset Management -----			Row 1 to 13 of 13	
Command ==>			SCROLL ==> CSR	
Hotbar: FLIP CLRVOL FILLVOL REFRESH UTIL CUT SET UPDATE				
Open list ==> ELUX (or BLANK for reference list)				
TSO PARMS ==>				
Command	Member	Numbr	Data Set Names / Objects	Class
-----				
		1	!----- FTP Up/Download -----	
	FTP%LD	2	'SE16661.ELUX.PCTL'	PDSE
		3	!----- Spreadsheets -----	
		4	'SE16661*ELUX*CSV'	LIST
		5	!----- Program elements -----	
	CONASS	6	'SE16661.ELUX.EXEC'	PDSE
	REXXCL	7	'SE16661.ELUX.JCL'	PDSE
	CONASS	8	'SE16661.ELUX.JCL'	PDSE
		9	!----- Job status -----	
		10	>OUT SE16661C	USER
		11	+SDSF ST SE16661C	CMD
		12	!----- FTP transfer in OMVS -----	
		13	) /U/SE16661/FTPDNLD.REX	OE
----- END OF LIST -----				

Figure 2-77 Using the FILLVOL command – populating the Class column – Part 2

Just to understand the rather special generic nature of ISPF-PT patterns, enter a B against line 4 (or place the cursor on the line because Browse is the default command), and press Enter, as shown in Figure 2-78.

File Edit Find Display Populate Settings Menu Util Test Help Exit				
-----				
-IPT- OLIST (B) ----- Asset Management -----			Row 1 to 13 of 13	
Command ==>			SCROLL ==> CSR	
Hotbar: FLIP CLRVOL FILLVOL REFRESH UTIL CUT SET UPDATE				
Open list ==> ELUX (or BLANK for reference list)				
TSO PARMS ==>				
Command	Member	Numbr	Data Set Names / Objects	Class
-----				
		1	!----- FTP Up/Download -----	
	FTP%LD	2	'SE16661.ELUX.PCTL'	PDSE
		3	!----- Spreadsheets -----	
(b)		4	'SE16661*ELUX*CSV'	LIST
		5	!----- Program elements -----	
	CONASS	6	'SE16661.ELUX.EXEC'	PDSE
	REXXCL	7	'SE16661.ELUX.JCL'	PDSE
	CONASS	8	'SE16661.ELUX.JCL'	PDSE
		9	!----- Job status -----	
		10	>OUT SE16661C	USER
		11	+SDSF ST SE16661C	CMD
		12	!----- FTP transfer in OMVS -----	
		13	) /U/SE16661/FTPDNLD.REX	OE
----- END OF LIST -----				

Figure 2-78 Intuitive searches using generic patterns – Part 1

The CSV files that match this intuitive search on the generic pattern are displayed as shown in Figure 2-79 on page 48.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (B) ----- LEVEL SE16661*ELUX*CSV --- "A" will display assist
Command   ==> _                               SCROLL ==> CSR
Hotbar: FLIP   CLRVOL  FILLVOL  REFRESH  UTIL      CUT      SET      UPDATE
                                         *TEMPORARY LIST*

TSO PARMS ==>
Command  Member  Numbr Data Set Names / Objects                                Class
-----
          1 'SE16661.ELUX.ASSETS.CSV'
          2 'SE16661.ELUX.CONASS.CSV'
          3 'SE16661.ELUX.CONTACTS.CSV'
          4 'SE16661.ELUX.UK260907.CSV'
          5 'SE16661.ELUX.US260907.CSV'
          6 'SE16661.ELUX.Z0260907.CSV'
          7 'SE16661.ELUX.ZS260907.CSV'
          8 'SE16661.ELUX.ZU260907.CSV'
          ----- END OF LIST -----

```

Figure 2-79 Intuitive searches using generic patterns – Part 2

## Intuitive searching of generic patterns

This intuitive searching of generic patterns goes much further than the standard ISPF.

It is important to note that if ISPF-PT encounters a pattern that fits the standard ISPF pattern syntax, it uses ISPF's own generic search method; however, if it does not fit the standard ISPF pattern but does meet the ISPF-PT generic pattern syntax, it uses its own method.

The patterns follow a common and well-defined system, which is similar to products such as, ISPF, DFDSS, and VSAM but without some of their restrictions:

- ▶ \* any number of characters, 0+
- ▶ % any character (including dots)

Because the ISPF-PT system is so much less restrictive it is preferable to force this type of search by ensuring that the pattern does not conform to the ISPF standard. Table 2-2 shows some examples.

Table 2-2 Patterns

Generic pattern	Datasets found by intuitive search
SYS%**CLIB Or SYS%.****CLIB	'SYS030.EPS.PROCLIB' 'SYS1.DGTCLIB' 'SYS1.MACLIB' 'SYS1.PROCLIB' 'SYS1.PROCLIB.ZT00PLEX.MOP' 'SYS1.SISTCLIB' 'SYS1.SPROCLIB' 'SYS1.SVCLIB' 'SYS2.PROCLIB' 'SYS2.PROCLIB.FLEXES' 'SYS2.PROCLIB.FO' 'SYS2.PROCLIB.ZT00PLEX.MOP'
SYS1**LIB*.%%%	'SYS1.PARMLIB.ZT00PLEX.MOP' 'SYS1.PROCLIB.ZT00PLEX.MOP'

### 2.3.3 Project ELUX workFlow

In this section, we cover the workflow of this project.

1. Upload CSV files to the host, and then browse (View or Edit) the FTP command members, as shown in Figure 2-80.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (B) ----- Asset Management ----- Row 1 to 13 of 13
Command ==> SCROLL ==> CSR
Hotbar: FLIP CLRVOL FILLVOL REFRESH UTIL CUT SET UPDATE
Open list ==> ELUX (or BLANK for reference list)
TSO PARMS ==>
Command Member Numbr Data Set Names / Objects Class
-----
1 !----- FTP Up/Download -----
b FTP%LD 2 'SE16661.ELUX.PCTL' PDSE
3 !----- Spreadsheets -----

```

Figure 2-80 Browsing the FTP up/download masked memberlist

2. Select the upload, as shown in Figure 2-81.

```

File Display Library Settings Menu Utilities Test Help Exit
-----
-IPT--BROWSE L1---- SE16661.ELUX.PCTL ----- "A" will display assist
COMMAND ==> SCROLL ==> PAGE
HOTBAR?
ON VOLUME Z00P02
NAME RENAME LIB VV.MM CREATED CHANGED SIZE INIT MOD USERID
FTPDNLD 1 01.00 07/09/20 07/09/20 09:42 7 7 0 SE16661
S FTPUPLD 1 01.01 07/09/20 07/09/20 12:14 9 8 4 SE16661
--END--

```

Figure 2-81 Browsing the FTP upload member

3. Select the command lines using the mouse or Ctrl and the arrow keys, as shown in Figure 2-82.

```

Menu Utilities Compilers Help
-----
-IPT- BROWSE SE16661.ELUX.PCTL (FTPUPLD) - 01.01 Line 00000000 Col 001 080
Command ==> Scroll ==> CSR
***** Top of Data *****
CLS 00010000
FTP 00030000
OPEN 9.212.143.123 00040000
SE16661 00050000
LLEWEDIH 00061001
QUOTE SITE LRECL=4096 RECFM=VB BLKSIZE=32760 00061101
PUT C:/TEMP/ASSETS.CSV ELUX.ASSETS.CSV 00062001
PUT C:/TEMP/CONTACTS.CSV ELUX.CONTACTS.CSV 00063001
QUIT 00080000
***** Bottom of Data *****

```

Figure 2-82 Selecting and cutting the FTP upload command

4. Cut and paste them to your workstation PC.
5. Select the prompt line within an MSDOS window, and paste (Right-Click+Paste), as shown in Figure 2-83.

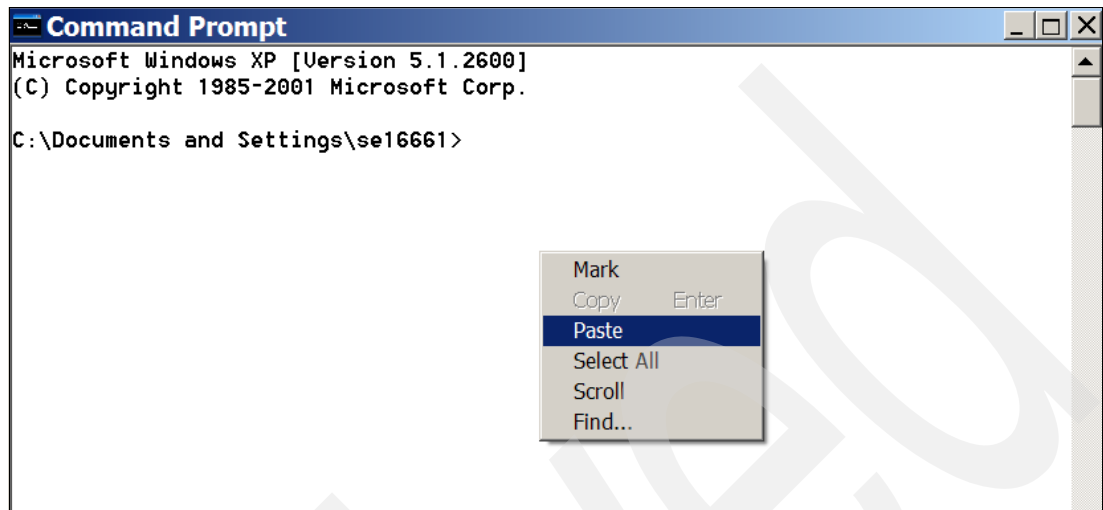


Figure 2-83 Pasting the FTP upload command onto the prompt of an MSDOS window

The files are uploaded as shown in Figure 2-84.

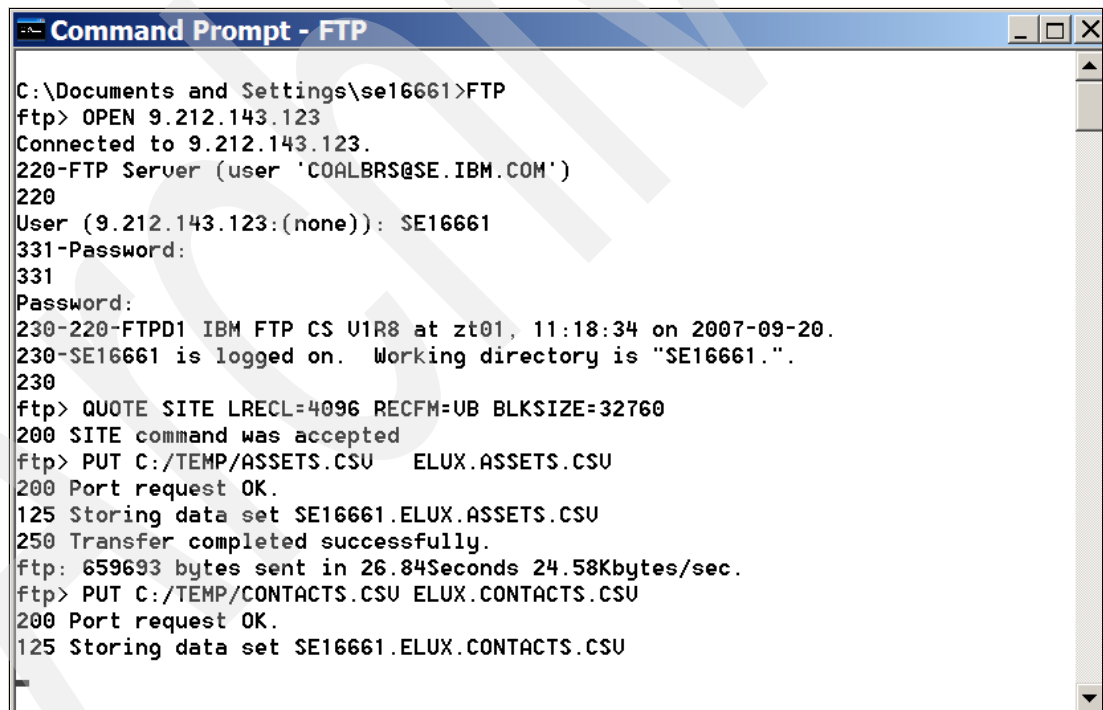


Figure 2-84 FTP uploading the assets and contacts lists saved from EXCEL in CSV format

Looking back to the OLIST, we can check that the two files are uploaded, as shown in Figure 2-85 on page 51. We will skip the actual browsing because these are real files with customer-sensitive data.



```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (B) ----- LEVEL SE16661*ELUX*CSV ----- Row 1 to 8 of 8
Command ===>
Hotbar: FLIP CLRVOL FILLVOL REFRESH UTIL CUT SET UPDATE
*TEMPORARY LIST*

TSO PARMS ===>
Command Member Numbr Data Set Names / Objects Class
-----
-B 1 'SE16661.ELUX.ASSETS.CSV' SEQ
2 'SE16661.ELUX.CONASS.CSV' SEQ
-B 3 'SE16661.ELUX.CONTACTS.CSV' SEQ
4 'SE16661.ELUX.UK260907.CSV'
5 'SE16661.ELUX.US260907.CSV'
6 'SE16661.ELUX.Z0260907.CSV'
7 'SE16661.ELUX.ZS260907.CSV'
8 'SE16661.ELUX.ZU260907.CSV'
----- END OF LIST -----

```

Figure 2-85 Browsing the uploaded FTP files to ensure they were correctly uploaded

6. Modify run JCL to accept new input and output filenames by selecting e against line 8, as shown in Figure 2-86 and Figure 2-87 on page 52.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (B) ----- Asset Management ----- Invalid command:CLRCMD
Command ===>
Hotbar: FLIP CLRVOL FILLVOL REFRESH UTIL CUT SET UPDATE
Open list ===> ELUX (or BLANK for reference list)
TSO PARMS ===>
Command Member Numbr Data Set Names / Objects Class
-----
FTP%LD 1 !----- FTP Up/Download -----
2 'SE16661.ELUX.PCTL' PDSE
3 !----- Spreadsheets -----
4 'SE16661*ELUX*CSV' LIST
5 !----- Program elements -----
CONASS 6 'SE16661.ELUX.EXEC' PDSE
REXXCL 7 'SE16661.ELUX.JCL' PDSE
CONASS 8 'SE16661.ELUX.JCL' PDSE
9 !----- Job status -----
10 >OUT SE16661C USER
11 +SDSF ST SE16661C CMD
12 !----- FTP transfer in OMVS -----
13 )/U/SE16661/FTPDNLD.REX OE
----- END OF LIST -----

```

Figure 2-86 Editing the run JCL to check that the data set names are correct for this file match- Part 1

```

File Edit Edit_Settings Menu Utilities Compilers Test Help
-IPT- EDIT SE16661.ELUX.JCL(CONASS) - 01.06 Columns 00001 00072
Command ==> Scroll ==> CSR
***** Top of Data *****
000001 //SE16661A JOB (DUMMY), 'CONASS',MSGCLASS=H,NOTIFY=&SYSUID,
000002 //      MSGLEVEL=(1,1),CLASS=A,TIME=1440,REGION=0M
000003 //*----- SHORTHANDS -----
000004 //      SET MD='(MOD,DELETE),SPACE=(TRK,0)'
000005 //      SET MC='(MOD,CATLG)'
000006 //      SET NC='(,CATLG)'
000007 //      SET DC='LRECL=4096,RECFM=VB,BLKSIZE=32760'
000008 //*=====
000009 //CONASS  PROC
000010 //*-----
000011 //CLEANUP EXEC PGM=IEFBR14
000012 //SYSREPT DD DISP=&MD,DSN=SE16661.ELUX.CONASS.CSV
000013 //*-----
000014 //CONASS  EXEC PGM=IKJEFT1A,PARM=CONASS
000015 //STEPLIB DD DISP=SHR,DSN=SE16661.ELUX.LOAD
000016 //*YSEEXEC DD DISP=SHR,DSN=SE16661.ELUX.EXEC
000017 //SYSICON DD DISP=SHR,DSN=SE16661.ELUX.CONTACTS.CSV
000018 //SYSIASS DD DISP=SHR,DSN=SE16661.ELUX.ASSETS.CSV
000019 //SYSPRT DD SYSOUT=*,FREE=CLOSE,SPIN=UNALLOC
000020 //SYSOASS DD DISP=&MC,DSN=SE16661.ELUX.CONASS.CSV,
000021 //      SPACE=(CYL,(5,5)),&DC
000022 //SYSTSIN DD DUMMY
000023 //*-----
000024 //LISTCSV EXEC PGM=ICEGENER,COND=(0,LE)
000025 //SYSUT1 DD DISP=SHR,DSN=SE16661.ELUX.CONASS.CSV
000026 //SYSUT2 DD SYSOUT=*
000027 //SYSPRINT DD SYSOUT=*

```

Figure 2-87 Editing the run JCL to check that the data set names are correct for this file match- Part 2

7. In Figure 2-87 we can see that the program name and file names were suitably adjusted. Press END to save the changes and modify and compile the JCL to compile the new exec.
8. Select **E** against line 7, as shown in Figure 2-88.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-IPT- OLIST (B) ----- Asset Management ----- Invalid command:CLRCMD
Command ==> SCROLL ==> CSR
Hotbar: FLIP CLRVOL FILLVOL REFRESH UTIL CUT SET UPDATE
Open list ==> ELUX (or BLANK for reference list)
TSO PARMS ==>
Command Member Numbr Data Set Names / Objects Class
-----
1 !----- FTP Up/Download -----
FTP%LD 2 'SE16661.ELUX.PCTL' PDSE
3 !----- Spreadsheets -----
4 'SE16661*ELUX*CSV' LIST
5 !----- Program elements -----
CONASS 6 'SE16661.ELUX.EXEC' PDSE
REXXCL 7 'SE16661.ELUX.JCL' PDSE
CONASS 8 'SE16661.ELUX.JCL' PDSE
9 !----- Job status -----
10 >OUT SE16661C USER
11 +SDSF ST SE16661C CMD
12 !----- FTP transfer in OMVS -----
13 )/U/SE16661/FTPDNLD.REX OE
----- END OF LIST -----

```

Figure 2-88 Selecting the compile JCL to check that the data set names are correct for this file match

We are now in Edit on the JCL, which compiles and linkedits the REXX exec, as shown in Figure 2-89.

```

File Edit Edit_Settings Menu Utilities Compilers Test Help
-IPT- EDIT SE16661.ELUX.JCL(REXXCL) - 01.03 Columns 00001 00072
Command ==> Scroll ==> CSR
***** ***** Top of Data *****
000001 //SE16661C JOB (DUMMY), 'REXXCL',MSGCLASS=H,
000002 //      MSGLEVEL=(1,1),CLASS=A,TIME=(5,0),NOTIFY=&SYSUID
000003 //JOBPROC JCLLIB ORDER=SE16661.USER.PROC
000004 //REXXCL EXEC REXXCL,NAME=CONASS,
000005 //      HLI='SE16661.ELUX',
000006 //      OPT='OBJECT CEXEC SLINE DLINK SOURCE' XREF'
000007 //*
***** ***** Bottom of Data *****

```

Figure 2-89 Editing the compile JCL to check that the data set names are correct for this file match

The compile/link-edit routine that is invoked by this job is a catalogued procedure. The exec/program name is passed together with the HLI of the program libraries and the compile options.

9. Press END to save the changes, as shown in Figure 2-90.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-IPT- OLIST (B) ----- Asset Management ----- Row 1 to 13 of 13
Command ==> SCROLL ==> CSR
Hotbar: FLIP CLRVOL FILLVOL REFRESH UTIL CUT SET UPDATE
Open list ==> ELUX (or BLANK for reference list)
TSO PARMS ==>
Command Member Numbr Data Set Names / Objects Class
-----
1 !----- FTP Up/Download -----
FTP%LD 2 'SE16661.ELUX.PCTL' PDSE
3 !----- Spreadsheets -----
4 'SE16661*ELUX*CSV' LIST
5 !----- Program elements -----
CONASS 6 'SE16661.ELUX.EXEC' PDSE
REXXCL 7 'SE16661.ELUX.JCL' PDSE
CONASS 8 'SE16661.ELUX.JCL' PDSE
9 !----- Job status -----
10 >OUT SE16661C USER
11 +SDSF ST SE16661C CMD
12 !----- FTP transfer in OMVS -----
13 )/U/SE16661/FTPDNLD.REX OE
----- END OF LIST -----

```

Figure 2-90 Saving the edited compile JCL

10. Create a new REXX exec, as shown in Figure 2-91.

```

File Edit Edit_Settings Menu Utilities Compilers Test Help
-IPT- EDIT SE16661.ELUX.EXEC(CONASS) - 01.22 Columns 00001 00072
Command ==> Scroll ==> CSR
***** ***** Top of Data *****
000001 /*REXX*/
000002
000003 "EXECIO 0 DISKR SYSIASS (OPEN"
000004 IF( RC<>0 )THEN EXIT RC
000005 "EXECIO 0 DISKR SYSICON (OPEN"
000006 IF( RC<>0 )THEN EXIT RC
000007 "EXECIO 0 DISKW SYSOASS (OPEN"
000008 IF( RC<>0 )THEN EXIT RC
000009
000010 PARSE VALUE 0 WITH ass. 1 con. 1 nas.
000011 "EXECIO * DISKR SYSIASS (STEM ASS. FINIS"
000012 SAY "SYSIASS file contains" ass.0 "records."
000013 "EXECIO * DISKR SYSICON (STEM CON. FINIS"
000014 SAY "SYSICON file contains" con.0 "records."
000015 PARSE VALUE ass.1,,,,,,,,,
000016 WITH aa,"ab","ac","ad","ae","af",.
000017 PARSE VALUE con.1,,,,,,,,,
000018 WITH ca,"cb","cc","cd","ce","cf","cg","ch","ci","cj","ck",.
000019 nas1 = aa,"ab","ac","ad","ae","af","ck","ce
000020 "EXECIO 1 DISKW SYSOASS (STEM NAS"
000021 DO a = 2 TO ass.0
000022 PARSE VALUE ass.a,,,,,,,,,
000023 WITH aa,"ab","ac","ad","ae","af",.
000024 IF( ae<>"" & af<>"" )THEN
000025 ckce = LOOKUP_CONTACT(SPACE(ae af))
000026 ELSE
000027 ckce = ""

```

Figure 2-91 Editing the REXX code for the file match program

11. Select **E** against line 6, as shown in Figure 2-92 on page 55. Inside the empty edit member, we can copy an entire previous similar exec, re-write the exec from scratch, or copy sections (Cut+Paste) from earlier ones. When complete, press END to save the changes.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (B) ----- Asset Management ----- Row 1 to 13 of 13
Command ===>
Hotbar: FLIP CLRVOL FILLVOL REFRESH UTIL CUT SET UPDATE
Open list ===> ELUX (or BLANK for reference list)
TSO PARMS ===>
Command Member Numbr Data Set Names / Objects Class
-----
1 !----- FTP Up/Download -----
FTP%LD 2 'SE16661.ELUX.PCTL' PDSE
3 !----- Spreadsheets -----
4 'SE16661*ELUX*CSV' LIST
5 !----- Program elements -----
-E CONASS 6 'SE16661.ELUX.EXEC' PDSE
-E REXXCL 7 'SE16661.ELUX.JCL' PDSE
-E CONASS 8 'SE16661.ELUX.JCL' PDSE
9 !----- Job status -----
10 >OUT SE16661C USER
11 +SDSF ST SE16661C CMD
12 !----- FTP transfer in OMVS -----
13 )/U/SE16661/FTPDNLD.REX OE
----- END OF LIST -----

```

Figure 2-92 Saving the edited REXX code

12. Compile the REXX exec to a load module. Submit the job by typing SUB in the command field on line 7, as shown in Figure 2-93.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (B) ----- Asset Management ----- Row 1 to 13 of 13
Command ===>
Hotbar: FLIP CLRVOL FILLVOL REFRESH UTIL CUT SET UPDATE
Open list ===> ELUX (or BLANK for reference list)
TSO PARMS ===>
Command Member Numbr Data Set Names / Objects Class
-----
1 !----- FTP Up/Download -----
FTP%LD 2 'SE16661.ELUX.PCTL' PDSE
3 !----- Spreadsheets -----
4 'SE16661*ELUX*CSV' LIST
5 !----- Program elements -----
-F CONASS 6 'SE16661.ELUX.EXEC' PDSE
sub REXXCL 7 'SE16661.ELUX.JCL' PDSE
-E CONASS 8 'SE16661.ELUX.JCL' PDSE
9 !----- Job status -----
10 >OUT SE16661C USER
11 +SDSF ST SE16661C CMD
12 !----- FTP transfer in OMVS -----
13 )/U/SE16661/FTPDNLD.REX OE
----- END OF LIST -----

```

Figure 2-93 Submitting the job to compile and linkedit the REXX code

13. Press Enter. In our example, IKJ56250I JOB SE16661C(JOB06746) is SUBMITTED

\*\*\*

followed by

12.49.56 JOB06746 \$HASP165 SE16661C ENDED AT MOPZT00 MAXCC=0 CN(INTERNAL)

\*\*\*

14. Press any key to clear the previous message. Figure 2-94 is displayed.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (B) ----- Asset Management ----- Row 1 to 13 of 13
Command ==>
Hotbar: FLIP CLRVOL FILLVOL REFRESH UTIL CUT SET UPDATE
Open list ==> ELUX (or BLANK for reference list)
TSO PARMS ==>
Command Member Numbr Data Set Names / Objects Class
-----
1 !----- FTP Up/Download -----
2 'SE16661.ELUX.PCTL' PDSE
3 !----- Spreadsheets -----
4 'SE16661*ELUX*CSV' LIST
5 !----- Program elements -----
6 'SE16661.ELUX.EXEC' PDSE
-E CONASS 7 'SE16661.ELUX.JCL' PDSE
-SUB REXXCL 8 'SE16661.ELUX.JCL' PDSE
SUB CONASS 9 !----- Job status -----
10 >OUT SE16661C USER
11 +SDSF ST SE16661C CMD
12 !----- FTP transfer in OMVS -----
13 )/U/SE16661/FTPDNLD.REX OE
----- END OF LIST -----

```

Figure 2-94 Submit for the compile job completed, the command field is updated

15. Run the REXX program. We similarly submit this job by typing SUB in the command field, this time on line 8, and pressing Enter as shown in Figure 2-95.

In our example, IKJ56250I JOB SE16661A(JOB06757) SUBMITTED is returned.

\*\*\*

16. Press any key to clear the previous message. Figure 2-95 is displayed.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (B) ----- Asset Management ----- Row 1 to 13 of 13
Command ==>
Hotbar: FLIP CLRVOL FILLVOL REFRESH UTIL CUT SET UPDATE
Open list ==> ELUX (or BLANK for reference list)
TSO PARMS ==>
Command Member Numbr Data Set Names / Objects Class
-----
1 !----- FTP Up/Download -----
2 'SE16661.ELUX.PCTL' PDSE
3 !----- Spreadsheets -----
4 'SE16661*ELUX*CSV' LIST
5 !----- Program elements -----
6 'SE16661.ELUX.EXEC' PDSE
-E CONASS 7 'SE16661.ELUX.JCL' PDSE
-SUB REXXCL 8 'SE16661.ELUX.JCL' PDSE
-SUB CONASS 9 !----- Job status -----
10 >OUT SE16661C USER
11 +SDSF ST SE16661C CMD
12 !----- FTP transfer in OMVS -----
13 )/U/SE16661/FTPDNLD.REX OE
----- END OF LIST -----

```

Figure 2-95 Submit for the run job completed, the command field is updated

Now we can check the status of the job by using either the custom UDO, OUT on line 10, or by invoking SDSF with the status parameters on line 11.

1. Type **D** for DISPLAY against the customized UDO (notice the > prefix) with the “OUT” verb, and press Enter, as shown in Figure 2-96.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (B) ----- Asset Management ----- Row 1 to 13 of 13
Command ===>
Hotbar: FLIP CLRVOL FILLVOL REFRESH UTIL CUT SET UPDATE
Open list ===> ELUX (or BLANK for reference list)
TSO PARMS ===>
Command Member Numbr Data Set Names / Objects Class
-----
FTP%LD 1 !----- FTP Up/Download ----- PDSE
2 'SE16661.ELUX.PCTL'
3 !----- Spreadsheets -----
4 'SE16661*ELUX*CSV' LIST
5 !----- Program elements -----
6 'SE16661.ELUX.EXEC' PDSE
7 'SE16661.ELUX.JCL' PDSE
8 'SE16661.ELUX.JCL' PDSE
9 !----- Job status -----
10 >OUT SE16661C USER
11 +SDSF ST SE16661C CMD
12 !----- FTP transfer in OMVS -----
13 )/U/SE16661/FTPDNLD.REX OE
----- END OF LIST -----

```

Figure 2-96 Display the status of the executing file match job by invoking the OUT UDO – Part 1

This puts the status of the job in the short message field. Press HELP for more detail. If more than one job names SE16661A were found, the first four are listed, as shown in Figure 2-97.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (B) ----- Asset Management ----- EXECUTING
Command ===>
Hotbar: FLIP CLRVOL FILLVOL REFRESH UTIL CUT SET UPDATE
Open list ===> ELUX (or BLANK for reference list)
TSO PARMS ===>
Command Member Numbr Data Set Names / Objects Class
-----
FTP%LD 1 !----- FTP Up/Download ----- PDSE
2 'SE16661.ELUX.PCTL'
3 !----- Spreadsheets -----
4 'SE16661*ELUX*CSV' LIST
5 !----- Program elements -----
6 'SE16661.ELUX.EXEC' PDSE
7 'SE16661.ELUX.JCL' PDSE
8 'SE16661.ELUX.JCL' PDSE
9 !----- Job status -----
10 >OUT SE16661C USER
11 +SDSF ST SE16661C CMD
12 !----- FTP transfer in OMVS -----
13 )/U/SE16661/FTPDNLD.REX OE
----- END OF LIST -----

```

Figure 2-97 Display the status of the executing file match job by invoking the OUT UDO – Part 2

In Figure 2-97, we can see that the job is still executing. These are very big files, which take time even for a compiled REXX program.

2. Type **S** against line 11, and press Enter, as shown in Figure 2-98.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (B) ----- Asset Management ----- EXECUTING
Command ==>
Hotbar: FLIP CLRVOL FILLVOL REFRESH UTIL CUT SET UPDATE
Open list ==> ELUX (or BLANK for reference list)
TSO PARMS ==>
Command Member Numbr Data Set Names / Objects Class
-----
1 !----- FTP Up/Download -----
2 'SE16661.ELUX.PCTL' PDSE
3 !----- Spreadsheets -----
4 'SE16661*ELUX*CSV' LIST
5 !----- Program elements -----
6 'SE16661.ELUX.EXEC' PDSE
7 'SE16661.ELUX.JCL' PDSE
8 'SE16661.ELUX.JCL' PDSE
9 !----- Job status -----
10 >OUT SE16661C USER
11 +SDSF ST SE16661C CMD
12 !----- FTP transfer in OMVS -----
13 )/U/SE16661/FTPDNLD.REX OE
----- END OF LIST -----

```

Figure 2-98 Status command

This invokes the TSO command (notice the + prefix) SDSF with the parameters following, which requests a status list of all jobs that begin with user ID SE16661, as show in Figure 2-99.

```

SDSF STATUS DISPLAY ALL CLASSES LINE 1-7 (7)
COMMAND INPUT ==> SCROLL ==> PAGE
NP JOBNAM JobID Owner Prty Queue C Pos SAff ASys Status
SE16661A JOB01236 SE16661 6 EXECUTION A ZT02
SE16661 TSU00925 SE16661 15 EXECUTION ZT01 ZT01
SE16661S JOB00322 SE16661 1 PRINT A 678
SE16661C JOB00324 SE16661 1 PRINT A 679
SE16661S JOB00325 SE16661 1 PRINT A 705
SE16661 TSU00314 SE16661 1 PRINT 725
SE16661C JOB01229 SE16661 1 PRINT A 1061

```

Figure 2-99 Display the status of the executing file match job by invoking the SDSF command

3. Clear up some of the jobs, and then press END repeatedly until you return to the IPT OLIST display, as shown in Figure 2-100 on page 59.



File	Edit	Find	Display	Populate	Settings	Menu	Util	Test	Help	Exit
-----										
-IPT- OLIST (B)			Asset Management				Row 1 to 13 of 13			
Command ==>							SCROLL ==> CSR			
Hotbar: FLIP			CLRVOL	FILLVOL	REFRESH	UTIL	CUT	SET	UPDATE	
Open list ==>			ELUX (or BLANK for reference list)							
TSO PARMS ==>										
Command	Member	Numbr	Data Set Names / Objects						Class	
-----										
		1	!----- FTP Up/Download -----							
	FTP%LD	2	'SE16661.ELUX.PCTL'						PDSE	
		3	!----- Spreadsheets -----							
		4	'SE16661*ELUX*CSV'						LIST	
		5	!----- Program elements -----							
	CONASS	6	'SE16661.ELUX.EXEC'						PDSE	
	REXXCL	7	'SE16661.ELUX.JCL'						PDSE	
	CONASS	8	'SE16661.ELUX.JCL'						PDSE	
		9	!----- Job status -----							
-D		10	>OUT SE16661C						USER	
-OK		11	+SDSF ST SE16661C						CMD	
		12	!----- FTP transfer in OMVS -----							
		13	)/U/SE16661/FTPDNLD.REX						OE	
----- END OF LIST -----										

Figure 2-100 Command field of the OLIST updated for both the UDO and SDSF invocations

- Review the output CSV file. The matches usually take some time to run. In our example, the following message is displayed:

```
13.16.27 JOB06757 $HASP165 SE16661A ENDED AT MOPZT00 MAXCC=1 CN(INTERNAL)
***
```

The REXX exec CONASS, which this job ran, returns a code **1** if matches are found (similar to the SuperC program ISRSUPC found in standard ISPF options 3.12-15); therefore, we do not need to check the listing, but it is good practice to do so.

- Select object **11** again to view the status of the jobs, and select the appropriate job on the HELD OUTPUT queue. Figure 2-101 on page 60 is displayed.

```

SDSF OUTPUT DISPLAY SE16661A JOB000063 DSID      2 LINE 0      COLUMNS 02- 81
COMMAND INPUT ==> _      SCROLL ==> PAGE
***** TOP OF DATA *****
              J E S 2   J O B   L O G   --   S Y S T E M   Z T 0 1   --   N O D E

17.23.55 JOB000063 ---- WEDNESDAY, 24 OCT 2007 ----
17.23.55 JOB000063 IRR010I  USERID SE16661  IS ASSIGNED TO THIS JOB.
17.23.55 JOB000063 ICH70001I SE16661  LAST ACCESS AT 16:53:01 ON WEDNESDAY, OCTO
17.23.55 JOB000063 $HASP373 SE16661A STARTED - INIT 2    - CLASS A - SYS ZT02
17.23.55 JOB000063 IEF403I SE16661A - STARTED - TIME=17.23.55
17.23.55 JOB000063 -                                     --TIMINGS (MINS.)--
17.23.55 JOB000063 -JOBNAME  STEPNAME  PROCSTEP      RC    EXCP    CPU    SRB    CLOCK
17.23.55 JOB000063 -SE16661A CONASS    CLEANUP      00      10     .00     .00     .00
17.33.15 JOB000063 -SE16661A CONASS    CONASS       01     269    9.04     .00    9.33
17.33.15 JOB000063 -SE16661A CONASS    LISTCSV  FLUSH      0     .00     .00     .00
17.33.15 JOB000063 IEF404I SE16661A - ENDED - TIME=17.33.15
17.33.15 JOB000063 -SE16661A ENDED.  NAME=CONASS          TOTAL CPU TIME=
17.33.15 JOB000063 $HASP395 SE16661A ENDED
----- JES2 JOB STATISTICS -----
      24 OCT 2007 JOB EXECUTION DATE
          33 CARDS READ
          186 SYSOUT PRINT RECORDS
           0 SYSOUT PUNCH RECORDS
          12 SYSOUT SPOOL KBYTES
          9.33 MINUTES EXECUTION TIME
      1 //SE16661A JOB (DUMMY), 'CONASS',MSGCLASS=H,NOTIFY=&SYSUID,
        //          MSGLEVEL=(1,1),CLASS=A,TIME=1440,REGION=0M
        /*----- SHORTHANDS -----
          IEFC653I SUBSTITUTION JCL - (DUMMY), 'CONASS',MSGCLASS=H,NOTIFY=SE16661
          REGION=0M
      2 //          SET MD= ' (MOD,DELETE) ,SPACE= (TRK,0) '
      3 //          SET MC= ' (MOD,CATLG) '

```

*Figure 2-101* Checking the job output using the SDSF Status command and viewing the job on the held output queue

Leaving SDSF and returning to the OLIST, we will now browse the generic spreadsheet files.

1. Enter the B command to browse the generic spreadsheet files, as shown in Figure 2-102 on page 61.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (B) ----- LEVEL SE16661*ELUX*CSV --- "A" will display assist
Command ==>
Hotbar: FLIP      CLRVOL  FILLVOL  REFRESH  UTIL      CUT      SET      UPDATE
                                         *TEMPORARY LIST*

TSO PARMS ==>
Command  Member  Numbr Data Set Names / Objects                                Class
-----
1 'SE16661.ELUX.ASSETS.CSV'
2 'SE16661.ELUX.CONASS.CSV'
3 'SE16661.ELUX.CONTACTS.CSV'
4 'SE16661.ELUX.UK260907.CSV'
5 'SE16661.ELUX.US260907.CSV'
6 'SE16661.ELUX.Z0260907.CSV'
7 'SE16661.ELUX.ZS260907.CSV'
8 'SE16661.ELUX.ZU260907.CSV'
----- END OF LIST -----

```

Figure 2-102 Browse the results of the file match – select the generic spreadsheets – Part-1

2. Select the output, and Figure 2-103 is displayed.

```

Menu Utilities Compilers Help
-----
-IPT- BROWSE SE16661.ELUX.CONASS.CSV Line 00000000 Col 001 080
Command ==>
***** Top of Data *****
ComputerName, SerialNumber, Manufacturer, Model, Surname, First name, Site, Contact ID

```

Figure 2-103 Browse the results of the file match – Looking at the output

Again, here we cannot show it because of the sensitive information it may contain.

- Download the output CSV to the PC. As we did with the upload FTP command, return to the top of the OLIST, and browse (View or Edit) the FTP commands in member FTPDNLD from the member selection list.
- Select the command lines. Figure 2-104 is displayed.

```

Menu Utilities Compilers Help
-----
-IPT- BROWSE SE16661.ELUX.PCTL (FTPDNLD) - 01.00 Line 00000000 Col 001 080
Command ==>
***** Top of Data *****
CLS 00010000
FTP 00020000
OPEN 9.212.143.123 00030000
SE16661 00040000
LLEWEDIH 00050000
GET ELUX.CONASS.CSV C:/TEMP/CONASS.CSV 00060000
QUIT 00080000
***** Bottom of Data *****

```

Figure 2-104 Browsing the FTP download member. Selecting and cutting the FTP upload command

5. Cut and paste the command lines to your PC workstation, as shown in Figure 2-105.
6. Select the prompt line within an MSDOS window, and paste (Right-Click+Paste), as shown in Figure 2-105.

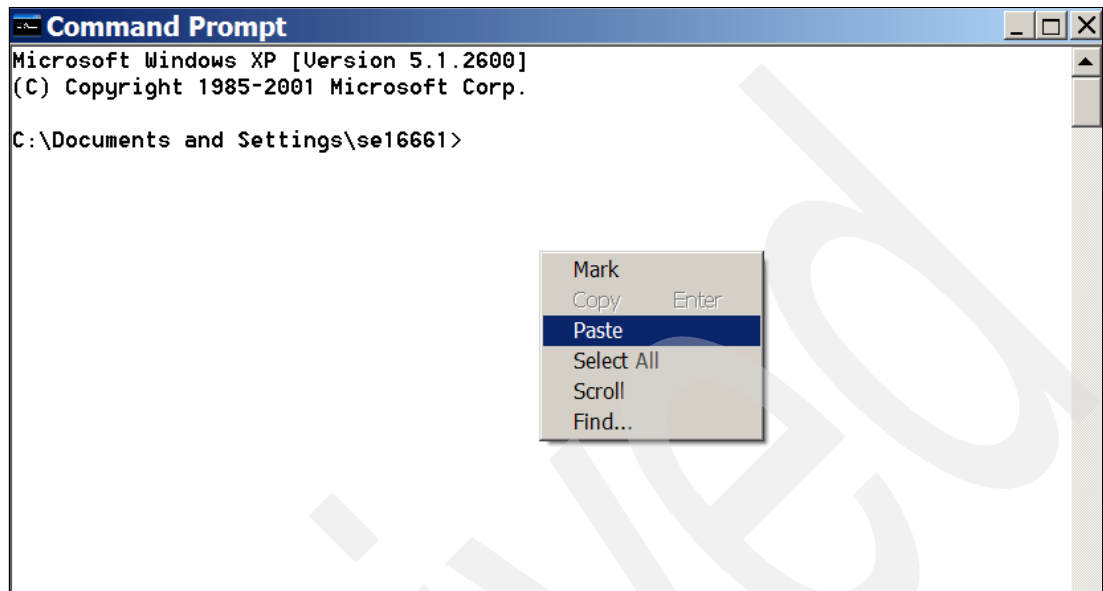


Figure 2-105 Pasting the FTP download command onto the prompt of an MSDOS window

The files get downloaded, as shown in Figure 2-106.

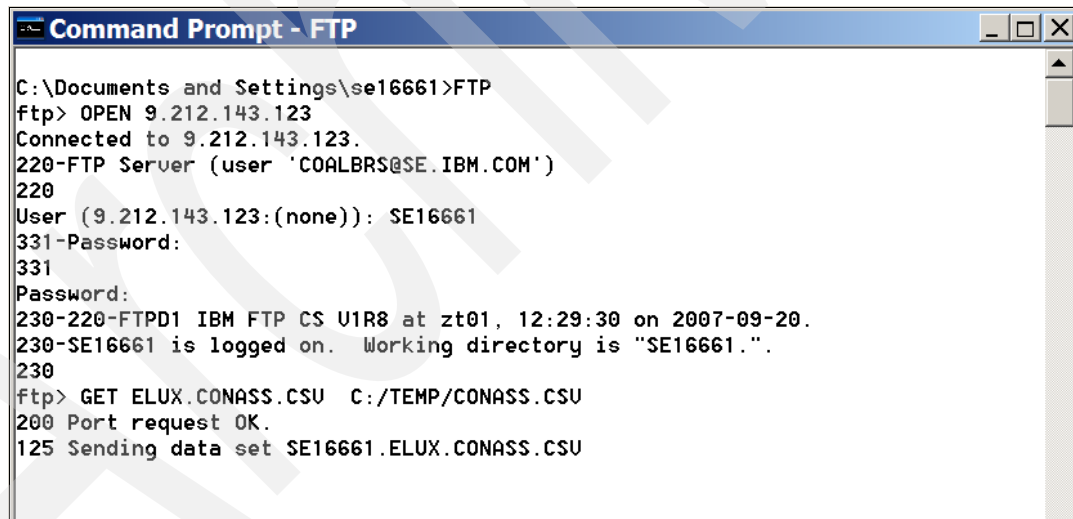


Figure 2-106 FTP downloading the result of the file match in CSV format

Then the programmer rapidly sends the output back to the audit team as a Lotus Notes attachment to use, or to request modifications to the process, and the cycle is repeated.

7. But remember, we are still waiting to be told that the UNIX download-and-send routine is ready for use. Browse it by typing B against line 13, as shown in Figure 2-107 on page 63.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (B) ----- Asset Management ----- Row 1 to 13 of 13
Command ==>
Hotbar: FLIP CLRVOL FILLVOL REFRESH UTIL CUT SET UPDATE
Open list ==> ELUX (or BLANK for reference list)
TSO PARMS ==>
Command Member Numbr Data Set Names / Objects Class
-----
FTP%LD 1 !----- FTP Up/Download ----- PDSE
2 'SE16661.ELUX.PCTL'
3 !----- Spreadsheets -----
4 'SE16661*ELUX*CSV' LIST
5 !----- Program elements -----
CONASS 6 'SE16661.ELUX.EXEC' PDSE
REXXCL 7 'SE16661.ELUX.JCL' PDSE
CONASS 8 'SE16661.ELUX.JCL' PDSE
9 !----- Job status -----
10 >OUT SE16661C USER
11 +SDSF ST SE16661C CMD
12 !----- FTP transfer in OMVS -----
13 )/U/SE16661/FTPDNLD.REX OE
----- END OF LIST -----

```

Figure 2-107 Browsing the HFS file holding the REXX exec to see the stage of development – Part 1

Figure 2-108 results showing an error.

```

----- BROWSE - ENTRY PANEL -----
Command ==>

Directory ==> /U/SE16661/

Filename ==> FTPDNLD.REX

If file is to be browsed as fixed length records:
Record length ==> Leave blank to browse delimited text files

Errno=79x The parameter is incorrect; Reason=055B005Bx The file system is
already mounted. Press Enter to continue.

```

Figure 2-108 Browsing the HFS file holding the REXX exec to see the stage of development – Part 2

What can the problem be? Well, now we can see, the filename is in uppercase.

**Note:** Many people familiar with TSO make this mistake. TSO folds all names to uppercase, but UNIX does not, and there is a difference between ABC and abc.

- Go back and change the entry to lowercase. (We do not show this procedure, but you either overtype the line in lowercase or UPDATE the OLIST, and use the LC line command.) Browse again, as shown in Figure 2-109.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (B) ----- Asset Management ----- Row 1 to 13 of 13
Command ===>
Hotbar: FLIP      CLRVOL  FILLVOL  REFRESH  UTIL      CUT      SET      UPDATE
Open list ===> ELUX      (or BLANK for reference list)
TSO PARMS ===>
Command  Member  Numbr  Data Set Names / Objects  Class
-----
          1 !----- FTP Up/Download -----
          2 'SE16661.ELUX.PCTL'          PDSE
          3 !----- Spreadsheets -----
          4 'SE16661*ELUX*CSV'          LIST
          5 !----- Program elements -----
          6 'SE16661.ELUX.EXEC'        PDSE
          7 'SE16661.ELUX.JCL'        PDSE
          8 'SE16661.ELUX.JCL'        PDSE
          9 !----- Job status -----
         10 >OUT SE16661C              USER
         11 +SDSF ST SE16661C          CMD
         12 !----- FTP transfer in OMVS -----
         13 )/u/se16661/ftpdnld.rex    OE
          ----- END OF LIST -----

```

Figure 2-109 Browsing the HFS file holding the REXX exec to see the stage of development – Part 3

- Press Enter, and Figure 2-110 is displayed.

```

BROWSE -- /u/se16661/ftpdnld.rex ----- Line 00000000 Col 001 018
Command ===>
***** Top of Data *****
/*REXX*/ TRACE "I"
SAY "Hello World"
***** Bottom of Data *****

```

Figure 2-110 Browsing the HFS file holding the REXX exec to see the stage of development – Part 4

OK! We clearly see that the promised code has not yet been delivered – just a stub!

We continue this scenario in Appendix C, “Permanent OLISTs and UNIX System Services objects” on page 299.

## 2.4 Object List MEMFIND command

Using the MEMFIND command within an OLIST, you can search multiple PDS or PDSE datasets for a specific member name or even search for member names that begin with a pattern followed by an asterisk.

The MEMFIND or MF command locates the first, next, or all libraries in the OLIST that contain the specified member.

Figure 2-111 shows an OLIST with the MEMFIND (MF) command entered without a text string specified. On this panel, you enter the search argument, and you can also limit the search scope:

- NEXT** Searches the list for the next library containing the specified member. This is the default.
- FIRST** Searches the list in order, and stops as soon as the library is found that contains the member.
- ALL** Searches all libraries for the specified member.

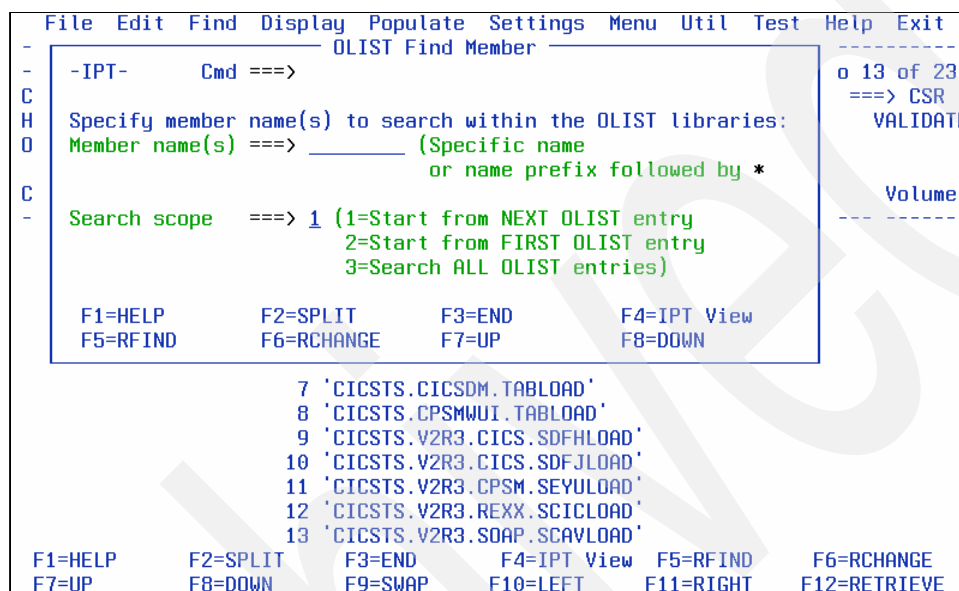


Figure 2-111 MEMFIND search window

Figure 2-112 shows an OLIST before you press the Enter key on a MEMFIND (MF) command with a specified text string.

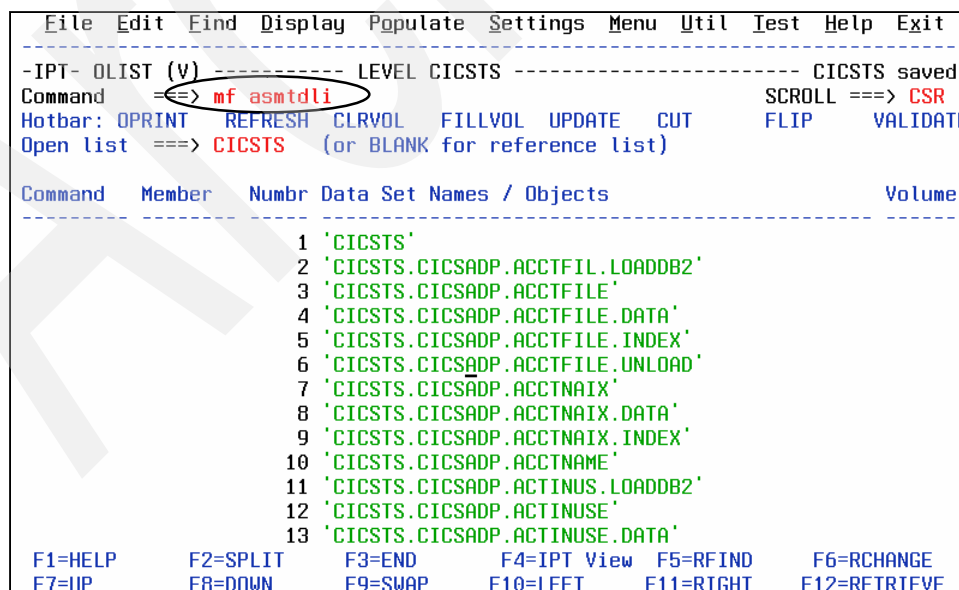


Figure 2-112 OLIST MEMFIND command

Figure 2-113 shows the first library in the OLIST that contained the MEMFIND (MF) string. The RFIND PF3 key, or command, advances to the next data set in the OLIST.

File	Edit	Find	Display	Populate	Settings	Menu	Util	Test	Help	Exit	
-----											
-IPT- OLIST (V)			-----			LEVEL CICSTS		-----			Row 392 to 404 of 2,000
Command ==>											SCROLL ==> CSR
Hotbar: OPRINT			REFRESH	CLRVL	FILLVOL	UPDATE	CUT	FLIP	VALIDATE		
Open list ==>			CICSTS (or BLANK for reference list)								
-----											
Command	Member	Numbr	Data Set Names / Objects							Volume	
-----											
--FOUND--	ASMTDLI	392	'CICSTS.V2R3.CICS.SDFHLOAD'							DMPP30	
		393	'CICSTS.V2R3.CICS.SDFHMAC'								
		394	'CICSTS.V2R3.CICS.SDFHMLIB'								
		395	'CICSTS.V2R3.CICS.SDFHMSG'								
		396	'CICSTS.V2R3.CICS.SDFHMSRC'								
		397	'CICSTS.V2R3.CICS.SDFHPARM'								
		398	'CICSTS.V2R3.CICS.SDFHPLIB'								
		399	'CICSTS.V2R3.CICS.SDFHPL1'								
		400	'CICSTS.V2R3.CICS.SDFHPROC'								
		401	'CICSTS.V2R3.CICS.SDFHSAMP'								
		402	'CICSTS.V2R3.CICS.SDFHSDCK'								
-----											
IQIP1030 Member ASMTDLI found in 'CICSTS.V2R3.CICS.SDFHLOAD'											
-----											
F1=H										HANGE	
F7=UP	F8=DOWN	F9=SWAP	F10=LEFT	F11=RIGHT	F12=RETRIEVE						

Figure 2-113 Results of MEMFIND command

If you run a MEMFIND (MF) command and you do not specify ALL, you must RFIND through out the entire OLIST. If it is a large OLIST, rather than paging up and down to find the libraries that contained the member, you can use the SORT MEM command, and it moves the libraries that contain the member to the top of the OLIST.

Figure 2-114 depicts the results of a MEMFIND command without the word ALL, followed by several RFIND commands, and then the command SORT MEM. The libraries are re-sequenced when you use the SORT MEM command.

File	Edit	Find	Display	Populate	Settings	Menu	Util	Test	Help	Exit
<hr/>										
-IPT- OLIST (V)			----- LEVEL CICSTS -----			"A" will display assist				
Command ==>			SCROLL ==> CSR							
Hotbar: OPRINT			REFRESH	CLRVL	FILLVOL	UPDATE	CUT	FLIP	VALIDATE	
Open list ==>			CICSTS (or BLANK for reference list)							
<hr/>										
Command	Member	Numbr	Data Set Names / Objects							Volume
<hr/>										
--FOUND--	ASMTDLI	1	'CICSTS.V2R3.CICS.SDFHLOAD'							DMPP30
--FOUND--	ASMTDLI	2	'CICSTS.V3R1.CICS.SDFHLOAD'							DMPP04
		3	'CICSTS.CICSADP.ACCTFIL.LOADDB2'							DMPU09
		4	'CICSTS.CICSADP.ACCTFILE.DATA'							
		5	'CICSTS.CICSADP.ACCTFILE.INDEX'							
		6	'CICSTS.CICSADP.ACCTFILE.UNLOAD'							DMPU14
		7	'CICSTS.CICSADP.ACCTFILE'							
		8	'CICSTS.CICSADP.ACCTNAIX.DATA'							
		9	'CICSTS.CICSADP.ACCTNAIX.INDEX'							
<hr/>										
F1=HELP		F2=SPLIT		F3=END		F4=IPT View		F5=RFIND		F6=RCHANGE
F7=UP		F8=DOWN		F9=SWAP		F10=LEFT		F11=RIGHT		F12=RETRIEVE
<hr/>										
Display		Filter	View	Print	Options	Help				
<hr/>										
SDSF OUTPUT DISPLAY CICSTIVX STC03899 DSID 3 LINE 84 COLS 02- 81										

Figure 2-114 MEMFIND results



Figure 2-115 depicts the finished MEMFIND (MF) ALL command with the GLOBAL EXCLUDE set to YES.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-IPT- OLIST (V) ----- LEVEL CICSTS -----
Command ==> MF ASMTDLI ALL SCROLL ==> CSR
Hotbar: OPRINT REFRESH CLRVOL FILLVOL UPDATE CUT FLIP VALIDATE
Open list ==> CICSTS (or BLANK for reference list)

Command Member Numbr Data Set Names / Objects Volume
-----
1 'CICSTS'
2 'CICSTS.CICSADP.ACCTFIL.LOADDB2'
3 'CICSTS.CICSADP.ACCTFILE'
4 'CICSTS.CICSADP.ACCTFILE.DATA'
5 'CICSTS.CICSADP.ACCTFILE.INDEX'
6 'CICSTS.CICSADP.ACCTFILE.UNLOAD'
7 'CICSTS.CICSADP.ACCTNAIX'
8 'CICSTS.CICSADP.ACCTNAIX.DATA'
9 'CICSTS.CICSADP.ACCTNAIX.INDEX'
10 'CICSTS.CICSADP.ACCTNAME'
11 'CICSTS.CICSADP.ACTINUS.LOADDB2'

IQIP1029 Searching for member ASMTDLI (75 libraries searched)

F1=H F7=UP F8=DOWN F9=SWAP F10=LEFT F11=RIGHT F12=RETRIEVE ANGE

```

Figure 2-115 MEMFIND Search progress message

Figure 2-116 depicts the finished MEMFIND (MF) ALL command with the GLOBAL EXCLUDE set to YES. You can populate the volume column using the FILLVOL command. To see all of the commands, type the ASSIST (A) command.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-IPT- OLIST (V) ----- LEVEL CICSTS ----- Row 1 from 2000
Command ==> SCROLL ==> CSR
Hotbar: OPRINT REFRESH CLRVOL FILLVOL UPDATE CUT FLIP VALIDATE
Open list ==> CICSTS (or BLANK for reference list)

Command Member Numbr Data Set Names / Objects *EXCLUDE* Volume
-----
--FOUND-- ASMTDLI 1 'CICSTS.V2R3.CICS.SDFHLOAD' DMPP30
--FOUND-- ASMTDLI 2 'CICSTS.V3R1.CICS.SDFHLOAD' DMPP04
                        END OF LIST -----

IQIP1026 Member ASMTDLI found in 2 libraries

F1=HELP F7=UP F8=DOWN F9=SWAP F10=LEFT F11=RIGHT F12=RETRIEVE F6=RCHANGE
Display Filter View Print Options Help

SDSF OUTPUT DISPLAY CICSTIVX STC03899 DSID 3 LINE 84 COLS 02- 81

```

Figure 2-116 MEMFIND command with GLOBAL EXCLUDE

You can set GLOBAL EXCLUDE using the ISET primary command, and then select the Global option. If you specify ALL and you have the GLOBAL parameter of EXCLUDE set to YES at the end of the search, only the libraries that contained the member are displayed. The sequence number of the libraries did not change. Use the FLIP command to see the libraries that are excluded because they did not contain the member, as shown in Figure 2-117 on page 68.

File Edit Find Display Populate Settings Menu Util Test Help Exit									
-----									
-IPT- OLIST (V)		----- LEVEL CICSTS -----					Row 1 from 2000		
Command ==>							SCROLL ==> CSR		
Hotbar: OPRINT		REFRESH		CLRVOL		FILLVOL		UPDATE CUT	
Open list ==>		CICSTS		(or BLANK for reference list)					
								FLIP VALIDATE	
-----									
Command	Member	Numbr	Data Set Names / Objects					*EXCLUDE*	Volume
-----									
		1	'CICSTS'						
		2	'CICSTS.CICSADP.ACCTFIL.LOADDB2'						DMPU09
		3	'CICSTS.CICSADP.ACCTFILE'						
		4	'CICSTS.CICSADP.ACCTFILE.DATA'						
		5	'CICSTS.CICSADP.ACCTFILE.INDEX'						
		6	'CICSTS.CICSADP.ACCTFILE.UNLOAD'						DMPU14
		7	'CICSTS.CICSADP.ACCTNAIX'						
		8	'CICSTS.CICSADP.ACCTNAIX.DATA'						
		9	'CICSTS.CICSADP.ACCTNAIX.INDEX'						
		10	'CICSTS.CICSADP.ACCTNAME'						
		11	'CICSTS.CICSADP.ACTINUS.LOADDB2'						DMPU16
-----									
IQIP1012 Visible rows excluded. Invisible rows revealed.									
-----									
F1=HEL								RCHANGE	
F7=UP		F8=DOWN		F9=SWAP		F10=LEFT		F11=RIGHT	
								F12=RETRIEVE	

Figure 2-117 Result of the FLIP command

To remove the text box in Figure 2-117, press the Enter key.

Figure 2-118 shows an OLIST before you press the Enter key on a MEMFIND (MF) command with a wildcard or a member name that begins with a pattern followed by an asterisk. The MEMFIND (MF) 'wildcard' commands are the same as a specific member search.

File Edit Find Display Populate Settings Menu Util Test Help Exit									
-IPT- OLIST (V) ----- LEVEL CICSTS ----- Row 1 to 13 of 4,147									
Command ==> mf dfh0* SCROLL ==> CSR									
Hotbar: OPRINT REFRESH CLRVOL FILLVOL UPDATE CUT FLIP VALIDATE									
*TEMPORARY LIST*									
Command	Member	Numbr	Data Set Names / Objects						Volume
		1	'CICSTS'						
		2	'CICSTS.CICSADP.ACCTFIL.LOADDB2'						DMPU09
		3	'CICSTS.CICSADP.ACCTFILE'						
		4	'CICSTS.CICSADP.ACCTFILE.DATA'						DMPD27
		5	'CICSTS.CICSADP.ACCTFILE.INDEX'						DMPD27
		6	'CICSTS.CICSADP.ACCTFILE.UNLOAD'						DMPU14
		7	'CICSTS.CICSADP.ACCTNAIX'						
		8	'CICSTS.CICSADP.ACCTNAIX.DATA'						DMPD28
		9	'CICSTS.CICSADP.ACCTNAIX.INDEX'						DMPD28
		10	'CICSTS.CICSADP.ACCTNAME'						
		11	'CICSTS.CICSADP.ACTINUS.LOADDB2'						DMPU16
		12	'CICSTS.CICSADP.ACTINUSE'						
		13	'CICSTS.CICSADP.ACTINUSE.DATA'						DMPD22
F1=HELP	F2=SPLIT	F3=END	F4=IPT View	F5=RFIND	F6=RCHANGE				
F7=UP	F8=DOWN	F9=SWAP	F10=LEFT	F11=RIGHT	F12=RETRIEVE				

Figure 2-118 MEMFIND example with a wild card

When you complete your work in the OLIST, you can end or exit the panel using PF3.

If you do not reset the command, member field, or excluded libraries, when you return to the OLIST it will still have the members noted and libraries excluded. Use the RESET (RES) command to clear selected columns or reveal hidden objects in an OLIST display.

You can also use the REFRESH (REF) command to refresh the current OLIST from its original parameters. Some helpful commands are:

- 'RES'                      Clears the command column.
- 'RES M'                    Clears the member column.
- 'RES EXCLUDE'            Shows all of the libraries in the OLIST.
- 'RES ALL'                 Resets all of the fields, and returns the OLIST to its original state.

## 2.5 Object List FINDTEXT command

Using FINDTEXT on an OLIST, you can search Sequential, PDS, and PDSE datasets, up to 20,000 for a text string, and return the results with incredible speed.

**Note:** DB2 tables, VSAM files, UNIX System Services files, embedded Object Lists, and dynamic Object Lists are bypassed when you use the FINDTEXT command.

The FINDTEXT or FT command locates the first occurrence of the data string that you specified within an OLIST. If the data string contains blanks or special characters, enclose the text-string in quotation marks. The FINDTEXT command is not case-sensitive, for example the command FIND TRACE returns the lines with the word TRACE, trace, or any combination of upper and lower case.

The OLIST FINDTEXT command uses the same basic syntax as the MSL FINDTEXT but stops the first time the string is found. So if there are multiple members that contain the string in one library, an OLIST FT stops at the first member from that library. Because the search stops at the first member within the library, to search for multiple members, you switch into member list, and use the MSL GLOBAL FIND or FINDTEXT commands.

Figure 2-119 shows an OLIST before you press the Enter key on a FINDTEXT (FT) command with a specified text string.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (V) ----- LEVEL CICSTS ----- Row 1 to 13 of 2,000
Command ==> ft interdependency SCROLL ==> CSR
Hotbar: OPRINT REFRESH CLRVOL FILLVOL UPDATE CUT FLIP VALIDATE
Open list ==> CICSTS (or BLANK for reference list)

Command  Member  Numbr  Data Set Names / Objects  Volume
-----
1  'CICSTS'
2  'CICSTS.CICSADP.ACCTFIL.LOADDB2'
3  'CICSTS.CICSADP.ACCTFILE'
4  'CICSTS.CICSADP.ACCTFILE.DATA'
5  'CICSTS.CICSADP.ACCTFILE.INDEX'
6  'CICSTS.CICSADP.ACCTFILE.UNLOAD'
7  'CICSTS.CICSADP.ACCTNAIX'
8  'CICSTS.CICSADP.ACCTNAIX.DATA'
9  'CICSTS.CICSADP.ACCTNAIX.INDEX'
10 'CICSTS.CICSADP.ACCTNAME'
11 'CICSTS.CICSADP.ACTINUS.LOADDB2'
12 'CICSTS.CICSADP.ACTINUSE'
13 'CICSTS.CICSADP.ACTINUSE.DATA'

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

Figure 2-119 FINDTEXT OLIST command

Figure 2-120 shows an OLIST with the FINDTEXT (FT) command entered without a text string specified. On this panel, you enter the search argument, and you can also limit the number of items processed by changing the Prompt After field.

File	Edit	Find	Display	Populate	Settings	Menu	Util	Test	Help	Exit
- IPT- 2,000										
COMMAND ==> _										
Specify string to search within the OLIST entries:										
Search string ==> INTERDEPENDENCY										
Optionally, specify how many items to process before a prompt panel is displayed:										
Prompt after ==> 500 (Number of items to process before a prompt screen is displayed)										
F1=HELP F2=SPLIT F3=END F4=IPT View F5=RFIND										
F6=RCHANGE F7=UP F8=DOWN F9=SWAP F10=LEFT										
7 'CICSTS.CICSADP.ACCTNAIX'										
8 'CICSTS.CICSADP.ACCTNAIX.DATA'										
9 'CICSTS.CICSADP.ACCTNAIX.INDEX'										
10 'CICSTS.CICSADP.ACCTNAME'										
11 'CICSTS.CICSADP.ACTINUS.LOADDB2'										
12 'CICSTS.CICSADP.ACTINUSE'										
13 'CICSTS.CICSADP.ACTINUSE.DATA'										
F1=HELP F2=SPLIT F3=END F4=IPT View F5=RFIND F6=RCHANGE										
F7=UP F8=DOWN F9=SWAP F10=LEFT F11=RIGHT F12=RETRIEVE										

Figure 2-120 Prompt for number of items to scan with FINDTEXT

Figure 2-121 shows the first member in the first data set that contained the FINDTEXT (FT) string, which does not mean that there are no other members in this data set that contain the FT string, and after you press the RFIND key, the find advances to the next data set in the OLIST.

File	Edit	Find	Display	Populate	Settings	Menu	Util	Test	Help	Exit
- IPT- OLIST (V) ----- LEVEL CICSTS ----- Row 231 to 243 of 2,000										
Command ==> SCROLL ==> CSR										
Hotbar: OPRINT REFRESH CLRVOL FILLVOL UPDATE CUT FLIP VALIDATE										
Open list ==> CICSTS (or BLANK for reference list)										
Command	Member	Numb	Data Set Names / Objects							Volume
-TXT FND-	CIUJ23CR	231	'CICSTS.CICSIA.CIU130.JCL'							DMPU13
		232	'CICSTS.CICSIA.CIU130.SCIUDAT2'							
		233	'CICSTS.CICSIA.CIU130.SCIUDAT3'							
		234	'CICSTS.CICSPA.OMEGAMON.SMFDATA'							
		235	'CICSTS.CICSPA.SMF110.MQ'							
		236	'CICSTS.CICSPA.SMF110.SAMPLE1'							
		237	'CICSTS.CICSPA.SMF110.SAMPLE2'							
		238	'CICSTS.CICSPA.STLABC4.D020124C.MANX'							
		239	'CICSTS.CICSPA.STLABC4.D020124D.MANY'							
		240	'CICSTS.CICSTIVX.EZACACHE'							
		241	'CICSTS.CICSTIVX.EZACACHE.DATA'							
Found in CIUJ23CR --> SEG(CIUCIN1C),COMP(INTERDEPENDENCY),PROD(%PRODUCT										
F7=UP F8=DOWN F9=SWAP F10=LEFT F11=RIGHT F12=RETRIEVE										

Figure 2-121 Results of FINDTEXT OLIST command

A RFIND (PF5) command skips the rest of the members (in that library) and resumes searching from the following datasets in the OLIST. You might need to enter the RFIND (PF5) command until you reach the end of the OLIST.

Figure 2-122 shows an OLIST where the FINDTEXT (FT) search is finished.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (V) ----- LEVEL CICSTS ----- Row 230 from 2000
Command ==> SCROLL ==> CSR
Hotbar: OPRINT REFRESH CLRVOL FILLVOL UPDATE CUT FLIP VALIDATE
Open list ==> CICSTS (or BLANK for reference list)

Command Member Numbr Data Set Names / Objects *EXCLUDE* Volume
-----
-TXT FND- CIUJ23CR 230 'CICSTS.CICSDM.TABSRC' DMPU03
231 'CICSTS.CICSLA.CIU130.JCL' DMPU13
232 'CICSTS.CICSLA.CIU130.SCIUDAT2' DMPU14
262 'CICSTS.CICSTIV3.EZACONFG.DATA'
263 'CICSTS.CICSTIV3.EZACONFG.INDEX'
-TXT FND- CIUJCLRP 264 'CICSTS.CICSTOOL.JCL' DMPU14
265 'CICSTS.CICSTOR1.EZACACHE'
266 'CICSTS.CICSTOR1.EZACACHE.DATA'
267 'CICSTS.CICSTOR1.EZACACHE.INDEX'
268 'CICSTS.CICSTOR1.EZACONFG'
269 'CICSTS.CICSTOR1.EZACONFG.DATA'
270 'CICSTS.CICSTOR1.EZACONFG.INDEX'
271 'CICSTS.CICSVT.DRIVERS'

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

```

Figure 2-122 Completion of OLIST search

You can see all of the line commands and main commands that are available by pressing the F1 key and following the prompts. When you select the member that contains the data string, the FIND command is displayed on the command line.

Figure 2-123 shows a browse command on the first selected member with the FINDTEXT string.

```

File Edit Edit Settings Menu Utilities Compilers Test Help
-----
-IPT- VIEW CICSTS.CICSLA.CIU130.JCL(CIUJ23CR) - 01.03 Columns 00001 00072
Command ==> FIND INTERDEPENDENCY Scroll ==> CSR
***** ***** Top of Data *****
000001 //CIUJ23CR JOB USER=STANNA,MSGCLASS=H,NOTIFY=STANNA,CLASS=A,REGION=0M
000002 //*****
000003 //*
000004 //* JCL NAME = CIUJ23CR
000005 //*
000006 //* DESCRIPTIVE NAME = IBM CICS INTERDEPENDENCIES UTILITY
000007 //* SAMPLE JCL FOR CIU RESOURCE DEFINITIONS
000008 //* (CICS/TS 2.3)
000009 //*
000010 //* CHANGES TO BE MADE
000011 //* PLEASE CONSULT WITH YOUR DB2 ADMINISTRATOR
000012 //*
000013 //* 1) CHANGE THE JOB CARD TO SUIT YOUR SYSTEM CONVENTIONS
000014 //* 2) CHANGE THE FOLLOWING PARAMETERS:-
000015 //*
000016 //* _cicshlq_ THE HLQ FOR THE CICS LOAD LIBRARY
000017 //* _csdhlq_ THE HLQ FOR THE DFHCSO DATASET
F1=Help F2=Split F3=Exit F4=IPT View F5=Rfind F6=Rchange
F7=Up F8=Down F9=Swap F10=Left F11=Right F12=Retrieve

```

Figure 2-123 Browse operation after a FINDTEXT command

You can end or PF3 out of this member, and you are returned to the OLIST. The OLIST still has the other members that matched the FINDTEXT string noted. If you exit this OLIST completely, when you return these members will still be noted.

**Note:** The RESET M command removes the member names, and the command RESET C removes the comments in the command column.



## Member Selection Lists

In this chapter, we teach you many the time saving features and shortcuts using the ISPF-PT Member Selection List.

## 3.1 MSL functions

Using a Member Selection, you can:

- ▶ Identify specific members for subsequent actions using commands, such as:
  - FILTER to select members using various field headings, including statistics with wildcard characters.
  - GLOBAL Find and Change to locate or change data in some or all members of a PDS or PDSE.
  - EXCLUDE to exclude specific members.
  - FLIP to switch between excluded members and included members.
  - TAG to identify specific members.
- ▶ SORT on various field headings, including statistics.
- ▶ Combine multiple utility functions into a single common Member list.
- ▶ Find text using:
  - Global search capabilities
  - Locate, filter, member lists prior to search
- ▶ Perform global edits.
- ▶ Invoke immediate access to other libraries using library switching.
- ▶ Preview data.
- ▶ Invoke SCLM features for multiple members, such as:
  - Automatic check-out
  - Lock management
  - Builds
  - Promotions
  - Migrate members
- ▶ Issue multiple commands on multiple members using extended patterns.
- ▶ For further information about SCLM and ISPF-PT, see Chapter 6, “SCLM integration with the ISPF Productivity Tool” on page 141.

## 3.2 Using the ISPF-PT Member Selection List Assist

ISPF-PT provides an “Assist” for Object Lists, and an “Assist” for Member Selection Lists (MSL). In this section, we show you how to use the MSL Assist. Enter A to invoke the MSL Assist commands, as shown in Figure 3-1 on page 75.



```
File Display Library Settings Menu Utilities Test Help Exit
-----
-IPT--VIEW L1----- DNET424.PROD.COBOL ----- "A" will display assist
COMMAND ==> a SCROLL ==> CSR
HOTBAR: REFRESH FLIP GLOBAL INFO COMPRESS EXPDIR TAILOR TOTALS
ON VOLUME DMPU30

NAME RENAME LIB VV.MM CREATED CHANGED SIZE INIT MOD USERID
ADSORT 1 01.00 07/08/29 07/08/29 15:18 39 39 0 DNET424
ADSTAT 1 01.00 07/08/29 07/08/29 15:18 95 95 0 DNET424
ASAMDRV 1 01.02 07/08/29 07/08/30 12:41 40 40 0 DNET424
ATCDEMO 1 01.00 07/08/29 07/08/29 15:18 434 434 0 DNET424
ATCDEM2 1 01.00 07/08/29 07/08/29 15:18 130 130 0 DNET424
ATCDEM4 1 01.00 07/08/29 07/08/29 15:18 63 63 0 DNET424
ATCDEM5 1 01.00 07/08/29 07/08/29 15:18 93 93 0 DNET424
CDAT1 1 01.00 07/08/29 07/08/29 15:18 306 306 0 DNET424
CDAT2 1 01.00 07/08/29 07/08/29 15:18 117 117 0 DNET424
CDAT3 1 01.00 07/08/29 07/08/29 15:18 118 118 0 DNET424
COBISTUB 1 01.00 07/08/29 07/08/29 15:18 167 167 0 DNET424
COBTIMS 1 01.00 07/08/29 07/08/29 15:18 21 21 0 DNET424
COBVSAM 1 01.00 07/08/29 07/08/29 15:18 189 189 0 DNET424
DTDEMO 1 01.00 07/08/29 07/08/29 15:18 47 47 0 DNET424
IMSSTUB 1 01.00 07/08/29 07/08/29 15:18 39 39 0 DNET424
IMSSTUB2 1 01.00 07/08/29 07/08/29 15:18 47 47 0 DNET424
SAMI11 1 01.00 07/08/29 07/08/29 15:18 439 439 0 DNET424
```

Figure 3-1 Invoking the Assist panel

As shown in Figure 3-2, we can find more information for a command by entering A command, where *command* represents one of the ISPF-PT MSL Main or Line Commands. In Figure 3-2, we want assistance with the DEFault command.

File Display Library Settings Menu Utilities Test Help Exit

-----

-IPT--VIEW L1----- DNET424.PROD.COBOL ----- -ROW 00001 OF 00032

COMMAND ==> a def SCROLL ==> CSR

HOTBAR: REFRESH FLIP GLOBAL INFO COMPRESS EXPDIR TAILOR TOTALS

ON VOLUME DMPU30

NAME	RENAME	LIB	VV.MM	CREATED	CHANGED	SIZE	INIT	MOD	USERID
ADSORT									
ADSTAT									
ASAMDRV									
ATCDEMO									
ATCDEM2									
ATCDEM4									
ATCDEM5									
CDAT1									
CDAT2									
CDAT3									
COBISTUB									
COBTIMS									
COBVSAM									
DTDEMO									
IMSSTUB									
IMSSTUB2									
SAMII1									

----- -IPT- MSL MAIN COMMANDS -----

Assist	COMPRESS	CONFIRM	DEFault	DSName	EXCLUDE
EXIT	EXPDIR	FILTER	Find	FindText	FLIP
Global	INFO	LIB	LMAP	Locate	PROJECT
QUIT	REFRESH	RELEASE	RESET	RFind	SAVE
SCLM	SCLMPARM	SET	SORT	SSI	STATS
SUBMIT	TAG	TAILOR	TOTALS	TYPE	UNFilter
USAGE	VLF	WHERE	XFER		

----- -IPT- MSL LINE COMMANDS -----

B (browse)	C (copy)	D (Delete)	E (edit)	H (where)
I (ISPEXEC)	J (submit)	K (SCLM)	L (lmap)	M (move)
P (print)	R (rename)	S (select)	T (TSO)	V (view)
W (preview)	X (exclude)	Z (stats)		
= (repeat)	% (CLIST/REXX call)			

Use the SET command to control the assist window.

Figure 3-2 Request Assistance with specific MSL command

Figure 3-3 shows the function, syntax, and examples of the DEFAULT command.

File Display Library Settings Menu Utilities Test Help Exit									
-----									
-IPT--VIEW L1----- DNET424.PROD.COBOL					-----ROW 00001 OF 00032				
COMMAND ==> _					SCROLL ==> CSR				
HOTBAR: REFRESH FLIP GLOBAL INFO COMPRESS EXPDIR TAILOR TOTALS									
ON VOLUME DMPU30									
NAME	RENAME	LIB	VV.MM	CREATED	CHANGED	SIZE	INIT	MOD	USERID
ADSORT									Assist
ADSTAT									
ASAMDRV									
ATCDEMO									
ATCDEM2									
ATCDEM4									
ATCDEM5									
CDAT1									
CDAT2									
CDAT3									
COBISTUB									
COBTIMS									
COBVSAM									
DTDEMO									
IMSSTUB									
IMSSTUB2									
SAMII1									

Function: Change the default action of the S (select) command.

Syntax: Default <action> <process>

where: action is: B (browse), E (edit), or V (view)

process is: Ispf , WS , F indicating that browse/edit/view will be performed by ISPF, by the VSAM editor, or on the workstation.

Examples: DEF E - Change the default to EDIT

DEF V WS -Defaults to view (on the PC)

Notes: - This command is much faster than the jump function (=n) to switch to browse/edit/view.

- Regardless of the current defaults, you can override them with the E, B or V commands.

- Place the cursor on the current default tag on the MSL screen and press enter to change.

Figure 3-3 Assist with the DEFAULT command

## 3.3 ISPF-PT Search commands

ISPF-PT provides two methods to locate data. You can use the FindText or FT command as an Object List command or as a Member Selection List command. The FINDTEXT command will:

- ▶ Locate the first occurrence of data
- ▶ Search for multiple files from an OLIST
  - Search PDS and PDSE libraries
  - Search Sequential files
  - Ignore VSAM files and DB2 tables

The Global Find command:

- ▶ Is used only with Member Selection Lists
- ▶ Can be used to find *all* occurrences of Text for a PDS or PDSE
- ▶ Can perform case-sensitive searches
- ▶ Can Link, Find, or Change commands
- ▶ May increase search overhead

In the sections that follow, we describe the MSL FindText and Global Find and Change command.

## 3.4 Global Find and Global Change command

The Global command initiates global editing of the members in the displayed member list. You can locate (FIND) members by context and perform Global Find operations. Using the Global Find command, you can locate members, filter the Member List for only members containing specific text, and display found text along the member names. You can also change members by context and perform global change operations.

### 3.4.1 Changing the Global ISPF-PT defaults for your session

You can display or modify any of your current Global settings on your ISPF-PT profile. Type in the command ISET, and choose option **g** from the main command line, as shown in Figure 3-4.

```
-IPT- -----Setting IBMIPT Defaults-----
COMMAND ==> g
Select options by number, name, with cursor selection, or with line commands:
IBMIPT is running under ISPF version 5.8

- A - ALL          - Select all the below displayed options
- M - MSL          - Member Selection List options
- P - OLIST        - Object list options
- G - GLOBAL       - Global edit and Findtext options
- R - PRINT        - Print options
- D - DSLIST       - DSLIST options
- T - TSO          - TSO shell options
- E - EDIT         - Edit, Browse and View options
- I - INTERFACE    - Specify user interface options
- N - DIAGNOSE     - Diagnose ISPF errors
- L - LIBRARY      - Persistent table library options

Make your selection and press the ENTER key or press the END key to exit

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

Figure 3-4 Results of the ISET command

The results of the option **g** command are shown in Figure 3-5.

```
-IPT- -----GLOBAL/FINDTEXT commands options-----
COMMAND ==>

STOP AFTER  ==> 40  (Number of members to process successfully)
PROMPT AFTER ==> 9920 (Number of members to process before prompt is issued)

Specify Y (Yes) or N (No) for the following options:

AUTOMATIC  ==> Y (Process without editing successful members?)
LINK       ==> Y (Process each command only if previous command succeeds?)
PRINT      ==> N (Generate listing of each member changed and saved?)
EXCLUDE    ==> N (Exclude failing members from selection list?)

Press ENTER for options menu, END to exit, CANCEL for installation defaults.

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

Figure 3-5 Results of choosing option **G** from the ISPF-PT default menu

#### The Global Edit Command settings parameters

In this section, we provide the Global Edit Command settings and parameters:

► STOP AFTER

Lets you limit the number of members that are processed successfully. There will be times when you might need to process just a few members, for example if you are looking for a

member that contains the program IEBCOPY, instead of searching your entire MSL you can set this number to a low value that stops the process after it finds the first occurrences or the first 10 occurrences. The search argument could be in a member multiple times, but it is only counted as found one time per member. Setting this number to the highest amount ensures that the entire MSL is processed without any additional keystrokes from you.

► **PROMPT AFTER**

Lets you decide how many members will be processed before you must press Enter again. The default is 50, and the maximum number is 9999. Setting these numbers to the highest amount allows your Find or Change to run uninterrupted.

► **AUTOMATIC==>Y/N, (Process without editing successful members?)**

If this is set to Y, based on the find or change command you entered and based on the STOP and PROMPT numbers, it gives you a list of the members where it found or changed your global argument.

► **LINK==>Y/N, (Process each command only if previous command succeeds)**

► **PRINT==>Y/N, (Generate listing of each member changed and saved)**

If this is a Y, after making a CHANGE, a panel prompt is presented asking you to print the changed member. See Figure 3-7 on page 80, "Result of Global Change with Print=Y".

► **EXCLUDE==> Y/N, (Exclude failing members from selection list?)**

If this is set to Y, it excludes all the members in your MSL that did not contain the FIND or CHANGE argument. If this is set to N, it shows all of the members in your MSL, even if they did not contain the FIND or CHANGE argument.

You can also change the settings for Global commands using the MSL **G** or **Globa1** command, as shown in Figure 3-6, which depicts the Global EDIT command panel.

```

-IPT- ----- GLOBAL EDIT COMMANDS -----
COMMAND ==>                                SCROLL ==> CSR

STOP AFTER   ==> 9999 (Number of members to process successfully)
PROMPT AFTER ==> 200  (Number of members to process before prompt is issued)
AUTOMATIC    ==> Y   (Process without editing successful members?)
LINK         ==> Y   (Process each command only if previous command succeeds?)
PRINT        ==> N   (Generate listing of each member changed and saved?)
EXCLUDE      ==> N   (Exclude failing members from selection list?)
Specify below the ISPF EDIT commands or macros to be executed (one per line).
Press END to process the global commands, or enter CANCEL to cancel.
-----
***** ***** Top of Data *****
*****
*****
*****
*****
*****
***** Bottom of Data *****
*****

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

```

Figure 3-6 Changing Global Settings for a Specific Search

## Global Find and Change command invocation methods

You can invoke the Global FIND and CHANGE command by:

**Option 1:** A primary command from the MSL, as shown in the following examples:

```
gl find JIM
global FIND 'James'
g change abc def all
```

**Option 2:** Using the Global Edit Command panel, which is shown in Figure 3-6 on page 78:

```
G
Global
```

When you use a Global FIND/CHANGE with parameters from the MSL primary command, which we described in Option 1, the ISPF Productivity Tool performs the Global FIND/Change operation. As a result:

- ▶ The search requires less time, CPU, and I/O.
- ▶ Searches and changes are never case sensitive:
  - A Global Find command locates the data, regardless of the case that is entered on the command line or the text within the members.
  - The target parameter of the Global Change command always results in upper case.

When you use the Global Edit Command panel, which we described in Option 2, ISPF functions are used to perform the Find and Change. As a result:

- ▶ Find and Change commands can be case sensitive:
  - You can use FIND C'Sample' to locate text in members that have the word "Sample". The "C" prefix ensures that the case matches during the search operation.
  - A command, such as Change c'SAMPLE' c'Sample' all, changes all syntax of SAMPLE to Sample.
  - Commands can be *linked* together.
- ▶ Find and Change commands may take longer to complete:
  - The Global command, used in Figure 3-7 on page 80, invokes the ISPF environment to perform the command, which requires more overhead.

### **Examples using the Global Find and Change commands**

In this section, we provide examples using the Global Find and Change commands. Figure 3-7 on page 80 shows the results of a Global Find command G F IEBCOPY with the AUTOMATIC set as Y and the EXCLUDE set as Y.

```

File Display Library Settings Menu Utilities Test Help Exit
-----
-IPT--EDIT L1----- NW.TJM.TESTING.IPT59 -----ROW 00001 OF 00587
COMMAND ==>
HOTBAR?
*EXCLUDE*          59 HIDDEN          69 PROCESSED
NAME  RENAME  LIB VV.MM CREATED      CHANGED      SIZE INIT  MOD USERID
ABENDAID -G:OK Found: IEBCOPY 01 48 85/09/12 07/08/22 13:35 15 10
ANITA -G:OK Found: //STPX1 EXEC PGM=IEBCOPY
BKUPPDS -G:OK Found: //STPX1 EXEC PGM=IEBCOPY
BLBCTM2 -G:OK Found: //STP2 EXEC PGM=IEBCOPY,REGION=4M
BLDCTM1 -G:OK Found: //STP2 EXEC PGM=IEBCOPY,REGION=4M
CICSALCM -G:OK Found: //STPD EXEC PGM=IEBCOPY,REGION=4M
CICSALC1 -G:OK Found: //STPD EXEC PGM=IEBCOPY,REGION=4M
CICSALC4 -G:OK Found: //STPH EXEC PGM=IEBCOPY,REGION=4M
CICSJPD5 -G:OK Found: //STP1 EXEC PGM=IEBCOPY,REGION=4M
CICSTS13 -G:OK Found: //STPH EXEC PGM=IEBCOPY,REGION=4M
CIMSNUMS 1 01.03 00/04/27 02/04/17 10:37 11 7 0 IMOKP
COBBATCH 1 01.00 98/08/18 98/08/18 13:25 28 28 0 XIMHP
COBCICS 1 01.01 98/08/18 01/12/18 08:20 53 53 0 IMOKP
COBEXAMP 1 01.00 06/02/21 06/02/21 07:59 99 99 0 NWTJM
COBMVS 1 01.23 04/04/27 05/12/07 09:24 18 160 0 NWTJM
COBMVS2 1 01.24 04/04/27 07/04/03 10:39 18 160 0 NWTJM
. . . . .

```

Figure 3-7 Results of a Global Find with Automatic=Y and Exclude=Y

Figure 3-7 shows that it processed 67 members before it found the text IEBCOPY in 10 members. The STOP AFTER parameter is set at 10.

If AUTOMATIC is set to N, the **find** or **change** command stops each member where your global argument is found. To advance to the next member that meets the global argument, you must end out of the member.

Figure 3-8 shows the first results of the Global Find command G F IEBCOPY with the AUTOMATIC set as N and the EXCLUDE set as N.

```

-IPT- GLOBAL EDIT L1-- NW.TJM.TESTING.IPT59(ABENDAID) - 0 COLUMNS 00001 00072
COMMAND ==>
SCROLL ==> CSR

NOTE: QUIT terminates GLOBAL processing of the members not yet processed.
-----
***** Top of Data *****
=NOTE= FIND IEBCOPY
=NOTE= 1 FOUND ON 1 LINE(S)
-----
355 Line(s) not Displayed
000356 IEBCOPY 01 48 85/09/12 07/08/22 13:35 15 10 0 NWTJM
-----
1086 Line(s) not Displayed
***** Bottom of Data *****

```

Figure 3-8 Results of the Global Find with Automatic=N and Exclude=N

Setting the LINK field to Y allows you to link multiply find or change commands when you are in a MSL. To add these commands or macros, go to your MSL, and enter the command GLOBAL or G without operands. On this panel, you can specify GLOBAL EDIT options and any number of ISPF EDIT commands using EDIT itself to create and edit the commands.

If this field is N, it processes all of the augments, regardless if the previous argument is successful or not.

Figure 3-9 shows an example of two FIND arguments and one CHANGE argument.

```

-IPT- ----- GLOBAL EDIT COMMANDS -----
COMMAND ==>                                SCROLL ==> CSR

STOP AFTER  ==> 10  (Number of members to process successfully)
PROMPT AFTER ==> 50  (Number of members to process before prompt is issued)
AUTOMATIC   ==> Y   (Process without editing successful members?)
LINK        ==> Y   (Process each command only if previous command succeeds?)
PRINT       ==> N   (Generate listing of each member changed and saved?)
EXCLUDE     ==> Y   (Exclude failing members from selection list?)
Specify below the ISPF EDIT commands or macros to be executed (one per line).
Press END to process the global commands, or enter CANCEL to cancel.
-----
***** Top of Data *****
000001 FIND ALL IEBCOPY
000002 FIND ALL CICS
000003 CHANGE ALL CICS KICS
***** Bottom of Data *****

```

Figure 3-9 Global Find example using the Global Edit commands panel

If you do not use the word ALL, and the member contains the **find** or **change** argument more than once, ISPF-PT only changes the first reference it finds.

**Note:** Pressing END or PF3 processes this request. To cancel the request, you must type the word CANCEL on the command line.

The Print=Y setting invokes the Print panel shown in Figure 3-10.

```

IQIM389 Global executed with PRINT=Y option. Press ENTER to PRINT, END to exit
COMMAND ==>                                SCROLL ==> PAGE
IQIM389 Global executed with PRINT=Y option. Press ENTER to PRINT, END to exit
-----
Suppress page formatting ==> N (N=No, Y=Yes - members already formatted)
Highlight program elements ==> Y (N=No, Y=Yes-emphasize recognized items)
Print changed lines in bold ==> N (N=No, Y=Yes)
Process mode ==> I (I=Immediate, G=Group, L=LIST data set)

CLASS ==> A (Sysout class)  DESTINATION ==> R9431 (Destination id)
FORMS ==> (Form number)    FCB ==> (FCB name)
COPIES ==> 1 (How many?)   PAGESIZE ==> 60 (Lines per page)
HOLD ==> N (Y=Yes, N=No)   WRITER name ==> (Output WRITER)

Use the UP or DOWN PF keys to scroll, ENTER to print, or END key to cancel.
-----
.  NAME      RENAME    LIB VV.MM  CREATED    CHANGED    SIZE  INIT  MOD USERID
  ABENDRID          1 01.29 98/03/20 07/09/05 11:39 1442  20   0 NWTJM
-----

```

Figure 3-10 Result of Global Change with Print=Y

Figure 3-11 depicts a Global Find with Exclude set to N. In this example, you have the option of paging through your MSL to see all of the members that met the Global Find argument, or you can enter SORT TEXT or SORT NOTE, and the list of displayed members (regardless of EXCLUDE setting) is sorted by the text of the first found row in each member.

```

File  Display  Library  Settings  Menu  Utilities  Test  Help  Exit
-----
-IPT--EDIT L1----- NW.TJM.TESTING.IPT59 -----ROW 00001 OF 00646
COMMAND ==>                                     SCROLL ==> CSR
HOTBAR?

NAME      RENAME    LIB  VV.MM  CREATED      CHANGED      SIZE  INIT  MOD  USERID
$FOCLOG   -G:FAIL    1  01.28  01/12/19  05/09/01 09:41    60    62    0   NWTJM
#MYLIB    -G:FAIL    1  01.08  04/11/04  07/08/28 14:26     8     8    0   NWTJM
AAANOTES  -G:FAIL    1  01.99  93/11/17  07/08/29 10:44   153   35   153 NWTJM
ABRIDUT1  -G:FAIL    1  01.48  97/05/06  07/08/29 10:03    30    26    0   NWTJM
ABRIDUT2  -G:FAIL    1  01.22  99/08/19  07/08/28 10:19    25    25    0   NWTJM
ABRIDUT3  -G:FAIL    1  01.21  99/08/20  07/08/29 10:44    25    25    0   NWTJM
ABENDRID  -G:OK      Found: IEBCOPY 01 48 85/09/12 07/08/22 13:35 15 10
ACCOUNT   -G:FAIL    1  01.11  05/03/29  07/08/22 08:11    14     1   14  NWTJMT
ACF$TSOR  -G:FAIL    1  01.07  07/07/13  07/08/29 10:44   170   168    0   NWTJM
ACF2RPT   -G:FAIL    1  01.02  05/01/31  05/02/01 17:15    11     9    0   NWTJM
ACF2TSOR  -G:FAIL    1  01.09  07/07/13  07/08/29 10:44   171   168    0   NWTJM
ADDISPFD  -G:FAIL    1  01.48  99/04/05  07/08/28 10:19    11   122    5   NWTJM
ADDPTRGN  -G:FAIL    1  01.99  96/05/30  07/08/29 10:44    48    25   44  NWTJM
ADDPTRUI  -G:FAIL    1  01.16  97/11/05  07/08/29 10:44    48    27   32  NWTJM
ADDUSER   -G:FAIL    1  01.53  98/06/15  02/09/11 13:30    94   115   57  IMOKP
ADDVRPRT  -G:FAIL    1  01.02  97/12/03  07/08/29 10:44    35    25   18  NWTJM

```

Figure 3-11 Results of the Global Find with Exclude=N

Figure 3-12 shows the results of the same Global Find statement with the Exclude set to Y. Figure 3-12 shows that it processed 646 members, of these there are 564 hidden members. With Exclude=Y, ISPF-PT displays only the members that contained the search argument.

```

File Display Library Settings Menu Utilities Test Help Exit
-----
-IPT--EDIT L1----- NW.TJM.TESTING.IPT59 -----ROW 00001 OF 00082
COMMAND ==> SCROLL ==> CSR
HOTBAR? █

*EXCLUDE* 564 HIDDEN 646 PROCESSED

NAME RENAME LIB VV.MM CREATED CHANGED SIZE INIT MOD USERID
ABENDRID -G:OK Found: IEBCOPY 01 48 85/09/12 07/08/22 13:35 15 10
ANITA -G:OK Found: //STPX1 EXEC PGM=IEBCOPY
BKUPPDS -G:OK Found: //STPX1 EXEC PGM=IEBCOPY
BLBCTM2 -G:OK Found: //STP2 EXEC PGM=IEBCOPY,REGION=4M
BLDCTM1 -G:OK Found: //STP2 EXEC PGM=IEBCOPY,REGION=4M
CICSALCM -G:OK Found: //STPD EXEC PGM=IEBCOPY,REGION=4M
CICSALC1 -G:OK Found: //STPD EXEC PGM=IEBCOPY,REGION=4M
CICSALC4 -G:OK Found: //STPH EXEC PGM=IEBCOPY,REGION=4M
CICSJPDS -G:OK Found: //STP1 EXEC PGM=IEBCOPY,REGION=4M
CICSTS13 -G:OK Found: //STPH EXEC PGM=IEBCOPY,REGION=4M
COPYAUTH -G:OK Found: //STEP01 EXEC PGM=IEBCOPY
COPYDCU -G:OK Found: //STEP01 EXEC PGM=IEBCOPY
COPYEXCI -G:OK Found: //STEP01 EXEC PGM=IEBCOPY
COPYGET -G:OK Found: //STEP01 EXEC PGM=IEBCOPY
COPYLINK -G:OK Found: //STEP01 EXEC PGM=IEBCOPY
COPYLOAD -G:OK Found: //STEP01 EXEC PGM=IEBCOPY

```

Figure 3-12 Result of Global Find with Exclude=Y



The FLIP command displays hidden members. TYPE Flip on the command line, and press Enter. You can FLIP back and forth as needed, as shown in Figure 3-13.

File Display Library Settings Menu Utilities Test Help Exit

-IPT--EDIT L1----- NW.TJM.TESTING.IPT59 -----ROW 00001 OF 00564

COMMAND ==> █ SCROLL ==> CSR

HOTBAR?

\*EXCLUDE\*

82 HIDDEN

NAME	RENAME	LIB	VV.MM	CREATED	CHANGED	SIZE	INIT	MOD	USERID
\$FOCLOG	-G:FAIL	1	01.28	01/12/19	05/09/01 09:41	60	62	0	NWTJM
#MYLIB	-G:FAIL	1	01.08	04/11/04	07/08/28 14:26	8	8	0	NWTJM
AARNOTES	-G:FAIL	1	01.99	93/11/17	07/08/29 10:44	153	35	153	NWTJM
ABAIOUT1	-G:FAIL	1	01.48	97/05/06	07/08/29 10:03	30	26	0	NWTJM
ABAIOUT2	-G:FAIL	1	01.22	99/08/19	07/08/28 10:19	25	25	0	NWTJM
ABAIOUT3	-G:FAIL	1	01.21	99/08/20	07/08/29 10:44	25	25	0	NWTJM
ACCOUNT	-G:FAIL	1	01.11	05/03/29	07/08/22 08:11	14	1	14	NWTJMT
ACF\$TSOA	-G:FAIL	1	01.07	07/07/13	07/08/29 10:44	170	168	0	NWTJM
ACF2RPT	-G:FAIL	1	01.02	05/01/31	05/02/01 17:15	11	9	0	NWTJM
ACF2TSOA	-G:FAIL	1	01.09	07/07/13	07/08/29 10:44	171	168	0	NWTJM
ADDISPF	-G:FAIL	1	01.48	99/04/05	07/08/28 10:19	11	122	5	NWTJM
ADDPTRGN	-G:FAIL	1	01.99	96/05/30	07/08/29 10:44	48	25	44	NWTJM
ADDPTRUI	-G:FAIL	1	01.16	97/11/05	07/08/29 10:44	48	27	32	NWTJM
ADDUSER	-G:FAIL	1	01.53	98/06/15	02/09/11 13:30	94	115	57	IMOKP
ADDVRPRT	-G:FAIL	1	01.02	97/12/03	07/08/29 10:44	35	25	18	
ADVANTIS	-G:FAIL	1	01.06	98/07/31	07/08/29 10:44	14	13	0	

Enter

Figure 3-13 Results of the FLIP command

## MSL FindText

The FINDTEXT or FT command locates the first occurrence of the data string that you specified within a Member Selection List.

Syntax

FT text\_string

Where *text\_string* is the string to be found. If it contains blanks or special characters, enclose the text-string in quotation marks. Figure 3-14 shows the results of a FINDTEXT command.

File Display Library Settings Menu Utilities Test Help Exit

-IPT--BROWSE L1--- NW.TJM.TESTING.IPT59 -----ROW 00003 OF 00646

COMMAND ==> ft jes SCROLL ==> CSR

HOTBAR?

NAME	RENAME	LIB	VV.MM	CREATED	CHANGED	SIZE	INIT	MOD	USERID
AARNOTES	-TXT FND	Found:			INACT MMJESB,F				
ABAIOUT1		1	01.48	97/05/06	07/08/29 10:03	30	26	0	NWTJM
ABAIOUT2		1	01.22	99/08/19	07/08/28 10:19	25	25	0	NWTJM
ABAIOUT3		1	01.21	99/08/20	07/08/29 10:44	25	25	0	NWTJM
ABENDAID		1	01.30	98/03/20	07/09/05 14:57	1442	20	0	NWTJM
ACCOUNT		1	01.11	05/03/29	07/08/22 08:11	14	1	14	NWTJMT
ACF\$TSOA		1	01.07	07/07/13	07/08/29 10:44	170	168	0	NWTJM
ACF2RPT		1	01.02	05/01/31	05/02/01 17:15	11	9	0	NWTJM
ACF2TSOA		1	01.09	07/07/13	07/08/29 10:44	171	168	0	NWTJM
ADDISPF		1	01.48	99/04/05	07/08/28 10:19	11	122	5	NWTJM
ADDPTRGN		1	01.99	96/05/30	07/08/29 10:44	48	25	44	NWTJM
ADDPTRUI		1	01.16	97/11/05	07/08/29 10:44	48	27	32	NWTJM
ADDUSER		1	01.53	98/06/15	02/09/11 13:30	94	115	57	IMOKP
ADDVRPRT		1	01.02	97/12/03	07/08/29 10:44	35	25	18	NWTJM
ADVANTIS		1	01.06	98/07/31	07/08/29 10:44	14	13	0	NWTJM
ALAISSCO		1	01.14	00/11/28	07/08/29 10:45	27	12	0	NWTJM
ALAISSDD		1	01.13	00/11/28	07/08/29 10:45	66	12	0	NWTJM

Figure 3-14 Results of FindText command

ISPF-PT provides the feedback message “TXT FND” in the rename column and part of the line containing the text.

You can limit the number of members that you want to process by setting the PROMPT AFTER option of the GLOBAL processing options. You can set these options using the ISET command or by typing a G when you are in a MSL, which we described in 3.4.1, “Changing the Global ISPF-PT defaults for your session” on page 77.

After you enter the FINDTEXT string, you can use the RFIND (repeat find) command to continue to advance to the next occurrence. You can press RFIND until you have looked at the entire MSL or just until you find the text in a few of the members. If you enter SORT TEXT or SORT NOTE, the members that contained the FINDTEXT string are sorted to the top of the member list.

When you select the member that contains the data string, the find is displayed on the command line. Figure 3-15 shows a selected member with the FINDTEXT string on the command line.

```

Menu Utilities Compilers Help
-----
-IPT- BROWSE  NW.TJM.TESTING.IPT59(AAANOTES) - 01.9 Line 00000000 Col 001 080
Command ==> FIND      JES                      Scroll ==> CSR
***** Top of Data *****
MESSAGE ON THE TSO SIGNON PANEL -                                00072099
SE LIST                      (SHOWS MESSAGE(S))                  00073099
SE 1,DELETE                  (DELETE 1ST MESSAGE)                00074099
SE 2,DELETE                  (DELETE 2ND MESSAGE)                00075099
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
IN THE COLS TYPE .X .Y IN THE RANGE OF CHANGES                  00077099
THAN C //T //X .X .Y ALL                                         00077199
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
VTS -                                                              00077299
D SMS (SHOWS THE CDS DATASET NAME)                               00077999
V SMS,LIB(LIBNATV1),ONLINE   (DRIVES 500-507)                   00078099
V SMS,LIB(LIBVTS1),ONLINE   (DRIVES 5E0-5FF)                   00078199
TAPE -                                                            00078299
D U,TAPE,ALLOC (SHOWS ALLOCATED DRIVES)                         00078399
D U,TAPE,ONLINE (SHOWS ONLINE DRIVES)                           00078499
V XXX,OFFLINE (VARY DRIVE OFFLINE)                             00078599
V XXX,ONLINE (VARY DRIVE ONLINE)                               00078699
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
TCPIP                                                             00078799

```

Figure 3-15 Selected member that contains the FindText string on the command line

**Note:** FindText accepts scope commands, such as FIRST, LAST, PREV, NEXT, and ALL. During the FINDTEXT member search, the scope commands are *not* used. However, when you select a member, the command request, such as “FIND LAST EXAMPLE”, displays on the Command Line.

Figure 3-16 on page 85 shows a selected member with the ‘FINDTEXT string FIRST’.

*Figure 3-16 Selected member containing text from the “FT JES FIRST” command*

[illegible]

Figure 3-17 PDS member located with the “FIND JES FIRST” command

Figure 3-18 on page 86 shows how to find data with imbedded blanks.

File Display Library Settings Menu Utilities Test Help Exit									
-----									
-IPT--BROWSE L1---- NW TJM TESTING IPT59 -----ROW 00001 OF 00646									
COMMAND ==> ft 'PIC X(01) VALUE LOW' SCROLL ==> CSR									
HOTBAR?									
NAME	RENAME	LIB	VV.MM	CREATED	CHANGED	SIZE	INIT	MOD	USERID
\$FOCLOG		1	01.28	01/12/19	05/09/01 09:41	60	62	0	NWTJM
#MYLIB		1	01.08	04/11/04	07/08/28 14:26	8	8	0	NWTJM
AAANOTES		1	01.99	93/11/17	07/09/06 11:14	153	35	153	NWTJM
ABAIDUT1		1	01.48	97/05/06	07/08/29 10:03	30	26	0	NWTJM
ABAIDUT2		1	01.22	99/08/19	07/08/28 10:19	25	25	0	NWTJM
ABAIDUT3		1	01.21	99/08/20	07/08/29 10:44	25	25	0	NWTJM
ABENDRID		1	01.30	98/03/20	07/09/05 14:57	1442	20	0	NWTJM
ACCOUNT		1	01.11	05/03/29	07/08/22 08:11	14	1	14	NWTJMT
ACF\$TSOA		1	01.07	07/07/13	07/08/29 10:44	170	168	0	NWTJM
ACF2RPT		1	01.02	05/01/31	05/02/01 17:15	11	9	0	NWTJM
ACF2TSOA		1	01.09	07/07/13	07/08/29 10:44	171	168	0	NWTJM
ADDISPF		1	01.48	99/04/05	07/08/28 10:19	11	122	5	NWTJM
ADDPTRGN		1	01.99	96/05/30	07/08/29 10:44	48	25	44	NWTJM
ADDPTRUI		1	01.16	97/11/05	07/08/29 10:44	48	27	32	NWTJM
ADDUSER		1	01.53	98/06/15	02/09/11 13:30	94	115	57	IMOKP
ADDVRPT		1	01.02	97/12/03	07/08/29 10:44	35	25	18	NWTJM
ADVANTIS		1	01.06	98/07/31	07/08/29 10:44	14	13	0	NWTJM

Figure 3-18 FT 'PIC X(01) LOW VALUE' command

## 3.5 Partitioned Data Set Extended

Most z/OS mainframes now support Partitioned Data Set Extended (PDSE). The following Web site provides additional information about PDSE libraries.

[http://www.ibm.com/systems/storage/software/sms/whatis\\_sms/pdse.html](http://www.ibm.com/systems/storage/software/sms/whatis_sms/pdse.html)

A PDSE provides many advantages over a PDS, such as:

- ▶ Searching directories is more efficient because they are indexed and can be cached
- ▶ You can create multiple members at the same time
- ▶ Sharing occurs at member and data set levels
- ▶ PDSEs can expand to up to 123 extents
- ▶ Program objects stored in PDSEs allow the program management loader to load only part of a program before passing control
- ▶ PDSE program libraries allow more control information to be stored in the directory than partitioned data sets do
- ▶ PDSE program libraries offer superior buffer management and caching capability
- ▶ Alias entry management for PDSE libraries is better than that offered by PDS libraries
- ▶ PDSEs can be shared by all the processors in a multisystem complex

### 3.5.1 Defining PDSEs

If your installation is installed DFSMS/MVS™ with SMS active, you can allocate PDSEs in SMS storage groups. The following parameters are required:

- ▶ Directory space greater than zero or DSORG=PO
- ▶ DSNTYPE=LIBRARY
- ▶ They can be specified:
  - In a data class definition

- On a JCL DD statement
- Using the LIKE keyword
- On a TSO/E ALLOCATE command
- On the DYNALLOC macro
- Through installation default

## When should I use PDSs rather than PDSEs?

Continue to use PDSs rather than PDSEs when:

- ▶ You need a checkpoint data set. PDSEs cannot be used as checkpoint data sets.
- ▶ You are defining a program library that is used during initial program load (IPL), such as SYS1.NUCLEUS, SYS1.LPALIB, or SYS1.SVCLIB
- ▶ You need to ship the data set to or share the data set with a system that does not support PDSEs
- ▶ An application that uses the data set does not support PDSEs, for example, it uses EXCP, EXCPVR or XDAP
- ▶ An application that uses the data set is dependent on processing short blocks or SAM record null segments

## ISPF-PT Support PDS libraries – Automated Compress / Directory Expansion

Even though most z/OS mainframes support PDSE libraries, many people still use PDS libraries. If the PDS you are using requires maintenance, ISPF-PT can guide you through the steps that are required to correct the problem.

We save a change to the @FILECHK member, as shown in Figure 3-19.

File	Edit	Edit_Settings	Menu	Utilities	Compilers	Test	Help
-IPT- EDIT DNET424.ADLABNEW.SOURCE(@FILECHK) - 01.00 Columns 00001 00072							
Command ==> save Scroll ==> CSR							
***** Top of Data *****							
==MSG> -CAUTION- Profile is set to STATS ON. Statistics did not exist for							
==MSG> this member, but will be generated if data is saved.							
000001	/*	This REXX exec will determine if a file exists	*/				
000002	/*	If the file is found, it returns with RC = 0	*/				
000003	/*	If NOT found, it returns with RC = 8	*/				
000004	/*		*/				
000005	/*	The filename is passed in 2 parts: Prefix and Suffix	*/				
000006	/*		*/				
000007	TRACE	I;					
000008	parse	upper arg FilePrefix " " FileSuffix;					
000009	OutFileName	= " "FilePrefix"."FileSuffix";					
000010	"ALLOC	FI(LABFILE) DA("OutFileName") SHR";					
000011	if	RC = 0 then do;					
000012		"FREE FILE(LABFILE)";					
000013		exit 0;					
000014		end;					
000015		exit 4;					
F1=Help	F2=Split	F3=Exit	F4=IPT View	F5=Rfind	F6=Rchange		
F7=Up	F8=Down	F9=Swap	F10=Left	F11=Right	F12=Retrieve		

Figure 3-19 Saving a PDS member

The DNET424.ADLABNEW.SOURCE PDS has no directory blocks available. The save fails. ISPF-PT intercepts the failure, as shown in Figure 3-20 on page 88.

```

-IPT- EDIT ----- DNET424.ADLABNEW.SOURCE(@FILECHK) on DMPU11 -----
COMMAND ==>

You have issued a SAVE request for the following member: @FILECHK

Your request has failed because there is not enough room in the library
directory.

The SAVE request may be successful if IBMIPT expands the directory.

Expand the directory and then save? ==> Y (Y=yes,N=no)

Specify your choice and press ENTER.

F IQIS038 Please confirm the EXPDIR request before saving the member GE
F7=UP F8=DOWN F9=SWAP F10=LEFT F11=RIGHT F12=RETRIEVE

```

Figure 3-20 ISPF-PT intercepts Directory Stow error

To expand the directory, press Enter to continue. Figure 3-21 is displayed.

```

File Edit Edit_Settings Menu Utilities Compilers Test Help
-IPT- EDIT DNET424.ADLABNEW.SOURCE(@FILECHK) - 01.00 Member @FILECHK Saved
Command ==> Scroll ==> CSR
***** Top of Data *****
==MSG> -CAUTION- Profile is set to STATS ON. Statistics did not exist for
==MSG> this member, but will be generated if data is saved.
000001 /* This REXX exec will determine if a file exists */
000002 /* If the file is found, it returns with RC = 0 */
000003 /* If NOT found, it returns with RC = 8 */
000004 /* */
000005 /* The filename is passed in 2 parts: Prefix and Suffix */
000006 /* */
000007 TRACE I;
000008 parse upper arg FilePrefix " " FileSuffix;
000009 OutFileName = " "FilePrefix"."FileSuffix"";
000010 "ALLOC FI(LABFILE) DA("OutFileName") SHR";
000011 if RC = 0 then do;
000012 "FREE FILE(LABFILE)";
000013 exit 0;
000014 end;
000015 exit 4;
F1=Help F2=Split F3=Exit F4=IPT View F5=Rfind F6=Rchange
F7=Up F8=Down F9=Swap F10=Left F11=Right F12=Retrieve

```

Figure 3-21 Member saved successfully

ISPF-PT expanded the directory and saved the member @FILECHK, which is a significant improvement over ISPF. ISPF requires steps similar to the following:

1. Split the panel.
2. Navigate to ISPF option 3.2.
3. View the DNET424.ADLABNEW.SOURCE library.
4. Exit.
5. Allocate a similar library, with additional directory blocks as DNET424.ADLABNEW.SOURCE.
6. Navigate to ISPF option 3.3.

7. Copy all members from DNET424.ADLABNEW.SOURCE to the new library allocated.
8. Swap to the original library.
9. Use the SAVE *new library*, with a line command C9999 to save the member in the new PDS.
- 10.Exit from option 2.
- 11.Delete the original PDS library.
- 12.Rename the new PDS to the DNET424.ADLABNEW.SOURCE.

In Figure 3-22, we attempt to save the member ASAM1.

```

File Edit Edit_Settings Menu Utilities Compilers Test Help
-IPT- EDIT DNET424.ADLABNEW.SOURCE(ASAM1) - 01.02 Columns 00001 00072
Command ==> save Scroll ==> CSR
***** ***** Top of Data *****
==MSG> -CAUTION- Profile changed to CAPS ON (from CAPS OFF) because the
==MSG> data does not contain any lower case characters.
000001 ASAM1 AMODE 31
000002 ASAM1 CSECT
000003 * *****
000004 * SAMPLE PROGRAM SAMASM1
000005 * AUTHOR: DOUG STOUT
000006 * IBM CORP
000007 *
000008 * A SIMPLE PROGRAM THAT:
000009 * - READS A QSAM FILE
000010 * - WRITES THE FIRST 80 BYTES OF EACH RECORD TO AN OUTPUT FILE
000011 * IN CHARACTER AND HEX
000012 *
000013 * PARAMETERS PASSED FROM CALLING PROGRAM:
000014 * NONE
000015 *
F1=Help F2=Split F3=Exit F4=IPT View F5=Rfind F6=Rchange
F7=Up F8=Down F9=Swap F10=Left F11=Right F12=Retrieve

```

Figure 3-22 Save source member

However, an error message (not shown) indicates that we exceeded the space allocated to the library. Figure 3-23 depicts how ISPF-PT captures the failure.

```

-IPT- EDIT ----- DNET424.ADLABNEW.SOURCE(ASAM1) on DMPU11 -----
COMMAND ==>

You have issued a SAVE request for the following member: ASAM1

Your request has failed because there is not enough room in the library.

The SAVE request may be successful if IBMIPT first compresses the library.

Compress the library and then save? ==> Y (Y=yes,N=no)

Specify your choice and press ENTER.

IQIS035 Please confirm the COMPRESS request before saving the member
F7=UP F8=DOWN F9=SWAP F10=LEFT F11=RIGHT F12=RETRIEVE

```

Figure 3-23 ISPF-PT captures the Save failure

Press Enter to compress the library and to continue. Figure 3-24 shows the results.

```
File Edit Edit_Settings Menu Utilities Compilers Test Help
-IPT- EDIT DNET424.ADLABNEW.SOURCE(ASAM1) - 01.02      Member ASAM1 Saved
Command ==>      Scroll ---> CSR
***** ***** Top of Data *****
==MSG> -CAUTION- Profile changed to CAPS ON (from CAPS OFF) because the
==MSG>      data does not contain any lower case characters.
000001 ASAM1      AMODE 31
000002 ASAM1      CSECT
000003 * *****
000004 *      SAMPLE PROGRAM SAMASM1
000005 *      AUTHOR: DOUG STOUT
000006 *      IBM CORP
000007 *
000008 *      A SIMPLE PROGRAM THAT:
000009 *      - READS A QSAM FILE
000010 *      - WRITES THE FIRST 80 BYTES OF EACH RECORD TO AN OUTPUT FILE
000011 *      IN CHARACTER AND HEX
000012 *
000013 *      PARAMETERS PASSED FROM CALLING PROGRAM:
000014 *      NONE
000015 *
F1=Help      F2=Split      F3=Exit      F4=IPT View      F5=Rfind      F6=Rchange
F7=Up        F8=Down      F9=Swap      F10=Left      F11=Right     F12=Retrieve
```

Figure 3-24 Member saved successfully

After the library was compressed, the save operation completed successfully.

## Using MSL member identification functions

A member selection list is similar to the ISPF member list that is presented when you open a PDS or PDSE. However, an ISPF-PT Member Selection List provides enhancements, which perform complex functions and minimize keystrokes and navigation that are required to perform a task.

You might want to repeat a series of actions, such as Copy, Move, Print, or Delete, on a subset of members of a PDS or PDSE. In this chapter, we demonstrate the power of the Member Identification commands that are available when you use the ISPF-PT MSL. In the next four sections, we show you how to perform the following tasks:

- ▶ Copy only members identified by text data within a member.
- ▶ Copy only members identified by two or more text fields.
- ▶ Identify members, and move them to another PDS.
- ▶ Use ISPF concatenated libraries to manage change.

### ***Copy only members identified by text data within a member***

Suppose your new standards require you to separate the COBOL copybook members from the source COBOL libraries. The following examples provide a quick and easy way to separate copybook members.

For the purposes of this exercise, we assume that all of the members are either valid COBOL programs or valid COBOL copybook members.



In Figure 3-25, the libraries DNET424.PROD.COBOL and DNET424.PROD.COPYLIB are empty. In the library DNET424.PROD.SOURCE, we use the literal PROGRAM-ID to identify the COBOL programs, and we copy these members to the COBOL library. Next, we copy the remaining modules to the COPYLIB library.

Invoke the MSL for the SOURCE library, as shown in Figure 3-25.

File	Edit	Find	Display	Populate	Settings	Menu	Util	Test	Help	Exit
-----										
-IPT- OLIST (B)			----- LEVEL DNET424.PROD -----					Row 1 to 3 of 3		
Command ==>								SCROLL ==> CSR		
Hotbar: OPRINT			REFRESH	CLRVOL	FILLVOL	UTIL	UPDATE	CUT	FLIP	
								*TEMPORARY LIST*		
-----										
Command	Member	Numbr	Data Set Names / Objects							Class
-----										
		1	'DNET424.PROD.COBOL'							PDSE
		2	'DNET424.PROD.COPYLIB'							PDSE
		3	'DNET424.PROD.SOURCE'							PDSE
			----- END OF LIST -----							
-----										
v										
-----										
F1=HELP	F2=SPLIT	F3=END	F4=IPT View	F5=RFIND	F6=RCHANGE					
F7=UP	F8=DOWN	F9=SWAP	F10=LEFT	F11=RIGHT	F12=RETRIEVE					

Figure 3-25 COBOL, COPYLIB, and SOURCE libraries

In Figure 3-26, we issued a GLOBAL FIND, abbreviated GL, to locate the PROGRAM-ID field.

File	Display	Library	Settings	Menu	Utilities	Test	Help	Exit
-----								
-IPT--VIEW L1----- DNET424.PROD.SOURCE -----								
COMMAND ==> gl find program-id 7 72								
HOTBAR: REFRESH FLIP GLOBAL INFO COMPRESS EXPDIR TAILOR TOTALS								
ON VOLUME DMPU21								
NAME	RENAME	LIB	VV.MM	CREATED	CHANGED	SIZE	INIT	MOD USERID
ADSORT		1	01.00	07/08/29	07/08/29 15:18	39	39	0 DNET424
ADSTAT		1	01.00	07/08/29	07/08/29 15:18	95	95	0 DNET424
ASAMDRV		1	01.02	07/08/29	07/08/30 12:41	40	40	0 DNET424
ATCDEM0		1	01.00	07/08/29	07/08/29 15:18	434	434	0 DNET424
ATCDEM2		1	01.00	07/08/29	07/08/29 15:18	130	130	0 DNET424
ATCDEM4		1	01.00	07/08/29	07/08/29 15:18	63	63	0 DNET424
ATCDEM5		1	01.00	07/08/29	07/08/29 15:18	93	93	0 DNET424
CDAT1		1	01.00	07/08/29	07/08/29 15:18	306	306	0 DNET424
CDAT2		1	01.00	07/08/29	07/08/29 15:18	117	117	0 DNET424
CDAT3		1	01.00	07/08/29	07/08/29 15:18	118	118	0 DNET424
CEETEST1		1						
CEETEST2		1						
COBISTUB		1	01.00	07/08/29	07/08/29 15:18	167	167	0 DNET424
COBTIMS		1	01.00	07/08/29	07/08/29 15:18	21	21	0 DNET424
COBVSAM		1	01.00	07/08/29	07/08/29 15:18	189	189	0 DNET424
-----								
F1=HELP	F2=SPLIT	F3=END	F4=IPT View	F5=RFIND	F6=RCHANGE			
F7=UP	F8=DOWN	F9=SWAP	F10=LEFT	F11=RIGHT	F12=RETRIEVE			

Figure 3-26 Global Find PROGRAM-ID

Only the COBOL programs should have the PROGRAM-ID values specified.

In Figure 3-27, we exceeded the PROMPT value of 50.

```

-IPT--GLOBAL EDIT ----DNET424.PROD.SOURCE on DMPU21-----
COMMAND ==>
50    out of 77    members were processed. Member SAMI12 is being processed.

STOP  AFTER ==> 9999 (Number of members to process successfully)
PROMPT AFTER ==> 50 (Number of members to process before a prompt is issued)

Member  Result  Member  Result  Member  Result  Member  Result
RECBUF  -G: FAIL  RECBUF1 -G: FAIL  RECBUF2 -G: FAIL  REPLACE -G: FAIL
REPLACE -G: FAIL  SAMI11  -G: OK   SAMI12  -G: OK   SAMI13
SAMOS1   SAMOS2   SAMOS3   SAM1
SAM1V    SAM2     SAM2V    SAM3
SEGREC   SHIRE    SHIRENP  SHIREN2
SHRLSUB  STREET   STALSUB  SUBLSHR
SUBLSTR  SUBURB   TDM01A   TDM01B
TDM01C   TDM01D   TMAPAPI  TRANRCOB
TRANREC  XSTREET

IQIM388 50 Members processed. Press ENTER to continue or END key to quit

F7=UP      F8=DOWN    F9=SWAP    F10=LEFT   F11=RIGHT  F12=RETRIEVE

```

Figure 3-27 Global Find - PROMPT message issued

Press Enter to continue. The RENAME field depicts the response from the GLOBAL FIND command. -G:OK indicates that the literal PROGRAM-ID was found, and -G:FAIL indicates that the literal was not found, as shown in Figure 3-28.

```

File Display Library Settings Menu Utilities Test Help Exit
-----
-IPT--VIEW L1----- DNET424.PROD.SOURCE -----ROW 00001 OF 00077
COMMAND ==> filter rename -g:ok SCROLL ==> CSR
HOTBAR: REFRESH FLIP GLOBAL INFO COMPRESS EXPDIR TAILOR TOTALS
ON VOLUME DMPU21

NAME  RENAME  LIB  VV.MM  CREATED  CHANGED  SIZE  INIT  MOD  USERID
-----
ADSORT -G: OK   1  01.00  07/08/29 07/08/29 15:18  39   39   0  DNET424
ADSTAT -G: OK   1  01.00  07/08/29 07/08/29 15:18  95   95   0  DNET424
ASAMDRV -G: OK   1  01.02  07/08/29 07/08/30 12:41  40   40   0  DNET424
ATCDEM0 -G: OK   1  01.00  07/08/29 07/08/29 15:18  434  434   0  DNET424
ATCDEM2 -G: OK   1  01.00  07/08/29 07/08/29 15:18  130  130   0  DNET424
ATCDEM4 -G: OK   1  01.00  07/08/29 07/08/29 15:18  63   63   0  DNET424
ATCDEM5 -G: OK   1  01.00  07/08/29 07/08/29 15:18  93   93   0  DNET424
CDAT1   -G: OK   1  01.00  07/08/29 07/08/29 15:18  306  306   0  DNET424
CDAT2   -G: OK   1  01.00  07/08/29 07/08/29 15:18  117  117   0  DNET424
CDAT3   -G: OK   1  01.00  07/08/29 07/08/29 15:18  118  118   0  DNET424
CEETEST1 -G: FAIL  1
CEETEST2 -G: FAIL  1
COBISTUB -G: OK   1  01.00  07/08/29 07/08/29 15:18  167  167   0  DNET424
COBTIMS  -G: OK   1  01.00  07/08/29 07/08/29 15:18  21   21   0  DNET424
COBVSAM  -G: OK   1  01.00  07/08/29 07/08/29 15:18  189  189   0  DNET424

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

```

Figure 3-28 Global Find Results for PROGRAM-ID. FILTER command invoked

The FILTER command reduces the list to only COBOL program members.

**Hint:** You can elect to set the option of the Global options to **Exclude failing members from selection list**. Using this method, the FILTER command is not required. To set the Global option, enter the primary command “GLOBAL”.

The COPY \* command initiates the copy process as shown in Figure 3-29.

File Display Library Settings Menu Utilities Test Help Exit											
-----											
-IPT--VIEW L1----- DNET424.PROD.SOURCE -----ROW 00001 OF 00032											
COMMAND ==> <b>copy *</b> -----SCROLL ==> CSR											
HOTBAR: REFRESH FLIP -----GLOBAL INFO COMPRESS EXPDIR TAILOR TOTALS											
*FILTER* -----45 HIDDEN 77 PROCESSED ON VOLUME DMPU21											
NAME	RENAME	LIB	VV.MM	CREATED	CHANGED	SIZE	INIT	MOD	USERID		
ADSORT	-G:OK	1	01.00	07/08/29	07/08/29 15:18	39	39	0	DNET424		
ADSTAT	-G:OK	1	01.00	07/08/29	07/08/29 15:18	95	95	0	DNET424		
ASAMDRV	-G:OK	1	01.02	07/08/29	07/08/30 12:41	40	40	0	DNET424		
ATCDEMO	-G:OK	1	01.00	07/08/29	07/08/29 15:18	434	434	0	DNET424		
ATCDEM2	-G:OK	1	01.00	07/08/29	07/08/29 15:18	130	130	0	DNET424		
ATCDEM4	-G:OK	1	01.00	07/08/29	07/08/29 15:18	63	63	0	DNET424		
ATCDEM5	-G:OK	1	01.00	07/08/29	07/08/29 15:18	93	93	0	DNET424		
CDAT1	-G:OK	1	01.00	07/08/29	07/08/29 15:18	306	306	0	DNET424		
CDAT2	-G:OK	1	01.00	07/08/29	07/08/29 15:18	117	117	0	DNET424		
CDAT3	-G:OK	1	01.00	07/08/29	07/08/29 15:18	118	118	0	DNET424		
COBISTUB	-G:OK	1	01.00	07/08/29	07/08/29 15:18	167	167	0	DNET424		
COBTIMS	-G:OK	1	01.00	07/08/29	07/08/29 15:18	21	21	0	DNET424		
COBVSAM	-G:OK	1	01.00	07/08/29	07/08/29 15:18	189	189	0	DNET424		
DTDEMO	-G:OK	1	01.00	07/08/29	07/08/29 15:18	47	47	0	DNET424		
IMSSTUB	-G:OK	1	01.00	07/08/29	07/08/29 15:18	39	39	0	DNET424		
-----											
F1=HELP			F2=SPLIT			F3=END			F4=IPT View		
F7=UP			F8=DOWN			F9=SWAP			F10=LEFT		
									F11=RIGHT		
									F12=RETRIEVE		

Figure 3-29 Filtered Members. COPY command initiated

We entered the target library, as shown in Figure 3-30.

```

-IPT----- COPY PANEL ----- (FROM DNET424.PROD.SOURCE) -----
COMMAND ==>
Specify "T0" data set below (32 members will be processed):

PROJECT ==> DNET424
GROUP ==> PROD
TYPE ==> cobol

Or other Partitioned, Sequential Data Set or @H (History List):
DATA SET NAME ==>
VOLUME SERIAL ==> (If not catalogued)
PASSWORD ==> (If password protected)

COPY OPTIONS:
REPLACE like-named library members ==> Y (Y=Yes, N=No, 0=Target is Older)
Disposition for sequential target ==> OLD (OLD or MOD)
Note: If statistics are not available, replace option 0 (older) is
treated as option Y (Yes).

Press ENTER to copy END key to cancel.

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

```

Figure 3-30 Identify target library

You can rename members using this panel. Press Enter to copy identified members to the target library.

You can rename members using Figure 3-31. Press Enter to copy identified members to the target library.

-IPT--VIEW----- COPY PANEL -----
COMMAND ==> \_

You have requested that 32 members be COPIED to DNET424.PROD.COBOL.

Indicate a new member name in the RENAME column, or leave it blank to use the same member name.

Press ENTER to process the list, or the END key to cancel.
(Use the PF keys for scrolling to see the entire list.)

NAME	RENAME	LIB	VV.MM	CREATED	CHANGED	SIZE	INIT	MOD	USERID
ADSORT		1	01.00	07/08/29	07/08/29 15:18	39	39	0	DNET424
ADSTAT		1	01.00	07/08/29	07/08/29 15:18	95	95	0	DNET424
ASAMDRV		1	01.02	07/08/29	07/08/30 12:41	40	40	0	DNET424
ATCDEM0		1	01.00	07/08/29	07/08/29 15:18	434	434	0	DNET424
ATCDEM2		1	01.00	07/08/29	07/08/29 15:18	130	130	0	DNET424
ATCDEM4		1	01.00	07/08/29	07/08/29 15:18	63	63	0	DNET424
ATCDEM5		1	01.00	07/08/29	07/08/29 15:18	93	93	0	DNET424
CDAT1		1	01.00	07/08/29	07/08/29 15:18	306	306	0	DNET424
CDAT2		1	01.00	07/08/29	07/08/29 15:18	117	117	0	DNET424
CDAT3		1	01.00	07/08/29	07/08/29 15:18	118	118	0	DNET424

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

Figure 3-31 Confirmation panel

Use the Hotbar FLIP command to show the copylib members, as shown in Figure 3-32.

File Display Library Settings Menu Utilities Test Help Exit
-IPT--VIEW L1----- DNET424.PROD.SOURCE -----ROW 00001 OF 00032
COMMAND ==> SCROLL ==> CSR
HOTBAR: REFRESH **FLIP** GLOBAL INFO COMPRESS EXPDIR TAILOR TOTALS
\*FILTER\* 45 HIDDEN ON VOLUME DMPU21

NAME	RENAME	LIB	VV.MM	CREATED	CHANGED	SIZE	INIT	MOD	USERID
ADSORT	-COPIED	1	01.00	07/08/29	07/08/29 15:18	39	39	0	DNET424
ADSTAT	-COPIED	1	01.00	07/08/29	07/08/29 15:18	95	95	0	DNET424
ASAMDRV	-COPIED	1	01.02	07/08/29	07/08/30 12:41	40	40	0	DNET424
ATCDEM0	-COPIED	1	01.00	07/08/29	07/08/29 15:18	434	434	0	DNET424
ATCDEM2	-COPIED	1	01.00	07/08/29	07/08/29 15:18	130	130	0	DNET424
ATCDEM4	-COPIED	1	01.00	07/08/29	07/08/29 15:18	63	63	0	DNET424
ATCDEM5	-COPIED	1	01.00	07/08/29	07/08/29 15:18	93	93	0	DNET424
CDAT1	-COPIED	1	01.00	07/08/29	07/08/29 15:18	306	306	0	DNET424
CDAT2	-COPIED	1	01.00	07/08/29	07/08/29 15:18	117	117	0	DNET424
CDAT3	-COPIED	1	01.00	07/08/29	07/08/29 15:18	118	118	0	DNET424
COBISTUB	-COPIED	1	01.00	07/08/29	07/08/29 15:18	167	167	0	DNET424
COBTIMS	-COPIED	1	01.00	07/08/29	07/08/29 15:18	21	21	0	DNET424
COBVSAM	-COPIED	1	01.00	07/08/29	07/08/29 15:18	189	189	0	DNET424
DTDEMO	-COPIED	1	01.00	07/08/29	07/08/29 15:18	47	47	0	DNET424
IMSSTUB	-COPIED	1	01.00	07/08/29	07/08/29 15:18	39	39	0	DNET424

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

Figure 3-32 FLIP command

Initiate the COPY command to copy the remaining members, as shown in Figure 3-33 on page 95.

```

File Display Library Settings Menu Utilities Test Help Exit
-----
-IPT--VIEW L1----- DNET424.PROD.SOURCE -----ROW 00001 OF 00045
COMMAND ==> copy * SCROLL ==> CSR
HOTBAR: REFRESH FLIP GLOBAL INFO COMPRESS EXPDIR TAILOR TOTALS
*EXCLUDE* 32 HIDDEN ON VOLUME DMPU21
NAME RENAME LIB VV.MM CREATED CHANGED SIZE INIT MOD USERID
CEETEST1 -G: FAIL 1
CEETEST2 -G: FAIL 1
CUSTCOPY -G: FAIL 1
CUSTMAST -G: FAIL 1
CUSTREC -G: FAIL 1
CUSTREC1 -G: FAIL 1
CUSTREC2 -G: FAIL 1
CUST1 -G: FAIL 1
CUST1V2 -G: FAIL 1
CUST2 -G: FAIL 1
CUST2COB -G: FAIL 1
CUST2CPY -G: FAIL 1
CUST2CUS -G: FAIL 1
CUST2PRO -G: FAIL 1
CUST2RDF -G: FAIL 1
F1=HELP F2=SPLIT F3=END F4=IPT View F5=RFIND F6=RCHANGE
F7=UP F8=DOWN F9=SWAP F10=LEFT F11=RIGHT F12=RETRIEVE

```

Figure 3-33 COPY the remaining members

Identify the COPYLIB target library, as shown in Figure 3-34.

```

-IPT----- COPY PANEL ----- (FROM DNET424.PROD.SOURCE) -----
COMMAND ==>
Specify "T0" data set below (45 members will be processed):

PROJECT ==> DNET424
GROUP ==> PROD
TYPE ==> copylib

Or other Partitioned, Sequential Data Set or @H (History List):
DATA SET NAME ==>
VOLUME SERIAL ==> (If not catalogued)
PASSWORD ==> (If password protected)

COPY OPTIONS:
REPLACE like-named library members ==> Y (Y=Yes, N=No, 0=Target is Older)
Disposition for sequential target ==> OLD (OLD or MOD)
Note: If statistics are not available, replace option 0 (older) is
treated as option Y (Yes).

Press ENTER to copy END key to cancel.

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

```

Figure 3-34 Target library for copy members

Press Enter to confirm the members, as shown in Figure 3-35 on page 96.

```

-IPT--VIEW----- COPY PANEL -----
COMMAND ==>

You have requested that 45 members be COPIED to DNET424.PROD.COPYLIB.

Indicate a new member name in the RENAME column, or leave it blank to
use the same member name.

      Press ENTER to process the list, or the END key to cancel.
      (Use the PF keys for scrolling to see the entire list.)

      NAME      RENAME      LIB VV.MM CREATED      CHANGED      SIZE INIT  MOD USERID
CEETEST1
CEETEST2
CUSTCOPY
CUSTMAST
CUSTREC
CUSTREC1
CUSTREC2
CUST1
CUST1V2
CUST2
F1=HELP      F2=SPLIT      F3=END      F4=IPT View  F5=RFIND      F6=RCHANGE
F7=UP        F8=DOWN       F9=SWAP     F10=LEFT    F11=RIGHT     F12=RETRIEVE

```

Figure 3-35 Confirmation panel for COPY

This completes the Member Selection List Example 1, as shown in Figure 3-36.

```

File Display Library Settings Menu Utilities Test Help Exit
-----
-IPT--VIEW L1----- DNET424.PROD.SOURCE -----ROW 00001 OF 00045
COMMAND ==>
HOTBAR: REFRESH FLIP GLOBAL INFO COMPRESS EXPDIR TAILOR TOTALS
*EXCLUDE* 32 HIDDEN ON VOLUME DMPU21
      NAME      RENAME      LIB VV.MM CREATED      CHANGED      SIZE INIT  MOD USERID
CEETEST1 -COPIED      1
CEETEST2 -COPIED      1
CUSTCOPY -COPIED      1
CUSTMAST -COPIED      1
CUSTREC  -COPIED      1
CUSTREC1 -COPIED      1
CUSTREC2 -COPIED      1
CUST1    -COPIED      1
CUST1V2  -COPIED      1
CUST2    -COPIED      1
CUST2COB -COPIED      1
CUST2CPY -COPIED      1
CUST2CUS -COPIED      1
CUST2PRO -COPIED      1
CUST2RDF -COPIED      1
F1=HELP      F2=SPLIT      F3=END      F4=IPT View  F5=RFIND      F6=RCHANGE
F7=UP        F8=DOWN       F9=SWAP     F10=LEFT    F11=RIGHT     F12=RETRIEVE

```

Figure 3-36 Results of the COPY request

### Copy only members that are identified by two or more text fields

In this example, we show you how to copy only PDS members that have both the literals PROCESS and PROGRAM-ID to another data set. Many of the steps that are required are the same as the example in “Copy only members identified by text data within a member” on page 90.

To copy PDS members with the literal PROCESS *and* PROGRAM-ID:

1. Browse, Edit, or View the PDS using the ISPF Productivity Tool.
2. Enter the Global Find (GL) command without any parameters.
3. Enter the data, as shown in Figure 3-37 on page 97.

```

-IPT- ----- GLOBAL EDIT COMMANDS -----
COMMAND ==>
                                SCROLL ==> CSR

STOP AFTER   ==> 9999 (Number of members to process successfully)
PROMPT AFTER ==> 100 (Number of members to process before prompt is issued)
AUTOMATIC    ==> Y (Process without editing successful members?)
LINK         ==> Y (Process each command only if previous command succeeds?)
PRINT        ==> N (Generate listing of each member changed and saved?)
EXCLUDE      ==> N (Exclude failing members from selection list?)
Specify below the ISPF EDIT commands or macros to be executed (one per line).
Press END to process the global commands, or enter CANCEL to cancel.
-----
***** ***** Top of Data *****
000001 FIND PROCESS first
000002 FIND program-id first
***** ***** Bottom of Data *****

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

```

Figure 3-37 Global Find Command with LINK=Y

4. The value of the LINK field determines if the search values are joined by an “or” condition or an “and” condition.
  - The value of Y in the LINK field ensures that the criteria is joined by an “and” condition. Both values must be found for the search to be considered “-G:OK”. The FIRST parameter on the FIND command ensures that the entire member is searched, not where the preceding FIND command completed.
  - A value of N in the LINK field does not link the find commands together. If the value N is used, the conditions are joined by an “or” condition.
5. Press PF3 to invoke the Global Search.
6. As shown in the previous section, separate COBOL programs from copybook members, and use the FILTER command to identify the members with “-G:OK” in the RENAME field.
7. Continue with the COPY process, as shown in the previous section.

### Identify members and move them to another PDS

Through the next series of figures, we will move several members that are identified by a member naming convention and change the date to another library beginning with Figure 3-38 on page 98.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (B) ----- OBJECTS LIST ----- Row 23 to 26 of 26
Command ==> 26 e SCROLL ==> CSR
Hotbar: OPRINT REFRESH CLRVOL FILLVOL UTIL UPDATE CUT FLIP
Open list ==> LAB (or BLANK for reference list)

Command Member Numbr Data Set Names / Objects Class
-----
23 'DNET424.IPT.CUSTFILE
24 <DNET424.DEV1.SOURCE SCLM
25 ADLAB.JCL PDSE
26 =DNET424 NEW ADLAB JCL ISPF
----- END OF LIST -----

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

```

Figure 3-38 Edit DNET424.NEW.JCL and DNET424.ADLAB.JCL

In Figure 3-39 we use the FILTER S\* command to select members.

```

File Display Library Settings Menu Utilities Test Help Exit
-----
-IPT--EDIT L1----- DNET424.ADLAB.COPYLIB ----- "A" will display assist
COMMAND ==> filter s* SCROLL ==> CSR
HOTBAR: REFRESH FLIP GLOBAL INFO COMPRESS EXPDIR TAILOR TOTALS
ON VOLUME DMPU15

NAME RENAME LIB VV.MM CREATED CHANGED SIZE INIT MOD USERID
CEETEST1 1
CEETEST2 1
CUSTCOPY 1
CUSTMAST 1
CUSTREC 1
CUSTREC1 1
CUSTREC2 1
CUST1 1
CUST1V2 1
CUST2 1
CUST2COB 1
CUST2CPY 1
CUST2CUS 1
CUST2PRO 1
CUST2RDF 1

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

```

Figure 3-39 Filter only "S" members

In Figure 3-40 on page 99, the FILTER command results in eight members. The MOVE \* command moves only the eight members that result from the FILTER command.



```

File Display Library Settings Menu Utilities Test Help Exit
-----
-IPT--EDIT L1----- DNET424.ADLAB.COPYLIB -----ROW 00001 OF 00008
COMMAND ==> move *                                SCROLL ==> CSR
HOTBAR: REFRESH FLIP      GLOBAL  INFO      COMPRESS EXPDIR  TAILOR  TOTALS
*FILTER*                   52 HIDDEN      60 PROCESSED  ON VOLUME DMPU15
NAME      RENAME  LIB  VV.MM  CREATED      CHANGED      SIZE  INIT  MOD  USERID
SHIRE      1 01.02 07/09/04 07/09/04 10:43      44  44  1 DNET424
SHIRENP    1 01.00 07/09/04 07/09/04 10:13       5   5  0 GEORGE
SHIREN2    1 01.00 07/09/04 07/09/04 10:13       4   4  0 GEORGE
SHRLSUB    1 01.00 07/09/04 07/09/04 10:13      18  18  0 GEORGE
STRLSUB    1 01.02 07/09/04 07/09/04 10:43      15  15  1 DNET424
SUBLSHR    1 01.01 07/09/04 07/09/04 10:41      44  44  0 DNET424
SUBLSTR    1 01.00 07/09/04 07/09/04 10:13      10  10  0 GEORGE
SUBURB     1 01.00 07/09/04 07/09/04 10:13      14  14  0 GEORGE
--END--

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

```

Figure 3-40 Move "S" members

In Figure 3-41, enter DNET424.ADLAB.COPYLIB to identify the target library.

```

-IPT----- MOVE PANEL ----- (FROM DNET424.ADLAB.COPYLIB) -----
COMMAND ==>
Specify "T0" data set below (10 members will be processed):

PROJECT ==> DNET424
GROUP   ==> ADLAB
TYPE    ==> COPYLIBi

Or other Partitioned, Sequential Data Set or @H (History List):
DATA SET NAME ==>
VOLUME SERIAL ==> (If not catalogued)
PASSWORD      ==> (If password protected)

MOVE OPTIONS:
REPLACE like-named library members ==> 0 (Y=Yes, N=No, 0=Target is Older)
Disposition for sequential target ==> OLD (OLD or MOD)
Note: If statistics are not available, replace option 0 (older) is
treated as option Y (Yes).

Press ENTER to move END key to cancel.

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

```

Figure 3-41 Identify target library

Note the "REPLACE like-name library members" selection of "O". Using this parameter, the MOVE operation occurs only if the target member is older than the source, which minimizes accidental destruction of your data.

The MOVE panel, shown in Figure 3-42, depicts all of the members prior to the MOVE operation, which gives you an opportunity to rename members.

```

-IPT--EDIT----- MOVE PANEL -----
COMMAND ==> _

You have requested that 8 members be MOVED to DNET424.ADLAB.COPYLIBI.

Indicate a new member name in the RENAME column, or leave it blank to
use the same member name.

      Press ENTER to process the list, or the END key to cancel.
      (Use the PF keys for scrolling to see the entire list.)

```

NAME	RENAME	LIB	VV.MM	CREATED	CHANGED	SIZE	INIT	MOD	USERID
SHIRE		1	01 02	07/09/04	07/09/04 10:43	44	44	1	DNET424
SHIRENP		1	01 00	07/09/04	07/09/04 10:13	5	5	0	GEORGE
SHIREN2		1	01 00	07/09/04	07/09/04 10:13	4	4	0	GEORGE
SHRLSUB		1	01 00	07/09/04	07/09/04 10:13	18	18	0	GEORGE
STRLSUB		1	01 02	07/09/04	07/09/04 10:43	15	15	1	DNET424
SUBLSHR		1	01 01	07/09/04	07/09/04 10:41	44	44	0	DNET424
SUBLSTR		1	01 00	07/09/04	07/09/04 10:13	10	10	0	GEORGE
SUBURB		1	01 00	07/09/04	07/09/04 10:13	14	14	0	GEORGE

```

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

```

Figure 3-42 Move panel

In Figure 3-43, six of the eight members remain in the source library because the target member was older than the source. Two members were moved successfully.

```

File Display Library Settings Menu Utilities Test Help Exit
-IPT--EDIT L1----- DNET424.ADLAB.COPYLIB -----ROW 00001 OF 00006
COMMAND ==> _
HOTBAR: REFRESH FLIP GLOBAL INFO COMPRESS EXPDIR TAILOR TOTALS
*FILTER* 52 HIDDEN ON VOLUME DMPU15

```

NAME	RENAME	LIB	VV.MM	CREATED	CHANGED	SIZE	INIT	MOD	USERID
SHIRENP	-NOT OLD	1	01.00	07/09/04	07/09/04 10:13	5	5	0	GEORGE
SHIREN2	-NOT OLD	1	01.00	07/09/04	07/09/04 10:13	4	4	0	GEORGE
SHRLSUB	-NOT OLD	1	01.00	07/09/04	07/09/04 10:13	18	18	0	GEORGE
SUBLSHR	-NOT OLD	1	01.01	07/09/04	07/09/04 10:41	44	44	0	DNET424
SUBLSTR	-NOT OLD	1	01.00	07/09/04	07/09/04 10:13	10	10	0	GEORGE
SUBURB	-NOT OLD	1	01.00	07/09/04	07/09/04 10:13	14	14	0	GEORGE

```

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

```

Figure 3-43 Newer member are not moved

## Using ISPF concatenated libraries to manage change

An ISPF library is a cataloged partitioned data set with a three-level data set name in this format:

project.group.type

Using ISPF, you can concatenate up to four ISPF libraries with the same project and type, as shown in Figure 3-44.

```
ISPF Library:
Project . . . PAYROLL
Group . . . SMITH . . . DEVELOP . . . MASTER . . .
Type . . . PLI
Member . . . (Blank or pattern for member selection list)
```

Figure 3-44 ISPF library

Using an ISPF Productivity Tool Object List, you can specify the same library structure as:  
=project group1 ... group4 type

Figure 3-45 shows how you can define an ISPF library using an ISPF-PT Object list.

```
File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (E) ----- OBJECTS LIST ----- Row 27 to 27 of 27
Command ==>
Hotbar: OPRINT REFRESH CLRVOL FILLVOL UTIL UPDATE CUT FLIP
Open list ==> LAB (or BLANK for reference list)
Command Member Numbr Data Set Names / Objects Class
-----
27 =PAYROLL SMITH DEVELOP MASTER PLI ISPF
-----
END OF LIST -----

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

Figure 3-45 ISPF library defined by an Object List

ISPF libraries can be PDS or PDSE libraries. They might also be SCLM libraries.

Chapter 6, “SCLM integration with the ISPF Productivity Tool” on page 141 demonstrates the use of SCLM with the IPT Member Selection Lists.

ISPF libraries provide an easy way to separate the changes you make from the “production” library. If a member is modified, a version is saved in library1 or group1. Therefore, if the original member is not in group1, the original version is not lost.

In Figure 3-45, changes are saved in the following library:

PAYROLL.SMITH.PLI.

**Helpful hint:** If you want to copy all members to another PDS or PDSE that have a specific value somewhere in the member, you can use the Global Change command with an ISPF library, for example, the command “GL CHANGE ALL PTM01 PTM01” saves all members in the first library that have the value PTM01. Because the source and target value are the same, the data is not changed.

There are several ISPF-PT commands that are designed to expedite your work when you use ISPF libraries or SCLM libraries:

- ▶ PROject – Switch to another project qualifier
- ▶ LIB – Add, remove, or switch to another library
- ▶ TYPE – Switch to another library type
- ▶ WHERE – (H) line command – Shows which libraries contain the member

Let us take a look at how to use the ISPF libraries with ISPF-PT, beginning with Figure 3-46.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (E) ----- OBJECTS LIST ----- Row 22 to 26 of 26
Command ===> 26 SCROLL ===> CSR
Hotbar: OPRINT REFRESH CLRVOL FILLVOL UTIL UPDATE CUT FLIP
Open list ===> LAB (or BLANK for reference list)

Command  Member  Numbr Data Set Names / Objects  Class
-----
22 'DNET424.ADLAB.LOAD
23 'DNET424.IPT.CUSTFILE
24 <DNET424.DEV1.SOURCE SCLM
25 ADLAB.JCL
26 =DNET424.ADLAB.SOURCE ISPF
----- END OF LIST -----

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

```

Figure 3-46 Using an Object List Concatenated data set

Note the ISPF value in the Class field. The equal (=) sign denotes an ISPF library in the OLIST.

In Figure 3-47 on page 103, DNET424.ADLAB.SOURCE identifies the group1 library. The LIB2=ADLAB identifies the second library in the concatenation. To switch from SOURCE to JCL libraries, use the TYPE command.

```

File Display Library Settings Menu Utilities Test Help Exit
-----
-IPT--EDIT L1-----DNET424.ADWORK.SOURCE-----"A" will display assist
COMMAND ==> type jcl SCROLL ==> CSR
HOTBAR: REFRESH FLIP GLOBAL INFO COMPRESS EXPDIR TAILOR TOTALS
LIB2=ADLAB
NAME RENAME LIB VV.MM CREATED CHANGED SIZE INIT MOD USERID
@FILECHK 1 01.00 07/09/04 07/09/04 09:30 15 15 0 DNET424
@SYSINFO 1 01.00 07/09/04 07/09/04 09:08 8 8 0 DNET424
@VSAMDEF 1 01.00 07/09/04 07/09/04 09:07 21 21 0 DNET424
A 1 01.00 07/09/04 07/09/04 09:30 15 15 0 DNET424
ADSORT 1
ADSTAT 1
ASAMDRV 1
ASAM1 1 01.02 07/08/10 07/09/04 09:18 228 227 0 DNET424
ASAM2 1
ATCDEMO 1
ATCDEM2 1
ATCDEM4 1
ATCDEM5 1
ATM01A 1
ATM01B 1
F1=HELP F2=SPLIT F3=END F4=IPT View F5=RFIND F6=RCHANGE
F7=UP F8=DOWN F9=SWAP F10=LEFT F11=RIGHT F12=RETRIEVE

```

Figure 3-47 ISPF library with an MSL

As we see in Figure 3-48, we are editing the ISPF library type of JCL. The LIB command adds or removes libraries. Enter A LIB if you want to see the syntax of the LIB command. We used the LIB 3 PROD command to add PROD as the third library.

```
File Display Library Settings Menu Utilities Test Help Exit
-----
-IPT--EDIT L1-----DNET424.ADWORK.JCL-----ROW 00001 OF 00102
COMMAND ==> lib 3 prod SCROLL ==> CSR
HOTBAR: REFRESH FLIP GLOBAL INFO COMPRESS EXPDIR TAILOR TOTALS
LIB2=ADLAB
NAME RENAME LIB VV.MM CREATED CHANGED SIZE INIT MOD USERID
$JOB CARD 2
BADSTAT 1 01.00 06/04/25 06/04/25 10:28 35 35 0 DNET424
BAPAPI 2
BASAM1 2
BASAM1DR 1 01.00 06/04/25 06/04/25 10:30 33 33 0 DNET424
BASSEM 2
BATCDEMO 1 01.00 06/04/25 06/04/25 10:31 142 142 0 DNET424
BATM01 2
BCOBOLE 2
BCOBOLO 2
BCOBOL2 2
BCOBTIMS 2
BCOBVSAM 2
BCPROG 2
BC01 2
F1=HELP F2=SPLIT F3=END F4=IPT View F5=RFIND F6=RCHANGE
F7=UP F8=DOWN F9=SWAP F10=LEFT F11=RIGHT F12=RETRIEVE
```

Figure 3-48 LIB MSL command

As shown in Figure 3-49, use the PRO or PROJECT command to change the first library qualifier.

```

File Display Library Settings Menu Utilities Test Help Exit
-----
-IPT--EDIT L1----- DNET424.ADWORK.JCL -----ROW 00001 OF 00102
COMMAND ==> pro_dnet074 SCROLL ==> CSR
HOTBAR: REFRESH FLIP GLOBAL INFO COMPRESS EXPDIR TAILOR TOTALS
      LIB2=ADLAB LIB3=PROD
  NAME      RENAME    LIB VV.MM  CREATED      CHANGED      SIZE    INIT    MOD  USERID
$JOBCARD      2
BADSTAT      1 01.00 06/04/25 06/04/25 10:28    35    35    0 DNET424
BAPAAPI      2
BASAM1      2
BASAM1DR     1 01.00 06/04/25 06/04/25 10:30    33    33    0 DNET424
BASSEM      2
BATCDEMO     1 01.00 06/04/25 06/04/25 10:31   142   142    0 DNET424
BATM01      2
BCOBOLE      2
BCOBOLO      2
BCOBOL2      2
BCOBTIMS     2
BCOBVSAM     2
BCPROG      2
BC01         2

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

```

Figure 3-49 PROJECT MSL command

Now, there are three libraries concatenated:

DNET424.ADWORK.JCL

DNET424.ADLAB.JCL

DNET424.PROD.JCL

As shown in Figure 3-50, the “LIB 3 – “command removes the PROD library.

```
File Display Library Settings Menu Utilities Test Help Exit
-----
-IPT--EDIT L1----- DNET424.ADWORK.JCL -----ROW 00001 OF 00102
COMMAND ==> lib 3 - SCROLL ==> CSR
HOTBAR: REFRESH FLIP GLOBAL INFO COMPRESS EXPDIR TAILOR TOTALS
LIB2=ADLAB LIB3=PROD
NAME RENAME LIB VV.MM CREATED CHANGED SIZE INIT MOD USERID
$JOBCARD 2
BADSTAT 1 01.00 06/04/25 06/04/25 10:28 35 35 0 DNET424
BAPAAPI 2
BASAM1 2
BASAM1DR 1 01.00 06/04/25 06/04/25 10:30 33 33 0 DNET424
BASSEM 2
BATCDEMO 1 01.00 06/04/25 06/04/25 10:31 142 142 0 DNET424
BATM01 2
BCOBOLE 2
BCOBOLO 2
BCOBOL2 2
BCOBTIMS 2
BCOBVSAM 2
BCPROG 2
BC01 2
F1=HELP F2=SPLIT F3=END F4=IPT View F5=RFIND F6=RCHANGE
F7=UP F8=DOWN F9=SWAP F10=LEFT F11=RIGHT F12=RETRIEVE
```

Figure 3-50 LIB MSL command – remove a library

The PROJECT command that we show in Figure 3-51 switches to a different high-level qualifier.

```
File Display Library Settings Menu Utilities Test Help Exit
-----
-IPT--EDIT L1----- DNET424.ADWORK.JCL -----ROW 00001 OF 00102
COMMAND ==> pro dnet074 SCROLL ==> CSR
HOTBAR: REFRESH FLIP GLOBAL INFO COMPRESS EXPDIR TAILOR TOTALS
LIB2=ADLAB
NAME RENAME LIB VV.MM CREATED CHANGED SIZE INIT MOD USERID
$JOBCARD 2
BADSTAT 1 01.00 06/04/25 06/04/25 10:28 35 35 0 DNET424
BAPAPI 2
BASAM1 2
BASAM1DR 1 01.00 06/04/25 06/04/25 10:30 33 33 0 DNET424
BASSEM 2
BATCDEMO 1 01.00 06/04/25 06/04/25 10:31 142 142 0 DNET424
BATM01 2
BCOBOLE 2
BCOBOLO 2
BCOBOL2 2
BCOBTIMS 2
BCOBVSAM 2
BCPROG 2
BC01 2
F1=HELP F2=SPLIT F3=END F4=IPT View F5=RFIND F6=RCHANGE
F7=UP F8=DOWN F9=SWAP F10=LEFT F11=RIGHT F12=RETRIEVE
```

Figure 3-51 PROJECT MSL command

As shown in Figure 3-52, the library high-level qualifier is DNET074. To determine where the member resides, enter the line command h.

```

File Display Library Settings Menu Utilities Test Help Exit
-----
-IPT--EDIT L1----- DNET074.ADWORK.JCL ----- "A" will display assist
COMMAND ==> SCROLL ==> CSR
HOTBAR: REFRESH FLIP GLOBAL INFO COMPRESS EXPDIR TAILOR TOTALS
LIB2=ADLAB
NAME RENAME LIB VV.MM CREATED CHANGED SIZE INIT MOD USERID
$AJ 1
$JOBCARD 1 01.00 06/05/19 06/05/19 15:08 2 2 0 DNET074
$JSAMP 1
$SET 1
BADSTAT 1
BAPAPI 2
h BASAM1 1
BASAM1DR - 1
BASSEM 2 01.01 07/01/31 07/02/01 11:16 68 90 0 DNET074
BATCDEMO 1
BATM01 2
BCOBOLE 2 01.06 07/01/28 07/01/28 17:04 76 74 0 DNET074
BCOBOLO 2 01.02 07/01/28 07/02/02 10:01 80 80 0 DNET074
BCOBOL2 2 01.00 07/01/28 07/01/28 17:04 71 71 0 DNET074
BCOBTIMS 1
F1=HELP F2=SPLIT F3=END F4=IPT View F5=RFIND F6=RCHANGE
F7=UP F8=DOWN F9=SWAP F10=LEFT F11=RIGHT F12=RETRIEVE

```

Figure 3-52 WHERE MSL command

In Figure 3-53, the BASAM1 member resides in libraries 1 and 2 or ADWORK and ADLAB, which are depicted in the RENAME column.

File Display Library Settings Menu Utilities Test Help Exit									
-----									
-IPT--EDIT L1----- DNET074.ADWORK.JCL					-----ROW 00007 OF 00111				
COMMAND ==> dsn					SCROLL ==> CSR				
HOTBAR: REFRESH FLIP					TAILOR TOTALS				
					LIB2=ADLAB				
NAME	RENAME	LIB	VV.MM	CREATED	CHANGED	SIZE	INIT	MOD	USERID
BASAM1	-IN:12	1							
BASAM1DR		1							
BASSEM		2	01.01	07/01/31	07/02/01 11:16	68	90	0	DNET074
BATCDEMO		1							
BATM01		2							
BCOBOLE		2	01.06	07/01/28	07/01/28 17:04	76	74	0	DNET074
BCOBOL0		2	01.02	07/01/28	07/02/02 10:01	80	80	0	DNET074
BCOBOL2		2	01.00	07/01/28	07/01/28 17:04	71	71	0	DNET074
BCOBTIMS		1							
BCOBVSAM		2							
BCPROG		2							
BC01		2							
BDTDEMO		1							
BIMSSTUB		2	01.00	07/07/24	07/07/24 23:50	38	38	0	DNET074
BLABONE		1							
F1=HELP F2=SPLIT F3=END F4=IPT View F5=RFIND F6=RCHANGE									
F7=UP F8=DOWN F9=SWAP F10=LEFT F11=RIGHT F12=RETRIEVE									

Figure 3-53 Result of the WHERE MSL command

The DSN command switches the Member Selection List to another data set. If the DSN command is entered without parameters, it invokes Option 2 of ISPF-PT.

The DSN command allows you to change the library name or the invocation parameters that might not be available using the Member Selection List, for example, you can enter an EDIT/VIEW initial Macro or profile name. In Figure 3-54, we entered the library TYPE of LOAD.

```

Menu Reflist Refmode Utilities Settings Test Help Exit
-----
-IPT--L1                EDIT - ENTRY PANEL

COMMAND ==> _____
HOTBAR?

More: +

ISPF LIBRARY:
  Project ==> DNET074
  Group   ==> ADWORK   ==> ADLAB   ==> _____ ==> _____
  Type    ==> load
  Member  ==> *        (Blank or pattern for selection list)
Other data set, VSAM file, @H, or @L for 'DNET074.ADLAB.JCL(*)':
  DSN/Cat. level ==> _____
  Volume serial  ==> _____ (Optional VOLSER or pattern for selection list)
  Password       ==> _____ (If password protected)
Default process  ==> E        (B=Browse, V=View, E=Edit, BF, EF, VF, or ?)
Execute TAILOR   ==> N        (Y=Yes ,N=no, D=define commands)

EDIT/VIEW parameters:
  Initial Macro ==> _____ Confirm Cancel/Move/Replace ==> N (Y, N)
  Profile Name  ==> _____ Action Bar in Edit/View      ==> Y (Y, N)
  Format Name    ==> _____ Highlight coloring in Edit/View ==> Y (Y, N)
                                     Exclusive access of viewed file ==> Y (Y, N)

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

```

Figure 3-54 DSN command



**Hint:** You can navigate directly to another data set using the DSN command with parameters, for example, the command DSN CUSTFILE.KSDS navigates directly to the user ID.CUSTFILE.KSDS file from a Member Selection List.

As shown in Figure 3-55, the 1 command displays information about the load module.

File Display Library Settings Menu Utilities Test Help Exit												
-----												
-IPT--EDIT L1----- DNET424.ADWORK.LOAD -----ROW 00001 OF 00018												
COMMAND ==>										SCROLL ==> CSR		
HOTBAR: REFRESH FLIP GLOBAL INFO COMPRESS EXPDIR TAILOR TOTALS												
LIB2=ADLAB												
NAME	RENAME	LIB	SIZE	TTR	ALIAS-OF	AC	RENT	REFR	REUS	TEST	AMOD	RMOD
ADSTAT		2	001C10	000D21		00					ANY	24
1 ASAMDRV	-	2	0013D0	000213		00					ANY	24
ASAM1		2	000888	000F1C		00					31	24
ATCDEMO		2	032C70	00030E		00					ANY	24
ATCDEM2		2	001570	000808		00					ANY	24
ATCDEM4		2	001350	000815		00					ANY	24
ATCDEM5		2	0023C0	000822		00					ANY	24
COBISTUB		2	001AC8	000220		00					ANY	24
DTDEMO		2	001388	000106		00					ANY	24
IMSSTUB		2	0017E8	000301		00					31	ANY
PLISAM		1	006F48	00050D		00					31	ANY
PLISAMT		1	006F28	000007		00					31	ANY
SAM1		2	003048	001513		00					ANY	24
SAM1V		2	0035B0	001121		00					31	ANY
SAM2		2	001388	001414		00					ANY	24
F1=HELP F2=SPLIT F3=END F4=IPT View F5=RFIND F6=RCHANGE												
F7=UP F8=DOWN F9=SWAP F10=LEFT F11=RIGHT F12=RETRIEVE												

Figure 3-55 MSL line command "L"

Use PF11 to view the right part of the report, as shown in Figure 3-56.

```

-IPT--Module analysis:DNET424.ADLAB.LOAD ----- LINE 00000020 COL 001 080
COMMAND ==>
SCROLL ==> CSR
Commands:Down, End, Find, Up
Member ==> ASAMDRV
Display mode ==> M (M=Map, H=History)
----- Press END to exit -----
*****LOAD MODULE PROCESSED EITHER BY VS LINKAGE EDITOR OR BIN
NUMERICAL MAP AND CROSS-REFERENCE LIST OF LOAD MODU
ASAMDRV MODULE ANALYSIS
CONTROL SECTION
ENTRY
LMOD LOC NAME LENGTH TYPE LMOD LO
00 ASAMDRV 780 SD
780 CEESG005 18 SD
798 CEEBETBL 28 SD
7C0 CEESTART 80 SD
870 IGZCBSO 578 SD
DE8 CEEARLU 80 SD
E98 CEEBPIRA 2D0 SD
E98
E98
E98
1168 CEECPYRT E2 SD
1250 CEEBPUBT 70 SD
F1=HELP F2=SPLIT F3=END F4=IPT View F5=RFIND F6=RCHANGE
F7=UP F8=DOWN F9=SWAP F10=LEFT F11=RIGHT F12=RETRIEVE

```

Figure 3-56 Load Module Members

## 3.6 MSL Point-and-Shoot fields

The ISPF Productivity Tool provides a large number of *Point-and-Shoot* fields on the Object List and Member Selection Panels. To use a Point-and-Shoot field, put your cursor under the field label, and press Enter.

The Member Selection List panel provides the following Point-and-Shoot fields.

- ▶ EDIT/BROWSE/VIEW
- ▶ Data set history
- ▶ Tailor Messages
- ▶ Column headings
  - NAME
  - RENAME
  - LIB
  - VV.MM
  - CREATED
  - CHANGED
  - SIZE
  - INIT
  - MOD
  - USERID
- ▶ HOTBAR

In this section, we describe the location and use of each of the fields listed above.

In Figure 3-57, the “action” field changes from Edit to Browse to View when using the Point-and-Shoot function. This action equates to using the DEFAULT command.

The screenshot shows the ISPF Member Selection List panel. At the top is a menu bar with options: File, Display, Library, Settings, Menu, Utilities, Test, Help, Exit. Below the menu bar, the current action is set to 'EDIT L1', which is circled in red. The panel displays a list of data sets with columns: NAME, RENAME, LIB, VV.MM, CREATED, CHANGED, SIZE, INIT, MOD, USERID. The list includes data sets like CEETEST1, CEETEST2, CUSTCOPY, CUSTMAST, CUSTREC, CUSTREC1, CUSTREC2, CUST1, CUST1V2, CUST2, CUST2COB, CUST2CPY, CUST2CUS, CUST2PRO, and CUST2RDF. At the bottom, there are function key definitions: F1=HELP, F2=SPLIT, F3=END, F4=IPT View, F5=RFIND, F6=RCHANGE, F7=UP, F8=DOWN, F9=SWAP, F10=LEFT, F11=RIGHT, F12=RETRIEVE.

```
File Display Library Settings Menu Utilities Test Help Exit
-----
-IPTC-EDIT L1----- DNET424.ADLAB.COPYLIB ----- "A" will display assist
COMMAND ==> SCROLL ==> CSR
HOTBAR: REFRESH FLIP GLOBAL INFO COMPRESS EXPDIR TAILOR TOTALS
ON VOLUME DMPU15
NAME RENAME LIB VV.MM CREATED CHANGED SIZE INIT MOD USERID
CEETEST1 1
CEETEST2 1
CUSTCOPY 1
CUSTMAST 1
CUSTREC 1
CUSTREC1 1
CUSTREC2 1
CUST1 1
CUST1V2 1
CUST2 1
CUST2COB 1
CUST2CPY 1
CUST2CUS 1
CUST2PRO 1
CUST2RDF 1
F1=HELP F2=SPLIT F3=END F4=IPT View F5=RFIND F6=RCHANGE
F7=UP F8=DOWN F9=SWAP F10=LEFT F11=RIGHT F12=RETRIEVE
```

Figure 3-57 Point-and-Shoot field Edit/Browse/View

Figure 3-58 identifies the Data set history Point-and-Shoot field.

```

File Display Library Settings Menu Utilities Test Help Exit
-----
-IPT--EDIT L1-----DNET424.ADLAB.COPYLIB----- "A" will display assist
COMMAND ==>          SCROLL ==> CSR
HOTBAR: REFRESH  FLIP      GLOBAL  INFO      COMPRESS EXPDIR  TAILOR  TOTALS
              ON VOLUME DMPU15

  NAME      RENAME    LIB VV.MM CREATED      CHANGED      SIZE  INIT   MOD USERID
CEETEST1           1
CEETEST2           1
CUSTCOPY           1
CUSTMAST           1
CUSTREC            1
CUSTREC1           1
CUSTREC2           1
CUST1              1
CUST1V2            1
CUST2              1
CUST2COB           1
CUST2CPY           1
CUST2CUS           1
CUST2PRO           1
CUST2RDF           1

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

```

Figure 3-58 Data set history Point-and-Shoot field

The Data set history Point-and-Shoot results in a dynamic Object List of the most recently used data sets, as shown in Figure 3-59.

File Edit Find Display Populate Settings Menu Util Test Help Exit									
-----									
-IPT- OLIST (V)			DATA SET HISTORY			"A" will display assist			
Command ==>			SCROLL ==> CSR						
Hotbar: OPRINT			REFRESH	CLRVOL	FILLVOL	UTIL	UPDATE	CUT	FLIP
			*TEMPORARY LIST*						
-----									
Command	Member	Numbr	Data Set Names / Objects						Class
-----									
		1	'DNET424.ADLAB.COPYLIB'						
		2	'DNET424.ADLAB.SOURCE'						
		3	'DNET424.ADWORK.SOURCE'						
	*	4	'DNET424.ADWORK.SOURCE'						
	*	5	'DNET424.ADLAB.SOURCE'						
	*	6	'DNET424.ADLAB.LOAD'						
	*	7	'DNET424.ADWORK.LOAD'						
	*	8	'DNET074.ADLAB.JCL'						
	*	9	'DNET074.ADWORK.JCL'						
	*	10	'DNET074.ADWORK.SOURCE'						
	*	11	'DNET074.ADLAB.SOURCE'						
	*	12	'DNET074.ADWORK.LOAD'						
	*	13	'DNET074.ADLAB.LOAD'						
F1=HELP	F2=SPLIT	F3=END	F4=IPT View	F5=RFIND	F6=RCHANGE				
F7=UP	F8=DOWN	F9=SWAP	F10=LEFT	F11=RIGHT	F12=RETRIEVE				

Figure 3-59 Data set history Object List

ISPF-PT automatically retains the last 100 data sets you used. They are presented when you request the history data sets using the above action.

The Tailor message can also be a Point-and-Shoot field. In Figure 3-60 on page 110, we previously issued the FILTER C\* command, which reduced the members shown.

```

File Display Library Settings Menu Utilities Test Help Exit
-----
-IPT--EDIT L1----- DNET424.ADLAB.COPYLIB -----ROW 00001 OF 00016
COMMAND ==> SCROLL ==> CSR
HOTBAR: REFRESH FLIP GLOBAL INFO COMPRESS EXPDIR TAILOR TOTALS
*FILTER* 42 HIDDEN 58 PROCESSED ON VOLUME DMPU15
  NAME      RENAME    LIB  VV.MM  CREATED      CHANGED      SIZE  INIT  MOD  USERID
CEETEST1           1
CEETEST2           1
CUSTCOPY           1
CUSTMAST           1
CUSTREC            1
CUSTREC1           1
CUSTREC2           1
CUST1              1
CUST1V2            1
CUST2              1
CUST2COB           1
CUST2CPY           1
CUST2CUS           1
CUST2PRO           1
CUST2RDF           1

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

```

Figure 3-60 Tailor Message Point-and-Shoot

To remove the filtering, use the Point-and-Shoot field that is circled in Figure 3-60, which is equivalent to the UNFilter command.

You can point-and-shoot using any of the column headings to sort the members. In Figure 3-61, we chose to sort the members by CREATED date.

```
File Display Library Settings Menu Utilities Test Help Exit
-----
-IPT--EDIT L1----- DNET424.ADLAB.COPYLIB -----ROW 00001 OF 00058
COMMAND ==> SCROLL ==> CSR
HOTBAR: REFRESH FLIP GLOBAL INFO COMPRESS EXPDIR TAILOR TOTALS
ON VOLUME DMPU15
NAME RENAME LIB VV.MM CREATED CHANGED SIZE INIT MOD USERID
CEETEST1 1
CEETEST2 1
CUSTCOPY 1
CUSTMAST 1
CUSTREC 1
CUSTREC1 1
CUSTREC2 1
CUST1 1
CUST1V2 1
CUST2 1
CUST2COB 1
CUST2CPY 1
CUST2CUS 1
CUST2PRO 1
CUST2RDF 1
F1=HELP F2=SPLIT F3=END F4=IPT View F5=RFIND F6=RCHANGE
F7=UP F8=DOWN F9=SWAP F10=LEFT F11=RIGHT F12=RETRIEVE
```

Figure 3-61 Point-and-Shoot using MSL column headings

Figure 3-62 depicts the members, sorted in CREATED sequence.

File Display Library Settings Menu Utilities Test Help Exit											
-----											
-IPT--EDIT L1----- DNET424.ADLAB.COPYLIB						-----ROW 00001 OF 00058					
COMMAND ==> █						SCROLL ==> CSR					
HOTBAR: REFRESH		FLIP		GLOBAL INFO		COMPRESS EXPDIR		TAILOR		TOTALS	
*SORT*						ON VOLUME DMPU15					
NAME	RENAME	LIB	VV.MM	CREATED	CHANGED	SIZE	INIT	MOD	USERID		
SHIRENP		1	01.00	07/09/04	07/09/04 10:13	5	5	0	GEORGE		
SHIREN2		1	01.00	07/09/04	07/09/04 10:13	4	4	0	GEORGE		
SHRLSUB		1	01.00	07/09/04	07/09/04 10:13	18	18	0	GEORGE		
SUBLSHR		1	01.01	07/09/04	07/09/04 10:41	44	44	0	DNET424		
SUBLSTR		1	01.00	07/09/04	07/09/04 10:13	10	10	0	GEORGE		
SUBURB		1	01.00	07/09/04	07/09/04 10:13	14	14	0	GEORGE		
CEETEST1		1									
CEETEST2		1									
CUSTCOPY		1									
CUSTMAST		1									
CUSTREC		1									
CUSTREC1		1									
CUSTREC2		1									
CUST1		1									
CUST1V2		1									
F1=HELP		F2=SPLIT		F3=END		F4=IPT View		F5=RFIND		F6=RCHANGE	
F7=UP		F8=DOWN		F9=SWAP		F10=LEFT		F11=RIGHT		F12=RETRIEVE	

Figure 3-62 Results of Point-and-Shoot using MSL column heading CREATED

## 3.7 MSL Hotbars

The *MSL HOTBAR* provides a unique way for you to tailor your MSL window with frequently used ISPF-PT commands using Point-and-Shoot technology.

When you first use ISPF-PT, the HOTBAR fields are not populated. You can update the HOTBAR fields by using the "HOTBAR?" heading as a Point-and-Shoot field, as shown in Figure 3-63.

```
File Display Library Settings Menu Utilities Test Help Exit
-----
-IPT--EDIT L1----- DNET424.ADLAB.COPYLIB -----ROW 00001 OF 00058
COMMAND ==>
SCROLL ==> CSR
HOTBAR?
ON VOLUME DMPU15
NAME RENAME LIB VV.MM CREATED CHANGED SIZE INIT MOD USERID
CEESTEST1 1
CEESTEST2 1
CUSTCOPY 1
CUSTMAST 1
CUSTREC 1
CUSTREC1 1
CUSTREC2 1
CUST1 1
CUST1V2 1
CUST2 1
CUST2COB 1
CUST2CPY 1
CUST2CUS 1
CUST2PRO 1
CUST2RDF 1
F1=HELP F2=SPLIT F3=END F4=IPT View F5=RFIND F6=RCHANGE
F7=UP F8=DOWN F9=SWAP F10=LEFT F11=RIGHT F12=RETRIEVE
```

Figure 3-63 Updating the MSL HOTBAR

At this point, you can enter up to eight MSL commands in the HOTBAR, as shown in Figure 3-64.

```
File Display Library Settings Menu Utilities Test Help Exit
-----
-IPT--EDIT L1----- DNET424.ADLAB.COPYLIB ----- Specify HOTBAR commands
COMMAND ==> SCROLL ==> CSR
HOTBAR= █
-----
NAME RENAME LIB VV.MM CREATED CHANGED ON VOLUME DMPU15 SIZE INIT MOD USERID
CEETEST1 1
CEETEST2 1
CUSTCOPY 1
CUSTMAST 1
CUSTREC 1
CUSTREC1 1
CUSTREC2 1
CUST1 1
CUST1V2 1
CUST2 1
CUST2COB 1
CUST2CPY 1
CUST2CUS 1
CUST2PRO 1
CUST2RDF 1
F1=HELP F2=SPLIT F3=END F4=IPT View F5=RFIND F6=RCHANGE
F7=UP F8=DOWN F9=SWAP F10=LEFT F11=RIGHT F12=RETRIEVE
```

Figure 3-64 Updating the MSL HOTBAR

You can use any valid MSL command and HOTBAR command. You can invoke the HOTBAR commands using Point-and-Shoot functions. The MSL window permanently retains your HOTBAR commands. An example is shown in Figure 3-65.

```

File Display Library Settings Menu Utilities Test Help Exit
-----
-IPT--EDIT L1----- DNET424.ADLAB.COPYLIB ----- Specify HOTBAR commands
COMMAND ==> SCROLL ==> CSR
HOTBAR= █ REFRESH FLIP GLOBAL INFO COMPRESS EXPDIR TAILOR TOTALS__
ON VOLUME DMPU15

NAME RENAME LIB VV.MM CREATED CHANGED SIZE INIT MOD USERID
CEETEST1 1
CEETEST2 1
CUSTCOPY 1
CUSTMAST 1
CUSTREC 1
CUSTREC1 1
CUSTREC2 1
CUST1 1
CUST1V2 1
CUST2 1
CUST2COB 1
CUST2CPY 1
CUST2CUS 1
CUST2PRO 1
CUST2RDF 1

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

```

Figure 3-65 Updating the MSL Hotbar

You can use the mouse to Point-and-Shoot. See Appendix D, “Customizing the IBM Personal Communications” on page 305 for additional information.

## 3.8 Using the TAILOR command

The TAILOR command defines and initiates one or more MSL commands. They can be defined using Option 1, Option 2, or when using a Member Selection List.

As shown in Figure 3-66, the ISPF-PT Option 1 or 2, Browse/View or view window, provides a field that is labeled Execute TAILOR command.

```

Menu  Reflist  Refmode  Utilities  Settings  Test  Help  Exit
-----
-IPT--L1          BROWSE - ENTRY PANEL
COMMAND ==> _____
HOTBAR? _____

ISPFLIBRARY:
Project ==> DNET424
Group   ==> ADWORK   ==> ADLAB   ==> _____ ==> _____
Type    ==> SOURCE
Member  ==> _____ (Blank or pattern for selection list)
Other data set, VSAM file, @H, or @L for 'DNET424.ADLAB.SOURCE':
DSN/Cat. level ==> _____
Volume serial ==> _____ (Optional VOLSER or pattern for selection list)
Password       ==> _____ (If password protected)
Default process ==> B (B=Browse, V=View, E=Edit, BF, EF, VF, or ?)
Execute TAILOR ==> d (Y=Yes, N=no, D=define commands)
EDIT/VIEW parameters:
Initial Macro ==> _____ Confirm Cancel/Move/Replace ==> N (Y, N)
Profile Name  ==> _____ Action Bar in Edit/View ==> Y (Y, N)
Format Name   ==> _____ Highlight coloring in Edit/View ==> Y (Y, N)
Exclusive access of viewed file ==> Y (Y, N)

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

```

Figure 3-66 Tailor Command

An action of D displays Figure 3-67.

```

-IPT- -----MSL TAILOR COMMAND-----
COMMAND ==> _____

Member List commands executed by TAILOR command:
==> FILTER CHA -60;SORT CHA

Examples:
==> FILTER USERID DNET424
==> FILTER ACT*A;SORT CHANGED

Other Member Selection List options:
Automatic preview (with LOCATE/FIND) ==> Y (Y=Yes, N=No)
Replace existing members (COPY/MOVE) ==> 0 (Y=Yes, N=No, 0=Target Older)

Press ENTER to execute, END key to return to previous screen.

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

```

Figure 3-67 Tailor Command definition

Figure 3-68 on page 114, the command FILTER CHA -60, retains only members that were modified in the last 60 days. The SORT CHA command sorts the remaining members by descending date and time. You can update the Automatic Preview and Replace existing members fields on this window.

As shown in Figure 3-68, the Execute TAILOR command is automatically set to Y. Press Enter to browse the MSL using the Tailor commands.

Menu
Reflist
Refmode
Utilities
Settings
Test
Help
Exit

- IPT--L1
BROWSE - ENTRY PANEL

COMMAND ==>

HOTBAR?

More: +

ISPF LIBRARY:

Project ==> DNET424
Group ==> ADWORK
Type ==> SOURCE
Member ==>

(Blank or pattern for selection list)

Other data set, VSAM file, @H, or @L for 'DNET424.ADLAB.SOURCE':

DSN/Cat. level ==>
Volume serial ==>
Password ==>

(Optional VOLSER or pattern for selection list)

(If password protected)

Default process ==> B
Execute TAILOR ==> Y

(B=Browse, V=View, E=Edit, BF, EF, VF, or ?)

(Y=Yes ,N=no, D=define commands)

EDIT/VIEW parameters:

Initial Macro ==>
Profile Name ==>
Format Name ==>

Confirm Cancel/Move/Replace ==> N (Y, N)
Action Bar in Edit/View ==> Y (Y, N)
Highlight coloring in Edit/View ==> Y (Y, N)
Exclusive access of viewed file ==> Y (Y, N)

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

Figure 3-68 Tailor Command

Figure 3-69 shows only members that changed in the last 60 days.

File
Display
Library
Settings
Menu
Utilities
Test
Help
Exit

- IPT--BROWSE L1--- DNET424.ADLAB.SOURCE ----- "A" will display assist
COMMAND ==>
SCROLL ==> CSR

HOTBAR: REFRESH FLIP GLOBAL INFO COMPRESS EXPDIR TAILOR TOTALS

\*FILTER\* \*SORT\* 71 HIDDEN LIB2=ADLAB

NAME	RENAME	LIB	VV.MM	CREATED	CHANGED	SIZE	INIT	MOD	USERID
@FILECHK		1	01.00	07/09/04	07/09/04 09:30	15	15	0	DNET424
A		1	01.00	07/09/04	07/09/04 09:30	15	15	0	DNET424
B		1	01.00	07/09/04	07/09/04 09:30	15	15	0	DNET424
C		1	01.00	07/09/04	07/09/04 09:30	14	14	0	DNET424
E		1	01.00	07/09/04	07/09/04 09:30	15	15	0	DNET424
ASAM1		1	01.02	07/08/10	07/09/04 09:18	228	227	0	DNET424
@SYSINFO		1	01.00	07/09/04	07/09/04 09:08	8	8	0	DNET424
@VSAMDEF		1	01.00	07/09/04	07/09/04 09:07	21	21	0	DNET424
SAM1		2	01.01	07/08/29	07/08/29 12:41	439	439	0	DNET424
DEMO		1	01.00	07/08/15	07/08/15 13:54	7	7	0	DNET424
SUBXMP		2	01.00	07/08/13	07/08/13 15:34	47	47	0	DNET424

--END--

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

Figure 3-69 Tailor Command used with the MSL

Members are listed in “Changed” descending date and time sequence. 71 members are hidden because they do not meet the Tailor criteria.

You can invoke the TAILOR command directly from the MSL window that is shown in Figure 3-69, for example, the command “TAILOR?” invokes the MSL TAILOR COMMAND window. The command TAILOR executes the currently defined commands from the Member selection list.



## Enhanced cut and paste

The ISPF Productivity Tool CUT and PASTE commands provide these additional benefits:

- ▶ Supports up to 200 CUT/PASTE clipboards. Clipboards can be named or numbered. You can edit, browse, copy, save, restore, and rename these clipboards.
- ▶ Ability to CUT excluded and non-excluded lines. You can CUT context-sensitive lines by combining the EXCLUDE and FIND commands with the CUT command.
- ▶ Both CUT and PASTE commands support the STATus keyword to display the list of all existing clipboards. You can use this display to select a new or existing clipboard to CUT into or from which to PASTE.
- ▶ You can paste lines from different sources:
  - Previously cut lines
  - Lines from another member
  - Captured output of TSO commands
  - Member names of a specified directory
  - Contents of a previously cut OLIST
- ▶ You can paste lines from the different sources of CUT directly to the printer.

Assistance for the CUT and PASTE command is available by typing the word CUT or PASTE on the command line of a member (not in browse), and pressing Enter.

Figure 4-1 on page 116 shows an example of cutting text and placing it in a clipboard called JC1. You must be in view or edit to use the **cut** command. If you use the CUT command and do not specify 'TO', your cut text is placed on a default ("00") clipboard.

You can block the areas you want to cut, or if you want the entire member, use the command word ALL with the CUT command.

**Note:** Only temporary (non-persistent) clipboards are deleted when you exit ISPF-PT. However, you can choose to make even clipboard 00 a persistent one.

```
File Edit Edit_Settings Menu Utilities Compilers Test Help
-IPT- EDIT NW.TJM.TESTING.IPT59(JOBCARD) - 01.04 Columns 00001 00072
Command ==> cut to jc1 Scroll ==> CSR
***** ***** Top of Data *****
cc0001 //NWTJM1 JOB NWTJM,'TESTING ID',CLASS=9,MSGCLASS=H
cc0002 /**
***** ***** Bottom of Data *****
```

Figure 4-1 Cut example 1

Figure 4-2 shows the message '2 to JC1' in the top right corner. This is the number of lines that were cut and the name of the clipboard they were cut to.

```
File Edit Edit_Settings Menu Utilities Compilers Test Help
-IPT- EDIT NW.TJM.TESTING.IPT59(JOBCARD) - 01.04 2 to JC1
Command ==> Scroll ==> CSR
***** ***** Top of Data *****
000001 //NWTJM1 JOB NWTJM,'TESTING ID',CLASS=9,MSGCLASS=H
000002 /**
***** ***** Bottom of Data *****
```

Figure 4-2 Results of Cut Example 1

Figure 4-3 shows the CUT to XXX SAVE command that makes the clipboard a persistent (saved) clipboard.

```

File Edit Edit_Settings Menu Utilities Compilers Test Help
-IPT- EDIT NW.TJM.TESTING.IPT59(ADDUSER) - 01.53 Columns 00001 00072
Command ==> cut to steps save Scroll ==> CSR
***** ***** Top of Data *****
000100 //RACFADMU JOB 210VRHDGENRLP,SECADM,CLASS=4,MSGCLASS=X,NOTIFY=&SYSUID
000200 //*ROUTE PRINT RMT24
000300 /*-----
000400 /* ADD A 'PRIMARY' USER-ID FOR AN INDIVIDUAL -- MULTIPLE JOBS (ALIASE)
000600 /* CHANGE INFORMATION (USER-ID) AS NEEDED AND RUN
000800 /*-----
000900 /* JOB-1
001000 /* 1. ADD THE BASIC USER, GROUP CONNECTIONS, AND TSO INFORMATION
001100 /* 2. LIST USER FOR REVIEW
001200 /* JOB-2,3
001300 /* 3. CREATE THE HLQ/USERCAT-ALIASE -- PROD / ACCP
001400 /* JOB-4
001500 /* 3. CREATE THE HLQ/USERCAT-ALIASE -- TEST
001600 /* 4. CREATE THE ISPPROF DS
001700 /* 5. VERIFY THE HLQ/USERCAT-ALIASE & BRODCAST DS
001800 /*-----
001900 //STEP010 EXEC PGM=IKJEFT01,DYNAMNBR=20
F1=Help F2=Split F3=Exit F5=Rfind F6=Rchange F7=Up
F8=Down F9=Swap F10=Left F11=Right F12=Cancel

```

Figure 4-3 Cut example 2

Figure 4-4 shows that nine lines were cut to the clipboard called STEPS.

```

File Edit Edit_Settings Menu Utilities Compilers Test Help
-IPT- EDIT NW.TJM.TESTING.IPT59(ADDUSER) - 01.53 9 to STEPS
Command ==> Scroll ==> CSR
***** ***** Top of Data *****
000100 //RACFADMU JOB 210VRHDGENRLP,SECADM,CLASS=4,MSGCLASS=X,NOTIFY=&SYSUID
000200 //*ROUTE PRINT RMT24
000300 /*-----
000400 /* ADD A 'PRIMARY' USER-ID FOR AN INDIVIDUAL -- MULTIPLE JOBS (ALIASE)
000600 /* CHANGE INFORMATION (USER-ID) AS NEEDED AND RUN
000800 /*-----
000900 /* JOB-1
001000 /* 1. ADD THE BASIC USER, GROUP CONNECTIONS, AND TSO INFORMATION
001100 /* 2. LIST USER FOR REVIEW
001200 /* JOB-2,3
001300 /* 3. CREATE THE HLQ/USERCAT-ALIASE -- PROD / ACCP
001400 /* JOB-4
001500 /* 3. CREATE THE HLQ/USERCAT-ALIASE -- TEST
001600 /* 4. CREATE THE ISPPROF DS
001700 /* 5. VERIFY THE HLQ/USERCAT-ALIASE & BRODCAST DS
001800 /*-----
001900 //STEP010 EXEC PGM=IKJEFT01,DYNAMNBR=20
F1=Help F2=Split F3=Exit F5=Rfind F6=Rchange F7=Up
F8=Down F9=Swap F10=Left F11=Right F12=Cancel

```

Figure 4-4 Results of Cut Example 2

Figure 4-5 shows some JCL that does not have a job card. If you type the command CUT STATUS or CUT STA, and press enter, you can see all of the clipboards you defined.

```

File Edit Edit_Settings Menu Utilities Compilers Test Help
-IPT- EDIT NW TJM TESTING.IPT59(BLBCTM2) - 01.23 Columns 00001 00072
Command ==> cut sta Scroll ==> CSR
***** Top of Data *****
000001 /**
000002 //STP1 EXEC PGM=IEFBR14,REGION=4M
000003 //SYSPRINT DD SYSOUT=*
000004 //INSTALL DD DSN=NWX.XXXINST.V5.INSTALL,DISP=(,CATLG,DELETE),
000005 // DCB=(BLKSIZE=3120,LRECL=80,RECFM=FB,DSORG=PO),
000006 // UNIT=SYSALLDA,SPACE=(TRK,(75,15,75))
000007 //INSTCTM DD DSN=NWX.XXXINST.V5.INSTCTM,DISP=(,CATLG,DELETE),
000008 // DCB=(BLKSIZE=3120,LRECL=80,RECFM=FB,DSORG=PO),
000009 // UNIT=SYSALLDA,SPACE=(TRK,(75,15,75))
000010 //CLIST DD DSN=NWX.XXXINST.V5.CLIST,DISP=(,CATLG,DELETE),
000011 // DCB=(BLKSIZE=3120,LRECL=80,RECFM=FB,DSORG=PO),
000012 // UNIT=SYSALLDA,SPACE=(TRK,(75,15,75))
000013 //PANEL DD DSN=NWX.XXXINST.V5.PANEL,DISP=(,CATLG,DELETE),
000014 // DCB=(BLKSIZE=3120,LRECL=80,RECFM=FB,DSORG=PO),
000015 // UNIT=SYSALLDA,SPACE=(TRK,(75,15,75))
000016 //PROCLIB DD DSN=NWX.XXXINST.V5.PROCLIB,DISP=(,CATLG,DELETE),
000017 // DCB=(BLKSIZE=3120,LRECL=80,RECFM=FB,DSORG=PO),
F1=Help F2=Split F3=Exit F5=Rfind F6=Rchange F7=Up
F8=Down F9=Swap F10=Left F11=Right F12=Cancel

```

Figure 4-5 Cut Status

Figure 4-6 shows your clipboards. The saved ones are marked with an \*. You can save them by using the command CUT TO XXX SAVE when you create them, or you can save it on the CUT STATUS panel by typing SAV next to it. After a clipboard is SAVED, it is called a persistent clipboard. You can RENAME, DELETE, and RESTORE persistent clipboards. If you do not save your clipboard, it is deleted when you exit ISPF-PT.

```

-IPT----- ACTIVE CLIPBOARDS -----
Command ==> _ Scroll ==> CSR

Command line clipboard selection: 0 to 99 or name
Line commands: B browse C copy E edit P print S select V view
                DEL delete REN rename RES restore SAV save
Press ENTER to process or F3 to cancel.

  8 active clipboards out of 200
Persistent clipboard repository NW.TJM.IPITBLIB
Persistent clipboards marked by *
Clip-
Cmd   board Records Size First line of text in clipboard
-----
*.. JC1          2   80 //NWTJM1 JOB NWTJM,'TESTING ID',CLASS=9,MSGCLASS=
*.. JC2          3   80 //NWTJM1 JOB NWTJM,'TESTING ID',CLASS=9,MSGCLASS=
*.. LISTCAT       1   80 LISTC ENT( ) ALL
*.. SPACECYL      4   80 //SYSPRINT DD DSN=YOUR.DATASET.NAME
*.. SPACETRK      4   80 // UNIT=SYSDA,
*.. STEPS         9   80 /** JOB-1
*.. WORKDIR       5   80 ADDISPF 01 48 99/04/05 07/08/28 10:19 11 122
... 00           3   80 //DEFALI EXEC PGM=IDCAMS,REGION=1M
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 4-6 Active clipboards

Figure 4-7 shows the paste command, which is pasting from the clipboard called JC1 on the line b (before) the /\*.

```

File Edit Edit_Settings Menu Utilities Compilers Test Help

-IPT- EDIT NW.TJM.TESTING.IPT59(BLBCTM2) - 01.23 Columns 00001 00072
Command ==> paste_from_jc1 Scroll ==> CSR
***** Top of Data *****
b00001 /*
000002 //STP1 EXEC PGM=IEFBR14,REGION=4M
000003 //SYSPRINT DD SYSOUT=*
000004 //INSTALL DD DSN=NWX.XXXINST.V5.INSTALL,DISP=(,CATLG,DELETE),
000005 // DCB=(BLKSIZE=3120,LRECL=80,RECFM=FB,DSORG=PO),
000006 // UNIT=SYSALLDA,SPACE=(TRK,(75,15,75))
000007 //INSTCTM DD DSN=NWX.XXXINST.V5.INSTCTM,DISP=(,CATLG,DELETE),
000008 // DCB=(BLKSIZE=3120,LRECL=80,RECFM=FB,DSORG=PO),
000009 // UNIT=SYSALLDA,SPACE=(TRK,(75,15,75))
000010 //CLIST DD DSN=NWX.XXXINST.V5.CLIST,DISP=(,CATLG,DELETE),
000011 // DCB=(BLKSIZE=3120,LRECL=80,RECFM=FB,DSORG=PO),
000012 // UNIT=SYSALLDA,SPACE=(TRK,(75,15,75))
000013 //PANEL DD DSN=NWX.XXXINST.V5.PANEL,DISP=(,CATLG,DELETE),
000014 // DCB=(BLKSIZE=3120,LRECL=80,RECFM=FB,DSORG=PO),
000015 // UNIT=SYSALLDA,SPACE=(TRK,(75,15,75))
000016 //PROCLIB DD DSN=NWX.XXXINST.V5.PROCLIB,DISP=(,CATLG,DELETE),
000017 // DCB=(BLKSIZE=3120,LRECL=80,RECFM=FB,DSORG=PO),
F1=Help F2=Split F3=Exit F5=Rfind F6=Rchange F7=Up
F8=Down F9=Swap F10=Left F11=Right F12=Cancel

```

Figure 4-7 Paste example 1

Figure 4-8 shows the results of the paste.

```

File Edit Edit_Settings Menu Utilities Compilers Test Help

-IPT- EDIT NW.TJM.TESTING.IPT59(BLBCTM2) - 01.24 2 from JC1
Command ==> Scroll ==> CSR
***** Top of Data *****
000001 //NWTJM1 JOB NWTJM,'TESTING ID',CLASS=9,MSGCLASS=H
000002 /*
000003 /*
000004 //STP1 EXEC PGM=IEFBR14,REGION=4M
000005 //SYSPRINT DD SYSOUT=*
000006 //INSTALL DD DSN=NWX.XXXINST.V5.INSTALL,DISP=(,CATLG,DELETE),
000007 // DCB=(BLKSIZE=3120,LRECL=80,RECFM=FB,DSORG=PO),
000008 // UNIT=SYSALLDA,SPACE=(TRK,(75,15,75))
000009 //INSTCTM DD DSN=NWX.XXXINST.V5.INSTCTM,DISP=(,CATLG,DELETE),
000010 // DCB=(BLKSIZE=3120,LRECL=80,RECFM=FB,DSORG=PO),
000011 // UNIT=SYSALLDA,SPACE=(TRK,(75,15,75))
000012 //CLIST DD DSN=NWX.XXXINST.V5.CLIST,DISP=(,CATLG,DELETE),
000013 // DCB=(BLKSIZE=3120,LRECL=80,RECFM=FB,DSORG=PO),
000014 // UNIT=SYSALLDA,SPACE=(TRK,(75,15,75))
000015 //PANEL DD DSN=NWX.XXXINST.V5.PANEL,DISP=(,CATLG,DELETE),
000016 // DCB=(BLKSIZE=3120,LRECL=80,RECFM=FB,DSORG=PO),
000017 // UNIT=SYSALLDA,SPACE=(TRK,(75,15,75))
F1=Help F2=Split F3=Exit F5=Rfind F6=Rchange F7=Up
F8=Down F9=Swap F10=Left F11=Right F12=Cancel

```

Figure 4-8 Results of the paste

You can use the CUT TO XXXX APP command to append additional sections into an existing clipboard. Figure 4-9 on page 120 shows an additional line being appended to the clipboard JC1.

```

File Edit Edit_Settings Menu Utilities Compilers Test Help

-IPT- EDIT NW.TJM.TESTING.IPT59(IEBCOPY1) - 01.05 Columns 00001 00072
Command ==> cut to jcl app Scroll ==> CSR
***** ***** Top of Data *****
000001 //NWTJMC JOB NWTJM,'IEBCOPY ',CLASS=9,MSGCLASS=H,NOTIFY=&SYSUID
000002 //JOBPARM L=20000
000003 //IEBCOPY1 EXEC PGM=IEBCOPY
000004 //SYSPRINT DD SYSOUT=A
000005 //SYSUT3 DD UNIT=SYSDA,DISP=NEW,SPACE=(TRK,(10,1))
000006 // UNIT=SYSDA,
000007 // VOL=SER=PRISV1
000008 // SPACE=(TRK,(10,5,5),RLSE),
000009 // DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120)
000010 //SYSUT4 DD UNIT=SYSDA,DISP=NEW,SPACE=(TRK,(10,1))
000011 //SYSUT1 DD DISP=SHR,DSN=BOR.BKUP.Y95JUN.IEP.SOURCE
000012 //SYSUT2 DD DISP=SHR,DSN=NW.TJM.CHARLENE
000013 //SYSIN DD DUMMY
***** ***** Bottom of Data *****

```

Figure 4-9 Cut Example with Append

Figure 4-10 shows a member called LISTDIR.

```

File Edit Edit_Settings Menu Utilities Compilers Test Help

-IPT- EDIT NW.TJM.TESTING.IPT59(LISTDIR) - 01.00 Columns 00001 00072
Command ==> Scroll ==> CSR
***** ***** Top of Data *****
000100 $FOCLOG 01 28 01/12/19 05/09/01 09:41 60 62 0 NWTJM
000200 HMYLIB 01 08 04/11/04 07/08/28 14:26 8 8 0 NWTJM
000300 AANOTES 01 99 93/11/17 07/09/06 13:58 153 35 153 NWTJM
000400 ABAIDUT1 01 48 97/05/06 07/08/29 10:03 30 26 0 NWTJM
000500 ABAIDUT2 01 22 99/08/19 07/08/28 10:19 25 25 0 NWTJM
000600 ABAIDUT3 01 21 99/08/20 07/08/29 10:44 25 25 0 NWTJM
000700 ABENDRID 01 31 98/03/20 07/10/10 14:25 33 20 0 NWTJM
000800 ACCOUNT 01 11 05/03/29 07/08/22 08:11 14 1 14 NWTJMT
000900 ACF$TSDA 01 07 07/07/13 07/08/29 10:44 170 168 0 NWTJM
001000 ACF2RPT 01 02 05/01/31 05/02/01 17:15 11 9 0 NWTJM
001100 ACF2TSDA 01 09 07/07/13 07/08/29 10:44 171 168 0 NWTJM
001200 ADDISPF 01 48 99/04/05 07/08/28 10:19 11 122 5 NWTJM
001300 ADDPTRGN 01 99 96/05/30 07/08/29 10:44 48 25 44 NWTJM
001400 ADDPTRUI 01 16 97/11/05 07/08/29 10:44 48 27 32 NWTJM
001500 ADDUSER 01 53 98/06/15 02/09/11 13:30 94 115 57 IMOKP
001600 ADDVRPRT 01 02 97/12/03 07/08/29 10:44 35 25 18 NWTJM
001700 ADVANTIS 01 06 98/07/31 07/08/29 10:44 14 13 0 NWTJM
F1=Help F2=Split F3=Exit F5=Rfind F6=Rchange F7=Up
F8=Down F9=Swap F10=Left F11=Right F12=Cancel

```

Figure 4-10 Edit session with LISTDIR member

Figure 4-11 is the LISTDIR member with all of the lines excluded and a FIND (F) all members that starts with the IEB command.

```
File Edit Edit_Settings Menu Utilities Compilers Test Help

-IPT- EDIT NW.TJM.TESTING.IPT59(LISTDIR) - 01.00 Columns 00001 00072
Command ==> f ieb all Scroll ==> CSR
***** Top of Data *****
- - - - - 644 Line(s) not Displayed
***** Bottom of Data *****
```

Figure 4-11 FIND members starting with IEB

Figure 4-12 shows the results of the FIND (F) command. There were 26 member names that started with the characters IEB.

```
File Edit Edit_Settings Menu Utilities Compilers Test Help

-IPT- EDIT NW.TJM.TESTING.IPT59(LISTDIR) - 01.00 26 CHARS 'IEB'
Command ==> Scroll ==> CSR
***** Top of Data *****
- - - - - 281 Line(s) not Displayed
028200 IEBCOPY 01 49 85/09/12 07/10/10 11:01 15 10 0 NWTJM
028300 IEBCOPYX 01 04 03/04/01 07/03/14 14:43 11 11 0 NWTJMT
028400 IEBCOPY1 01 05 06/03/06 07/10/10 11:13 13 10 0 NWTJM
028500 IEBCOPY3 01 17 05/03/11 07/03/26 11:10 11 11 0 NWTJM
028600 IEBCPYL 01 15 03/05/13 07/08/22 13:35 13 15 0 NWTJMT
028700 IEBCPY3 01 18 98/08/18 07/03/14 14:43 15 26 0 NWTJMT
028800 IEBCPY4 01 09 99/08/09 07/03/14 14:43 129 68 0 NWTJMT
028900 IEBFCCR 01 18 99/06/07 06/04/06 12:57 12 10 0 NWTJM
029000 IEBGENER 01 18 99/06/07 07/08/22 13:35 16 10 0 NWTJMT
029100 IEBGEN2 01 18 99/06/07 06/09/07 13:35 11 10 0 NWTJM
029200 IEBGNRX 01 49 98/08/18 07/02/23 15:41 10 8 0 NWTJM
029300 IEBGNR1 01 09 00/05/01 05/09/01 09:41 10 10 0 NWTJM
029400 IEBGNR4 01 13 98/08/18 05/09/01 09:41 8 10 0 NWTJM
029500 IEBGNR5 01 18 97/07/16 05/09/01 09:41 149 14 0 NWTJM
029600 IEBGNR6 01 25 97/07/16 05/09/01 09:41 52 14 0 NWTJM
029700 IEBGPRNT 01 33 03/12/12 05/03/22 14:14 30 10 0 NWRDH
F1=Help F2=Split F3=Exit F5=Rfind F6=Rchange F7=Up
F8=Down F9=Swap F10=Left F11=Right F12=Cancel
```

Figure 4-12 Only IEB lines shown

Figure 4-13 shows how to use the CUT command to create a clipboard that only contains the IEB members.

```

File Edit Edit_Settings Menu Utilities Compilers Test Help

-IPT- EDIT NW.TJM.TESTING.IPT59(LISTDIR) - 01.00 Columns 00001 00072
Command == cut_all to ieb nx Scroll ==> CSR
***** Top of Data *****
281 Line(s) not Displayed
028200 IEBCOPY 01 49 85/09/12 07/10/10 11:01 15 10 0 NWTJM
028300 IEBCOPYX 01 04 03/04/01 07/03/14 14:43 11 11 0 NWTJMT
028400 IEBCOPY1 01 05 06/03/06 07/10/10 11:13 13 10 0 NWTJM
028500 IEBOP3 01 17 05/03/11 07/03/26 11:10 11 11 0 NWTJM
028600 IEBPYL 01 15 03/05/13 07/08/22 13:35 13 15 0 NWTJMT
028700 IEBPY3 01 18 98/08/18 07/03/14 14:43 15 26 0 NWTJMT
028800 IEBPY4 01 09 99/08/09 07/03/14 14:43 129 68 0 NWTJMT
028900 IEBFCCR 01 18 99/06/07 06/04/06 12:57 12 10 0 NWTJM
029000 IEBGENER 01 18 99/06/07 07/08/22 13:35 16 10 0 NWTJMT
029100 IEBGEN2 01 18 99/06/07 06/09/07 13:35 11 10 0 NWTJM
029200 IEBGNRX 01 49 98/08/18 07/02/23 15:41 10 8 0 NWTJM
029300 IEBGNR1 01 09 00/05/01 05/09/01 09:41 10 10 0 NWTJM
029400 IEBGNR4 01 13 98/08/18 05/09/01 09:41 8 10 0 NWTJM
029500 IEBGNR5 01 18 97/07/16 05/09/01 09:41 149 14 0 NWTJM
029600 IEBGNR6 01 25 97/07/16 05/09/01 09:41 52 14 0 NWTJM
029700 IEBGPRNT 01 33 03/12/12 05/03/22 14:14 30 10 0 NWRDH

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

```

Figure 4-13 Cut only lines with IEB text

Figure 4-14 shows that 26 lines were cut to the clipboard called IEB.

```

File Edit Edit_Settings Menu Utilities Compilers Test Help

-IPT- EDIT NW.TJM.TESTING.IPT59(LISTDIR) - 01.00 26 to IEB
Command ==> Scroll ==> CSR
***** Top of Data *****
281 Line(s) not Displayed
028200 IEBCOPY 01 49 85/09/12 07/10/10 11:01 15 10 0 NWTJM
028300 IEBCOPYX 01 04 03/04/01 07/03/14 14:43 11 11 0 NWTJMT
028400 IEBCOPY1 01 05 06/03/06 07/10/10 11:13 13 10 0 NWTJM
028500 IEBOP3 01 17 05/03/11 07/03/26 11:10 11 11 0 NWTJM
028600 IEBPYL 01 15 03/05/13 07/08/22 13:35 13 15 0 NWTJMT
028700 IEBPY3 01 18 98/08/18 07/03/14 14:43 15 26 0 NWTJMT
028800 IEBPY4 01 09 99/08/09 07/03/14 14:43 129 68 0 NWTJMT
028900 IEBFCCR 01 18 99/06/07 06/04/06 12:57 12 10 0 NWTJM
029000 IEBGENER 01 18 99/06/07 07/08/22 13:35 16 10 0 NWTJMT
029100 IEBGEN2 01 18 99/06/07 06/09/07 13:35 11 10 0 NWTJM
029200 IEBGNRX 01 49 98/08/18 07/02/23 15:41 10 8 0 NWTJM
029300 IEBGNR1 01 09 00/05/01 05/09/01 09:41 10 10 0 NWTJM
029400 IEBGNR4 01 13 98/08/18 05/09/01 09:41 8 10 0 NWTJM
029500 IEBGNR5 01 18 97/07/16 05/09/01 09:41 149 14 0 NWTJM
029600 IEBGNR6 01 25 97/07/16 05/09/01 09:41 52 14 0 NWTJM
029700 IEBGPRNT 01 33 03/12/12 05/03/22 14:14 30 10 0 NWRDH

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

```

Figure 4-14 Results of Cut only lines with IEB text

You can enter the RESET (res) command to return the member to its original state or to cancel out.



You can use the PASTE command to combine members in a library. Figure 4-15 and Figure 4-16 show the paste of the member names WHODSN into the member named WHATDSN.

```

File Edit Edit_Settings Menu Utilities Compilers Test Help

-IPT- EDIT NW.TJM.TESTING.IPT59(WHATDSN) - 01.04 Columns 00001 00072
Command === paste member whodsn Scroll ==> CSR
***** ***** Top of Data *****
000010 //NWTJM1 JOB NWTJM,'TEST ID',CLASS=9,MSGCLASS=H
000100 //*****
000200 /* THIS JOB WILL LIST THE NAMES OF DATA SETS IN A DFDSS DUMP. *
000300 //*****
000400 //TAPEVTOC EXEC PGM=ADDRSSU,REGION=2M,PARM='TYPRUN=NORUN'
000500 //SYSPRINT DD SYSOUT=*
000600 //IN DD DISP=SHR,DSN=CICSNW.PROD.ID.RUNTIME.D070123
000700 //SYSIN DD *
000800 RESTORE INDDNAME(IN) -
a00900 DATASET(INCLUDE(**))
001000 /*
001100 //
***** ***** Bottom of Data *****

```

Figure 4-15 Paste entire member

Figure 4-16 depicts the results of the PASTE MEMBER command.

```

File Edit Edit_Settings Menu Utilities Compilers Test Help

-IPT- EDIT NW.TJM.TESTING.IPT59(WHATDSN) - 01.04 11 line(s) pasted.
Command ==> Scroll ==> CSR
***** ***** Top of Data *****
000010 //NWTJM1 JOB NWTJM,'TEST ID',CLASS=9,MSGCLASS=H
000100 //*****
000200 /* THIS JOB WILL LIST THE NAMES OF DATA SETS IN A DFDSS DUMP. *
000300 //*****
000400 //TAPEVTOC EXEC PGM=ADDRSSU,REGION=2M,PARM='TYPRUN=NORUN'
000500 //SYSPRINT DD SYSOUT=*
000600 //IN DD DISP=SHR,DSN=CICSNW.PROD.RUNTIME.D070123
000700 //SYSIN DD *
000800 RESTORE INDDNAME(IN) -
000900 DATASET(INCLUDE(**))
000910 //*****
000920 /* THIS JOB WILL LIST THE NAMES OF DATA SETS IN A DFDSS DUMP. *
000930 //*****
000940 //TAPEVTOC EXEC PGM=ADDRSSU,REGION=2M,PARM='TYPRUN=NORUN'
000950 //SYSPRINT DD SYSOUT=*
000960 //IN DD DISP=SHR,DSN=DUMP.DATASET.NAME
000970 //SYSIN DD *
F1=Help F2=Split F3=Exit F5=Rfind F6=Rchange F7=Up
F8=Down F9=Swsp F10=Left F11=Right F12=Cancel

```

Figure 4-16 Results of Paste Member

When you edit a member, you can use the PASTE command to issue TSO commands and have the data from that command returned to the member you are editing, for example, Figure 4-17 on page 124 shows the PASTE TSO command.

```

File Edit Edit_Settings Menu Utilities Compilers Test Help

-IPT- EDIT NW TJM TESTING.IPT59(IEBCOPY) - 01.49 Columns 00001 00072
Command ==> paste tso Scroll ==> CSR
***** ***** Top of Data *****
b00001 //NXTJM1 JOB NXTJM,'TESTING ID',CLASS=1,MSGCLASS=H
000002 /*
000003 //COPY EXEC PGM=IEBCOPY
000004 //SYSPRINT DD SYSOUT=*
000005 //SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
000006 //INDD DD DSN=NW.TJM.ACF2.WORK
000007 //OUTDD DD DSN=NX.TJM.DELETE.ME,DISP=SHR,
000008 // UNIT=SYSDA,
000009 // VOL=SER=PRISV1
000010 // SPACE=(TRK,(10,5,5),RLSE),
000011 // DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120)
000012 //SYSIN DD *
000013 COPY OUTDD=OUT1,INDD=((IN1,R))
000014 /*
000015 //SYSIN DD DUMMY
***** ***** Bottom of Data *****

```

Figure 4-17 Paste TSO command

After you type the PASTE TSO command, Figure 4-18 is displayed.

```

-IPT----- PASTE TSO COMMAND -----
COMMAND ==>

Enter a TSO command to be pasted:

==> _

Press ENTER to PASTE the TSO output or F3 to cancel.

```

Figure 4-18 TSO command

Figure 4-19 shows the syntax for entering the **listds** command with arguments.

```

-IPT----- PASTE TSO COMMAND -----
COMMAND ==>

Enter a TSO command to be pasted:

==> listds 'nw.tjm.acf2.work'

Press ENTER to PASTE the TSO output or F3 to cancel.

```

Figure 4-19 TSO LISTDS command

Press Enter to return the TSO command results back to the member. Cancel (can) out of the member if you do not want to save the changes. Figure 4-20 shows the TSO command and returned results being pasted in the member.

```

File Edit Edit_Settings Menu Utilities Compilers Test Help

-IPT- EDIT NW.TJM.TESTING.IPT59(IEBCOPY) - 01.50          7 line(s) pasted.
Command ==> Scroll ==> CSR
***** Top of Data *****
000001 LISTDS 'NW.TJM.ACF2.WORK'
000002 ---- RETURN CODE 0 ----
000003 NW.TJM.ACF2.WORK
000004 --RECFM-LRECL-BLKSIZE-DSORG
000005 FB 80 27920 P0
000006 --VOLUMES--
000007 SYSPOC
000008 //NXTJM1 JOB NXTJM,'TESTING ID',CLASS=1,MSGCLASS=H
000009 /*
000010 //COPY EXEC PGM=IEBCOPY
000011 //SYSPRINT DD SYSOUT=*
000012 //SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
000013 //INDD DD DSN=NW.TJM.ACF2.WORK
000014 //OUTDD DD DSN=NX.TJM.DELETE.ME,DISP=SHR,
000015 // UNIT=SYSDA,
000016 // VOL=SER=PRISV1
000017 // SPACE=(TRK,(10,5,5),RLSE),
F1=Help F2=Split F3=Exit F5=Rfind F6=Rchange F7=Up
F8=Down F9=Swap F10=Left F11=Right F12=Cancel

```

Figure 4-20 Results of PASTE TSO command.

The output of the LISTDS command is pasted in the IEBCOPY member.

The PASTE FROM xxx VIEW allows you to view a clipboard (either persistent or a default one) and select (with "CC" and "C" or "F") only the data you want to paste without permanently changing the clipboard. You can skip selecting any data by ending the view. Figure 4-21 shows the CUT command being used to pull data from the output of a batch job to a clipboard called temp.

```

File Edit Edit_Settings Menu Utilities Compilers Test Help

EDIT NWTJMPDS(J037710) LIST1 SYSTSPRT Columns 00001 00072
Command ==> cut all to temp Scroll ==> CSR
***** Top of Data *****
000001 1ACF0C038 ACF2 LOGONID ATTRIBUTES HAVE REPLACED DEFAULT USER ATTRIBUTES
000002 READY
000003 LISTD 'CICSNW.PROD.CTS13.SNWRLOAD' MEMBERS
000004 CICSNW.PROD.CTS13.SNWRLOAD
000005 --RECFM-LRECL-BLKSIZE-DSORG
000006 U ** 23476 P0
000007 --VOLUMES--
000008 PRISV2
000009 --MEMBERS--
000010 ACFAEUSC
000011 ATTRBCNV
000012 CICLEOP
000013 DFHCNV
000014 DFHDCTNW
000015 DFHMCTNW
000016 DFHPLTI1
000017 DFHPLTI2
F1=Help F2=Split F3=Exit F5=Rfind F6=Rchange F7=Up
F8=Down F9=Swap F10=Left F11=Right F12=Cancel

```

Figure 4-21 Cut the entire data set to TEMP

Figure 4-22 shows the PASTE command with the VIEW option.

```

File Edit Edit_Settings Menu Utilities Compilers Test Help
-IPT- EDIT NW.TJM.TESTING.IPT50(COPYMEM) - 01.03 Columns 00001 00072
Command == paste from temp view Scroll ==> CSR
***** Top of Data *****
000100 //NWTJMMOD JOB NWTJM,'COPY LMOD'
000200 /*JOBPARM L=10
000300 //STEP01 EXEC PGM=IEBCOPY
000400 //SYSPRINT DD SYSOUT=A
000500 //SYSUT1 DD DISP=SHR,DSN=CICSNW.PROD.CTS13.PROD.LOADLIB
000600 //SYSUT2 DD DISP=SHR,DSN=NWR.CICS.QA.PROD.LOADLIB
000700 //SYSUT3 DD UNIT=SYSDA,SPACE=(TRK,(5,1))
000800 //SYSUT4 DD UNIT=SYSDA,SPACE=(TRK,(5,1))
000900 //SYSIN DD *
a01000 COPY INDD=SYSUT1,OUTDD=SYSUT2
***** 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 4-22 PASTE using a VIEW parameter

Figure 4-23 is the results of the PASTE from xxxx VIEW command.

```

-IPT----- PASTE DISPLAY ----- Row 1 to 16 of 46
COMMAND ==> SCROLL ==> CSR

Press ENTER to PASTE or END key to cancel. Use the C or CC line command to
select a range to be pasted. Use F main command to search for a string.
***** Clipboard TEMP *****
..... 1ACF0C038 ACF2 LOGONID ATTRIBUTES HAVE REPLACED DEFAULT USER ATTRIBUTES
..... READY
..... LISTD 'CICSNW.PROD.CTS13.SNWRLOAD' MEMBERS
..... CICSNW.PROD.CTS13.SNWRLOAD
..... --RECFM=LRECL-BLKSIZE=DSORG
..... U ** 23476 PO
..... --VOLUMES--
..... PRISV2
..... --MEMBERS--
..... ACFRAEUSC
..... ATTRBCNV
..... CICLEOP
..... DFHCNV
..... DFHDCTNW
..... DFHMCTNW
..... DFHPLTI1
F1=HELP F2=SPLIT F3=END F4=RETURN F5=RFIN F6=RCHANGE
F7=UP F8=DOWN F9=SWAP F10=LEFT F11=RIGHT F12=RETRIEVE

```

Figure 4-23 PASTE view

Figure 4-24 shows the selected text that will be pasted from the clipboard using the VIEW option.

```

-IPT----- PASTE DISPLAY ----- Row 1 to 16 of 46
COMMAND ==>                                SCROLL ==> CSR

Press ENTER to PASTE or END key to cancel. Use the C or CC line command to
select a range to be pasted. Use F main command to search for a string.
***** Clipboard TEMP *****
..... 1ACF0C038 ACF2 LOGONID ATTRIBUTES HAVE REPLACED DEFAULT USER ATTRIBUTES
..... READY
..... LISTD 'CICSNW.PROD.CTS13.SNWRLOAD' MEMBERS
..... CICSNW.PROD.CTS13.SNWRLOAD
..... --RECFM=LRECL-BLKSIZE=DSORG
..... U      **      23476      PO
..... --VOLUMES--
..... PRISV2
..... --MEMBERS--
..... CC..... ACFAEUSC
.....          ATTRBCNV
.....          CICLEOP
.....          DFHCNV
.....          DFHDCTNW
.....          DFHMCTNW
.....          DFHPLTI1
..... CC.....
..... 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 4-24 Selected lines from the PASTE VIEW

Figure 4-25 shows the end result of the PASTE and VIEW command.

```

File Edit Edit_Settings Menu Utilities Compilers Test Help

-IPT- EDIT NW.TJM.TESTING.IPT59(COPYMEM) - 01.04          7 from TEMP
Command ==>                                Scroll ==> CSR
***** Top of Data *****
000100 //NWTJMMOD JOB NWTJM,'COPY LMOD'
000200 /*JOBPARM L=10
000300 //STEP01 EXEC PGM=IEBCOPY
000400 //SYSPRINT DD SYSOUT=A
000500 //SYSUT1 DD DISP=SHR,DSN=CICSNW.PROD.CTS13.PROD.LOADLIB
000600 //SYSUT2 DD DISP=SHR,DSN=NWR.CICS.QA.PROD.LOADLIB
000700 //SYSUT3 DD UNIT=SYSDA,SPACE=(TRK,(5,1))
000800 //SYSUT4 DD UNIT=SYSDA,SPACE=(TRK,(5,1))
000900 //SYSIN DD *
001000 COPY INDD=SYSUT1,OUTDD=SYSUT2
001100 ACFAEUSC
001200 ATTRBCNV
001300 CICLEOP
001400 DFHCNV
001500 DFHDCTNW
001600 DFHMCTNW
001700 DFHPLTI1
.....
F1=Help      F2=Split      F3=Exit      F5=Rfind      F6=Rchange      F7=Up
F8=Down      F9=Swap       F10=Left     F11=Right     F12=Cancel

```

Figure 4-25 End result of the PASTE with VIEW command

The PASTE FROM yyy EDIT command allows you to edit a clipboard (either persistent or a default one) and select only the data you want to paste. Editing changes that you make to the clipboard are permanent. Figure 4-26 on page 128 shows the command.

```

File Edit Edit_Settings Menu Utilities Compilers Test Help

-IPT- EDIT NW.TJM.TESTING.IPT59(COPYMEM) - 01.05 Columns 00001 00072
Command ==> paste from workdir edit Scroll ==> CSR
***** ***** Top of Data *****
000100 //NWTJMMOD JOB NWTJM,'COPY LMOD'
000200 /*JOBPARM L=10
000300 //STEP01 EXEC PGM=IEBCOPY
000400 //SYSPRINT DD SYSOUT=A
000500 //SYSUT1 DD DISP=SHR,DSN=CICSNW.PROD.CTS13.PROD.LOADLIB
000600 //SYSUT2 DD DISP=SHR,DSN=NWR.CICS.QA.PROD.LOADLIB
000700 //SYSUT3 DD UNIT=SYSDA,SPACE=(TRK,(5,1))
000800 //SYSUT4 DD UNIT=SYSDA,SPACE=(TRK,(5,1))
000900 //SYSIN DD *
a01000 COPY INDD=SYSUT1,OUTDD=SYSUT2
***** ***** Bottom of Data *****

```

Figure 4-26 PASTE command with EDIT parameter

Figure 4-27 shows where a block of lines were deleted and all lines past the member ADDVRPRT lines were deleted.

```

EDIT .....Clipboard..WORKDIR Columns 00001 00072
Command ==> Scroll ==> CSR
***** ***** Top of Data *****
dd0001 $FOCLOG 01 28 01/12/19 05/09/01 09:41 60 62 0 NWTJM
000002 HMYLIB 01 08 04/11/04 07/08/28 14:26 8 8 0 NWTJM
000003 AARNOTES 01 99 93/11/17 07/09/06 13:58 153 35 153 NWTJM
000004 ABAIDUT1 01 48 97/05/06 07/08/29 10:03 30 26 0 NWTJM
000005 ABAIDUT2 01 22 99/08/19 07/08/28 10:19 25 25 0 NWTJM
000006 ABAIDUT3 01 21 99/08/20 07/08/29 10:44 25 25 0 NWTJM
000007 ABENDAID 01 31 98/03/20 07/10/10 14:25 33 20 0 NWTJM
000008 ACCOUNT 01 11 05/03/29 07/08/22 08:11 14 1 14 NWTJMT
000009 ACF$TSOR 01 07 07/07/13 07/08/29 10:44 170 168 0 NWTJM
000010 ACF2RPT 01 02 05/01/31 05/02/01 17:15 11 9 0 NWTJM
dd0011 ACF2TSOR 01 09 07/07/13 07/08/29 10:44 171 168 0 NWTJM
000012 ADDISPF 01 48 99/04/05 07/08/28 10:19 11 122 5 NWTJM
000013 ADDPTRGN 01 99 96/05/30 07/08/29 10:44 48 25 44 NWTJM
000014 ADDPTRUI 01 16 97/11/05 07/08/29 10:44 48 27 32 NWTJM
000015 ADDUSER 01 53 98/06/15 02/09/11 13:30 94 115 57 IMOKP
000016 ADDVRPRT 01 02 97/12/03 07/08/29 10:44 35 25 18 NWTJM
d99999 ADVANTIS 01 06 98/07/31 07/08/29 10:44 14 13 0 NWTJM
000018 ALAISCO 01 14 00/11/28 07/08/29 10:45 27 12 0 NWTJM
000019 ALAISDD 01 13 00/11/28 07/08/29 10:45 66 12 0 NWTJM

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

```

Figure 4-27 Clipboard Directory

Figure 4-28 is the edited clipboard.

```

EDIT .....Clipboard..WORKDIR Columns 00001 00072
Command ==> Scroll ==> CSR
***** ***** Top of Data *****
000001 ADDISPF 01 48 99/04/05 07/08/28 10:19 11 122 5 NWTJM
000002 ADDPTRGN 01 99 96/05/30 07/08/29 10:44 48 25 44 NWTJM
000003 ADDPTRUI 01 16 97/11/05 07/08/29 10:44 48 27 32 NWTJM
000004 ADDUSER 01 53 98/06/15 02/09/11 13:30 94 115 57 IMOKP
000005 ADDVRPRT 01 02 97/12/03 07/08/29 10:44 35 25 18 NWTJM
***** ***** Bottom of Data *****

```

Figure 4-28 Results of edited clipboard

Figure 4-29 is the original member with the edited clipboard lines added. The upper-right corner shows the number of lines and the names of the clipboards that were pasted.

```

File Edit Edit_Settings Menu Utilities Compilers Test Help

-IPT- EDIT NW.TJM.TESTING.IPT59(COPYMEM) - 01.06          5 from WORKDIR
Command ==> _____ Scroll ==> CSR
***** ***** Top of Data *****
000100 //NWTJMMOD JOB NWTJM,'COPY LMOD'
000200 /*JOBPARM L=10
000300 //STEP01 EXEC PGM=IEBCOPY
000400 //SYSPRINT DD SYSOUT=A
000500 //SYSUT1 DD DISP=SHR,DSN=CICSNW.PROD.CTS13.PROD.LOADLIB
000600 //SYSUT2 DD DISP=SHR,DSN=NWR.CICS.QA.PROD.LOADLIB
000700 //SYSUT3 DD UNIT=SYSDA,SPACE=(TRK,(5,1))
000800 //SYSUT4 DD UNIT=SYSDA,SPACE=(TRK,(5,1))
000900 //SYSIN DD *
001000 COPY INDD=SYSUT1,OUTDD=SYSUT2
001100 ADDISPF0 01 48 99/04/05 07/08/28 10:19 11 122 5 NWTJM
001200 ADDPTRGN 01 99 96/05/30 07/08/29 10:44 48 25 44 NWTJM
001300 ADDPTRUI 01 16 97/11/05 07/08/29 10:44 48 27 32 NWTJM
001400 ADDUSER 01 53 98/06/15 02/09/11 13:30 94 115 57 IMOKP
001500 ADDVRPRT 01 02 97/12/03 07/08/29 10:44 35 25 18 NWTJM
***** ***** Bottom of Data *****

```

Figure 4-29 Clipboard data added

Archived





## Using IBM File Manager for z/OS with the ISPF Productivity Tool

You can customize the ISPF Productivity Tool to invoke a VSAM file Browser or Editor when you access files from an Object List. In this chapter, we teach you how to customize ISPF-PT to invoke the IBM product, File Manager. We also teach you some basic File Manager commands and how to use a copybook with File Manager.

## 5.1 IBM File Manager publications

There are four features that the IBM File Manager for z/OS product provides:

- ▶ File Manager Base
- ▶ File Manager for DB2
- ▶ File Manager for IMS
- ▶ File Manager for CICS®

With the current release of the ISPF Productivity Tool, an Object List can invoke only the File Manager Base feature of the IBM File Manager for z/OS product.

This chapter will help you get started with ISPF-PT and File Manager. We do not cover all of the functions that the File Manager provides. For additional information about the File Manager for z/OS product, see the *IBM File Manager for z/OS User's Guide and Reference*, using the following Web address:

<http://www.ibm.com/software/awdtools/filemanager/library/>

You can also review the User's Guide and Reference for additional information about the File Manager Base.

## 5.2 Customizing ISPF-PT to invoke File Manager

You can use the IQIWIZRD command to customize the ISPF Productivity Tool to invoke the File Manager from an Object List.

The following steps guide you with the customization procedure:

1. Logon to ISPF using a valid IPT LOGON procedure.
2. Exit from ISPF, and return to TSO.
3. Enter IQIWIZRD to invoke the IPT Customization Wizard.
4. When prompted, enter the SIQITLIB library name.
5. Follow the installation instructions provided by the ISPF-PT Installation and Customization Guide for the installation wizard.
6. Enter S for VSAM data sets, as shown in Figure 5-1 on page 133.

```

----- IBMPT - Customization Wizard -----
COMMAND ==>

Select the optional object classes to be supported. Internal
object classes are already preselected. If you are unsure, select
all options to get more information about each one. You will be
able to de-select undesired option later.

Select  Object Class                                Notes
-----
S      Sequential data sets                          Built-in IBMPT support
S      PDS and PDSE libraries                        Built-in IBMPT support
S      Open Edition files                            Requires OpenEdition MVS
S      PC files                                       Requires ISPF workstation
-      Panvalet libraries                            Interfaces to third party product
-      Librarian files                              Interfaces to third party product
-      PDSMAN libraries                             Activates built-in support
S      VSAM data sets                                Interfaces to third party product
S      DB2 tables                                    Interfaces to third party product
S      User defined objects                          Define your own interface
S      SCLM support                                  Activates built-in support

Press ENTER to proceed or the END key to return to the initial screen.

```

Figure 5-1 Select the VSAM data set interface

7. Type a 2 for the VSAM interface, as shown in Figure 5-2.

```

----- IBMPT - Customization Wizard -----
COMMAND ==>
When IBMPT detects a VSAM data set (in BROWSE, EDIT, VIEW, DSLIST, or object
list), it may use an interface to process it. You may select which kind
of VSAM support you want in your installation, if any. The table below
shows what CLIST will handle the support you have selected.

      VSAM interface ==> 2 (Select one of the following options)
Option Invoked VSAM interface                                Interface CLIST
-----
0      None (IBMPT will not handle VSAM files.) . . . . . Not applicable
1      Installation-written VSAM interface . . . . . IQIVSAM
2      IBM File Manager for z/OS . . . . . IQI$FMGR
3      File-Aid VSAM browser/editor (For Version 6 or below) . . . IQI$FAID
4      File-Aid VSAM browser/editor (Version 7.0) . . . . . IQI$XPRT
5      File-Aid VSAM browser/editor (Version 8.0 or above) . . . IQI$FAR8
6      Data-Xpert VSAM browser/editor . . . . . IQI$XPRT
7      MacKinney ISPF VSAM Utility . . . . . IQI$MCKN
8      IBM Ditto/ESA . . . . . IQI$DIT
9      Serena STARTOOL . . . . . IQI$STOL
10     VSAM RECORD BROWSER (A SAMPLE WORKING INTERFACE) . . . IQIVBRO

NOTES:- Options 2-9 require a third-party product. If such a product uses
LIBDEFS, include the LIBDEF statements in the interface CLIST.

```

Figure 5-2 Set the VSAM interface to the IBM File Manager for z/OS

8. Complete the ISPF-PT installation Wizard steps.
9. Options 2-9 require a third-party product. If such a product uses LIBDEFS, include the LIBDEF statements in the interface CLIST. For File Manager, add the File Manager library SFMNEXEC to the SYSEXEC allocation, and add the library SFMNSLIB to the ISPSLIB allocation to the ISPF-PT Logon CLIST or ISPF-PT LOGON procedure. See the *ISPF Productivity Facility Installation and Customization Guide* for details.

## 5.3 Using the File Manager with ISPF-PT

After the customization is complete, you can use ISPF-PT to invoke the File Manager from an Object List.

There are several reasons you might want to invoke the File Manager from ISPF-PT:

- ▶ VSAM Browse or Edit
- ▶ File is too large to Edit with ISPF
- ▶ Apply a copybook to File Manager:
  - Visibility to Packed Decimal and Binary fields in readable format
  - Table and Single views
  - Many more functions
- ▶ File Manager supports the following types of VSAM files:
  - ESDS (Entry Sequence Data Set)
  - KSDS (Key Sequence Data Set)
  - RRDS (Relative Record Data Set)
  - Fixed-length or variable-length
  - PATH
  - PATHs related to an alternate index are restricted to browse only
  - AIX® (Alternate Index)

When you use an Object List or Point-and-Shoot command to access a file, ISPF-PT inspects the file type prior to initiating the Browse, View, or Edit operation. If the file is a VSAM file, Browse or View requests automatically invoke the File Manager to present the file in Browse mode. For VSAM files, the Edit requests automatically invoke the File Manager to present the file in Edit mode. Table 5-1 shows how ISPF-PT responds with VSAM and Sequential files when Browse, Edit, or View actions.

Table 5-1 ISPF-PT response with Browse, View, and Edit commands with File Manager

File type	Object List line command	Point-and-Shoot Main Command	Product invoked
VSAM (all types)	B or V	BR or VI	File Manager – Browse
VSAM (all types)		ED	File Manager – Edit
VSAM (all types)	BF or VF	BF or VF	File Manager – Browse
VSAM (all types)	E or EF	ED or EF	File Manager – Edit
Sequential files	B, V, E	BR, VI, or ED	ISPF – Browse, View, or Edit
Sequential files	BF, VF or EF	BF, VF or EF	File Manager browses or edits the file, which is useful if you want to apply a copybook when accessing the file.
Sequential files that are too large for TSO address space memory	V, E	VI or ED	ISPF changes the format to Browse. The file is too large to fit into the TSO Address space memory, so Browse is invoked instead.
Sequential Files too large for TSO address space memory	EF	EF	File Manager will edit the file. Since the file will not fit in the TSO Address space, File Manager will use the “Auxiliary Edit” mode to allow you to edit the file.

**Note:** You can use the File Manager to browse or edit sequential files or PDS members. With File Manager, you can apply a copybook to a sequential file, or you can edit a large sequential file, which ISPF is unable to edit. Use the BF or EF Object List or Point-and-Shoot command to invoke the File Manager in Browse or Edit mode for sequential files.

Table 5-1 on page 134 does not show all ISPF-PT commands. Enter the primary command to see all shortcut commands.

## 5.4 File Manager walkthrough – applying a Copybook to a VSAM file

Figure 5-3 through Figure 5-11 on page 139 show you how to use File Manager with ISPF-PT. An action of E or EF edits the DNET424.IPT.CUSTFILE using File Manager from an ISPF-PT Object List, as shown in Figure 5-3.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (B) ----- LEVEL DNET424.IPT ----- Row 1 to 3 of 3
Command ===>          SCROLL ===> CSR
Hotbar: OPRINT  REFRESH  CLRVOL  FILLVOL  UTIL  UPDATE  CUT  FLIP
                                         *TEMPORARY LIST*

TSO PARMS ===>
Command  Member  Numbr Data Set Names / Objects  Class
-----
e 1 'DNET424.IPT.CUSTFILE'                      VSAM
  2 'DNET424.IPT.CUSTFILE.DATA'                  VSAMDA
  3 'DNET424.IPT.CUSTFILE.INDEX'                  VSAMIX
----- END OF LIST -----

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

```

Figure 5-3 Edit a VSAM file from the IPT Object List

With the panel shown in Figure 5-4 on page 136, you can insert, record, delete, or repeat records by using the I, D, or R line commands or block commands. You can see the data in hexadecimal format by entering HEX ON. The File Manager supports most ISPF commands.

```

Process Options Help
Edit DNET424.IPT.CUSTFILE Rec 0 of 30
Command ==> te Scroll PAGE
Type KSDS Format CHAR
Col 1 Insert length 2048
<====1==>-2-----3-----4-----5-----6-----7--
000000 **** Top of data ****
000001 01001C Lynn, Amanda .....Spirit Lake Musician
000002 02200C Graham, Anna .../.*..Atwon Cryptographer
000003 02202C Major, Art ....%..Harmon College student
000004 03003C Prentice, Anna .....Laramie New hire
000005 03390C Deeds, Darren ... ..Sandstone Air rescue pilot
000006 05500C Parker, Ford ...@..Dearborn Garage attendant
000007 06101C Early, Brighton .....Buxford Breakfast chef
000008 06106C Lander, Annette ...q<..Taledega Trapeze artist
000009 06711C Dubree, Dustin .....Indianapolis Demolition Contracto
000010 06900C Bacon, Chris P. ....Lisle Chef
000011 07008C Houston, Roger ...@..Banner Elk NASA Liaison
000012 07044C Schauer, April ...h...Naples Wildflower collector
000013 07077C Mann, Mr. E. ....*..Danville Private Investigator
000014 07707C Clime, Hilda .....Dowagiac Mountain climber
F1=Help F2=Zoom F3=Exit F4=CRetrieiv F5=RFind F6=RChange
F7=Up F8=Down F9=Swap F10=Left F11=Right F12=Retrieve

```

Figure 5-4 File Manager Editor without a Copybook

The TE primary command, shown in Figure 5-4, invokes the File Manager Template Workbench, as shown in Figure 5-5.

```

Process Options Help
File Manager Template Workbench
Command ==> rc More: +
RC Run using copybook RT Run using template
CC Create template from copybook E Edit field/record in template
CM Create template from model U Update template from copybook
MC Map from copybook MT Map from template

Copybook:
Data set name . 'DNET424.ADLAB.COPYLIB'
Member . . . . CUST2
Template:
Data set name .
Member . . . .
Model Template:
Data set name .
Member . . . .

Processing Options:
Enter "/" to select option
F1=Help F2=Split F3=Exit F4=CRetrieiv F5=RT F6=Describe
F7=Backward F8=Forward F9=Swap F10=Actions F12=Cancel

```

Figure 5-5 File Manager Template Workbench

Enter the copy library and copybook member name that is associated with the file. Enter RC to apply the selected copybook to your VSAM file. Figure 5-6 on page 137 depicts the file using the copybook in Figure 5-5.

Process	Options	Help
Edit	DNET424.IPT.CUSTFILE	Rec 0 of 30
Command ==>	fe	Scroll PAGE
		Format TABL
CUST-ID	RECORD-TYPE	FILLER
#3	#4	#5
AN 1:5	AN 6:1	AN 7:7
<--->	-	<-----1----->
000000	****	Top of data
000001	01001	C
000002	02200	C
000003	02202	C
000004	03003	C
000005	03390	C
000006	05500	C
000007	06101	C
000008	06106	C
000009	06711	C
000010	06900	C
000011	07008	C
000012	07044	C
F1=Help	F2=Zoom	F3=Exit
F7=Up	F8=Down	F9=Swap

Figure 5-6 File Manager Edit with a Copybook

With a copybook, we can see the field names, the field type (AN – Alphanumeric, PD – Packed Decimal, and BI – Binary), and a #n. The numbers preceded by “#” are *field reference numbers*. These numbers identify the field in **Find** or **Change** commands, for example, the command “**FIND 10 #8**” limits the search for the value “10” in the ORDERS-YTD field. A special File Manager command, FE or Field in Error, locates invalid data in Packed Decimal Fields. Figure 5-7 shows you an invalid Packed Decimal Field.

Process	Options	Help
Edit	DNET424.IPT.CUSTFILE	1 errors found
Command ==>	hex	Scroll PAGE
		Format TABL
CUST-ID	RECORD-TYPE	FILLER
#3	#4	#5
AN 1:5	AN 6:1	AN 7:7
<--->	-	<-----1----->
000029	23004	C
000030	24090	C
000031	****	End of data
F1=Help	F2=Zoom	F3=Exit
F7=Up	F8=Down	F9=Swap

Figure 5-7 Invalid Packed Decimal Data identified by File Manager

In Figure 5-7, you can overwrite the red asterisk field to correct the data. However, if you want to see the contents of the field, enter the HEX command. Figure 5-8 on page 138 shows the data in hexadecimal format using the HEX command.

Process Options Help							
Edit DNET424.IPT.CUSTFILE				Rec 29 of 30			
Command ==>				Scroll PAGE			
				Type KSDS		Format TABL	
CUST-ID	RECORD-TYPE	FILLER	NAME	ACCT-BALANCE	ORDERS-YTD		
#3	#4	#5	#6	#7	#8		
AN 1:5	AN 6:1	AN 7:7	AN 14:17	PD 31:5	BI 36:2		
<--->	-	<---+>	<-----1----->	<-----1>	<---+>		
000029	23004	C	Furst, Hugo	*****	1		
	FFFFF	C	4444444	CA9AA64CA89444444	0023A	00	
	23004	3	0000000	64923B08476000000	0033C	01	
000030	24090	C	Downe, Pat	399.96	2		
	FFFFF	C	4444444	C9A9864D8A4444444	00396	00	
	24090	3	0000000	46655B07130000000	0099C	02	
000031	**** End of data ****						
F1=Help	F2=Zoom	F3=Exit	F4=CRetrie v	F5=RFind	F6=RChange		
F7=Up	F8=Down	F9=Swap	F10=Left	F11=Right	F12=Retrieve		

Figure 5-8 Invalid Packed Decimal data in Hexadecimal form

Figure 5-9 shows the CUSTFILE in table format. Each line represents a record. The PF11 key will navigate to the right. You can also use the **Locate** command to locate field names. We can also choose the Format Single mode by entering the FS command on a specific record, as shown in Figure 5-9.

Process Options Help							
Edit DNET424.IPT.CUSTFILE				Rec 0 of 30			
Command ==>				Scroll PAGE			
CUST-ID	RECORD-TYPE	FILLER	NAME	Type KSDS	ACCT-BALANCE	ORDERS-YTD	Format TABL
#3	#4	#5	#6	#7	#8		
AN 1:5	AN 6:1	AN 7:7	AN 14:17	PD 31:5	BI 36:2		
<--->	-	<---+>	<-----1----->	<-----1>	<---+>		
000000	**** Top of data ****						
000001	01001	C	Lynn, Amanda	67.68	9		
000002	02200	C	Graham, Anna	610.05	10		
000003	02202	C	Major, Art	1234.56	5		
000004	03003	C	Prentice, Anna	396.63	7		
000005	03390	C	Deeds, Darren	74.00	3		
000006	05500	C	Parker, Ford	233.27	5		
000007	06101	C	Early, Brighton	311.08	10		
FS	06106	C	Lander, Annette	489.84	7		
000009	06711	C	Dubree, Dustin	192.98	1		
000010	06900	C	Bacon, Chris P.	1001.01	0		
000011	07008	C	Houston, Roger	296.97	10		
000012	07044	C	Schauer, April	88.83	7		
F1=Help F2=Zoom F3=Exit F4=CRetrie v F5=RFind F6=RChange F7=Up F8=Down F9=Swap F10=Left F11=Right F12=Retrieve							

Figure 5-9 File Manager Table Format – Request for Single Format



In Figure 5-10, the record is shown using *Single Format mode*.

Process	Options	Help
Edit	DNET424.IPT.CUSTFILE	Rec 9 of 30
Command ==>	sloc;type;ref	Scroll PAGE
	Type KSDS	Format <b>SNGL</b>
	Top Line is 1	of 8
Current 01: CUSTOMER-RECORD		
Field	Data	
CUST-ID	06711	
RECORD-TYPE	C	
FILLER		
NAME	Dubree, Dustin	
ACCT-BALANCE	192.98	
ORDERS-YTD	1	
CITY	Indianapolis	
OCCUPATION	Demolition Contractor	
*** End of record ***		
F1=Help	F2=Zoom	F3=Exit
F7=Up	F8=Down	F9=Swap
F4=CRetrie	F5=RFind	F6=RChange
F10=Previous	F11=Next	F12=Cancel

Figure 5-10 File Manager using "Format Single" mode

The commands SLOC, TYPE, and REF add the Starting location, length, type, and reference numbers to the panel, as shown in Figure 5-11.

Process	Options	Help	
Edit	DNET424.IPT.CUSTFILE	Reference numbers on	
Command ==>		Scroll PAGE	
	Type KSDS	Format <b>SNGL</b>	
	Top Line is 1	of 8	
Current 01: CUSTOMER-RECORD			
Ref Field	Typ	Start Len Data	
3 CUST-ID	AN	1 5	06711
4 RECORD-TYPE	AN	6 1	C
5 FILLER	AN	7 7	
6 NAME	AN	14 17	Dubree, Dustin
7 ACCT-BALANCE	PD	31 5	192.98
8 ORDERS-YTD	BI	36 2	1
9 CITY	AN	38 15	Indianapolis
10 OCCUPATION	AN	53 28	Demolition Contractor
*** End of record ***			
F1=Help	F2=Zoom	F3=Exit	
F7=Up	F8=Down	F9=Swap	
F4=CRetrie	F5=RFind	F6=RChange	
F10=Previous	F11=Next	F12=Cancel	

Figure 5-11 File Manager using Format Single mode with SLOC, TYPE, and REF

Archived



## **SCLM integration with the ISPF Productivity Tool**

The ISPF Productivity Tool (ISPF-PT) interfaces with the IBM Software Configuration and Library Manager (SCLM) to provide you with all of the functionality of ISPF-PT with frequently used SCLM functions.

## 6.1 Customization to support SCLM

SCLM requires more information when creating members, such as lock information, change codes, and alternate project libraries. To minimize the collection of SCLM information for non-SCLM users, the ISPF-PT customization wizard provides the following SCLM customization options:

- ▶ All users are SCLM users
- ▶ There are no SCLM users
- ▶ Users are identified by an IQISCLM DD DUMMY statement in their LOGON procedure.

Figure 6-1 is the beginning of the ISPF-PT customization wizard SCLM windows.

```
----- IBMPT - Customization Wizard -----
COMMAND ==>

Select the optional object classes to be supported. Internal
object classes are already preselected. If you are unsure, select
all options to get more information about each one. You will be
able to de-select undesired option later.

Select  Object Class                      Notes
-----
S       Sequential data sets              Built-in IBMPT support
S       PDS and PDSE libraries            Built-in IBMPT support
S       Open Edition files                Requires OpenEdition MVS
S       PC files                          Requires ISPF workstation
=       Panvalet libraries                Interfaces to third party product
-       Librarian files                   Interfaces to third party product
-       PDSMAN libraries                  Activates built-in support
S       VSAM data sets                    Interfaces to third party product
S       DB2 tables                        Interfaces to third party product
S       User defined objects              Define your own interface
S       SCLM support                      Activates built-in support

Press ENTER to proceed or the END key to return to the initial screen.
```

Figure 6-1 ISPF-PT customization wizard - Activate SCLM support

Enter S, and Figure 6-2 is displayed to provide SCLM support through ISPF-PT.

```
----- IBMPT - Customization Wizard -----
COMMAND ==>

SCLM is ISPF's Software Configuration and Library Manager. SCLM provides
controlled edit capabilities within standard PDS libraries (with member locking,
check-in/check-out), full development life cycle of applications (including
a "MAKE"), and extensive reporting. SCLM is an integrated part of ISPF.

IBMPT can provide significant benefits to SCLM users by integrating SCLM
support into the standard member list, DSLIST and OLIST. If your installation
uses SCLM, it is recommended that you activate the SCLM support on this screen.

Activate SCLM support ==> D Y =Yes - Support SCLM for all users
                        N =No - Do not provide SCLM support
                        D =Conditional support. SCLM will only be
                          supported for users who at the time IBMPT is
                          started, have a preallocated DD "IQISCLM".
                          This DD can be allocated in the LOGON PROC/CLIST.
                          Example: ALLOC DD(IQISCLM) DUMMY
                                  or //IQISCLM DD DUMMY

Press ENTER to proceed or the END key to return to the initial screen.
```

Figure 6-2 ISPF-PT customization wizard - Activate SCLM support

The installation provides conditional SCLM support. Your ISPF-PT session supports the SCLM commands, provided you have the DD name, IQISCLM, allocated to your LOGON procedure.

**Hint:** If your installation provided conditional SCLM support and your LOGON procedure does not have the //IQISCLM DD DUMMY statement, or CLIST does not have ALLOC DD(IQISCLM) DUMMY, you can dynamically add it. Exit from ISPF, and enter the command ALLOC DD(IQISCLM) DUMMY. Invoke ISPF, and your session will be SCLM enabled.

The SCLM parameter prompt in Figure 6-3 refers to the SCLMPARM window in Figure 6-4.

```

----- IBMIPT - Customization Wizard -----
COMMAND ==>
Verify SCLM options:
SCLM parameter prompt ==> F Y =Yes - Whenever an SCLM library is accessed.
                          N =No - Users must preset the parameters with
                          the SCLMPARM command (in the Member List).
                          F =A prompt screen is displays only on first time
                          access to a specific library. Thereafter the
                          previous parameters for that library are used.

Note:the above option can be changed by individual users via the SET command.

The folowing option applies to ISPF Version 4.1 or above:
Override SCLMCHK value ==> * (W=WARN, E=ERROR, N=NONE, *=no override)
This setting allows you to override the value of the "SCLMCHK" field in
ISRCNFG (ISPF's configuration table). This field defines how edit should
react when attempting to edit an SCLM-saved file without SCLM locking.

SCLMCHK values (See ISPF Customization for more details):
WARN  -Display a warning message if edited project/member is SCLM controlled.
NONE  -No checking is done. No warning messages issued.
ERROR -Reject editing of SCLM-saved members without SCLM or IBMIPT locking.

Press ENTER to proceed or the END key to return to the initial screen.

```

Figure 6-3 ISPF-PT customization wizard – Activate SCLM support

The Override SCLMCHK provides a way to override the SCLM locking. We recommend that you set this value to “\*”, providing that there are no SCLM lock overrides.

The recommended setting is “F”, which will display the SCLMPARM window in Figure 6-5 on page 144 when an SCLM library is first accessed.

```

                                          -IPT-
Command    ==>
GROUPS: 1=DEVI025  2=INT420  3=TEST420  4=RLSE420  5=
Project: SOFPAUD  Group: DEVI025  Type: ASM
Specify/verify SCLM parameter for this library:
Lock       ==> YES      (Yes, No)
Alternate  ==>          (Alternate project name)
Change code ==>          (Default language)
Language   ==>          (Default language)
Auth. code ==>          (Default language)
F1=HELP    F2=SPLIT    F3=END      F4=VIEW    F5=RFIND    F6=RCHANGE
F7=UP      F8=DOWN     F9=SWAP     F10=LEFT  F11=RIGHT

```

Figure 6-4 Initial SCLMPARM window

You can set the following values:

- ▶ Lock: "Yes" will lock members from other SCLM users.
- ▶ Alternate: SCLM macros define the characteristics of the SCLM hierarchy. The library hlq.PROJDEFS.LOAD contains the assembled and linked SCLM macro definitions. If you want to override the default member name of the SCLM definition, you can use a different (alternate) SCLM member, specified here.
- ▶ Change Code: You can specify the default Change Code.
- ▶ Language: You can specify the default language.
- ▶ Authorization Code: You can specify the default Authorization Code.

The fields that the SCLMPARM command defines, apply only to the SCLM library currently edited by ISPF-PT, for example, you can set SCLMPARM values for the library SOFTAUD.DEVI205.ASM to the language HLASM, and the SOFTAUD.DEVI205.JCL library can have the language TEXT.

**Hint:** The SCLMPARM provides the *default* values when you add new members to an SCLM library. You can override these parameters using the SPROF command.

You can override the default installation values of Lock, and Display SCLM parameters for your ISPF sessions using the ISET parameter. Select the SCLM options shown in Figure 6-5.

```
-IPT- -----Setting IBMIPT Defaults-----
COMMAND ==>
Select options by number, name, with cursor selection, or with line commands:
  IBMIPT is running under ISPF version 5.7

- A - ALL          - Select all the below displayed options
- M - MSL          - Member Selection List options
- P - OLIST        - Object list options
- G - GLOBAL       - Global edit and Findtext options
- R - PRINT        - Print options
- D - DSLIST       - DSLIST options
- T - TSO          - TSO shell options
- E - EDIT         - Edit, Browse and View options
- I - INTERFACE    - Specify user interface options
- S - SCLM         - SCLM options
- N - DIAGNOSE     - Diagnose ISPF errors
- L - LIBRARY      - Persistent table library options

Make your selection and press the ENTER key or press the END key to exit
```

Figure 6-5 User SCLM options

You can override the LOCK default for all SCLM libraries for your ID, as shown in Figure 6-6.

```
-IPT- -----SCLM Options-----
COMMAND ==>

Verify or specify the following options:

Edit locks members in SCLM controlled libraries ==> Y (Y=Yes, N=No)
Display SCLM parameters before accessing library ==> F (Y=Yes, N=No,
F=First time only)

Note: When the F option (first time) is active, IBMIPT will try to find the
previously specified parameters for the current combination of PROJECT
and TYPE. If none are found, a prompt window is displayed. As IBMIPT
"learns" about these parameters, the prompt window will be displayed
less frequently.

Press ENTER for options menu, END to exit, CANCEL for installation defaults.
```

Figure 6-6 SCLM Options panel

## 6.2 SCLM ISPF-PT interface

Figure 6-7 defines the SCLM hierarchy that we used in the examples in this section.

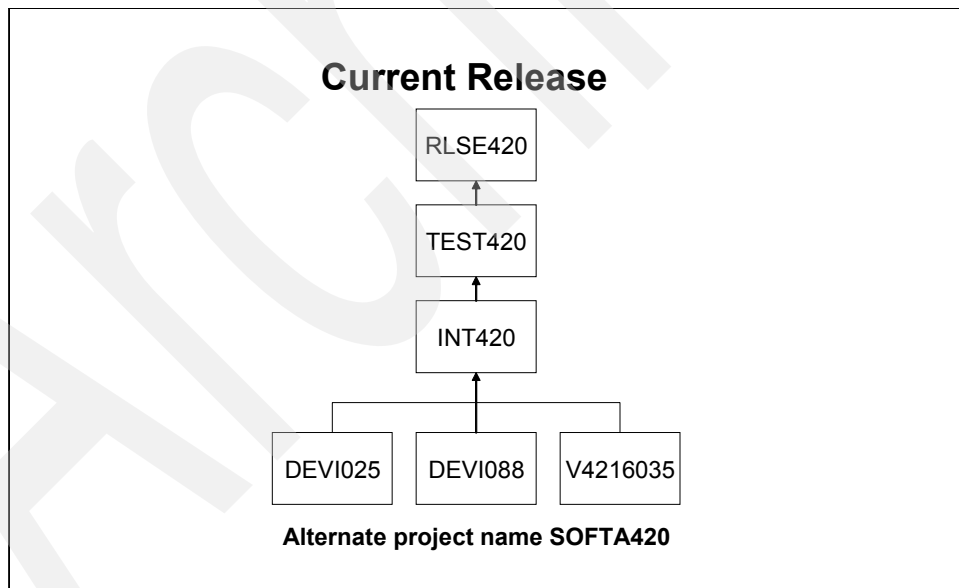


Figure 6-7 SCLM hierarchy

The Alternate project name SOFTA420 defines the SCLM hierarchy shown in Figure 6-7. Each level contains a number of data sets, such as SOFTAUD.DEVI025.ASM, LOAD, ARCHDEF, and so on.

The Object List '<' character, shown on line 6 of Figure 6-8, denotes an SCLM library. Enter the lowest level library in the SCLM hierarchy.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (B) ----- Personal list ----- Row 5 to 13 of 13
Command ==> 6 e SCROLL ==> CSR
Hotbar: REFRESH FLIP UTIL FILLVOL CLRVOL SET CUT UPDATE
Open list ==> MYLIST (or BLANK for reference list)
TSO PARMS ==>
Command Member Numbr Data Set Names / Objects Class
-----
5 |----- SOFTAUD LIBRARIES -----|
6 <SOFTAUD.DEVI025.ASM SCLM
7 =SOFTAUD.DEVI025.INT420 TEST420 RLSE420 ASM ISPF
8 'SOFTAUD.DEVI025*' LIST
9 'SOFTAUD.INT420*' LIST
10 'SOFTAUD.TEST420*' LIST
11 'SOFTAUD.RLSE420*' LIST
12 :LISTW SOFTAUD OLIST
13 'SOFTAUD.PROJ*' LIST
----- END OF LIST -----

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

```

Figure 6-8 Object List - SCLM Libraries

In Figure 6-9, ISPF-PT found the three libraries circled from the SCLM project definition, using the alternate project definition member SOFTA420.

```

File Display Library SCLM Settings Menu Utilities Test Help Exit
-----
-IPT--EDIT L1----- SOFTAUD.DEVI025.ASM ----- "A" will display assist
COMMAND ==> sclmparm SCROLL ==> CSR
HOTBAR: REFRESH FLIP SORT CHA TAILOR COMPRESS EXPDIR INFO SET
LIB2=INT420 LIB3=TEST420 LIB4=RLSE420
NAME RENAME LIB VV.MM CREATED CHANGED SIZE INIT MOD USERID
=NOTE>Enter the "SCLMPARM" command to change the following defaults:
=NOTE>Language:HLASM Change code:V4R2M0 Auth code:
=NOTE>Using alternate project SOFTA420
$AUDFORM S 4 01.10 96/02/26 06/04/24 12:50 97 89 0 INTT065
$AUDFSTK S 4 01.00 93/05/04 99/05/12 12:06 18 18 0 V20SHIP
$AUDGSTK S 4 01.07 93/05/04 02/06/11 21:20 61 29 0 INTT025
$AUDINIT S 4 01.05 95/03/14 03/12/04 11:45 66 38 0 INTT021
$CALL S 4 01.09 98/08/24 99/05/12 12:06 71 59 0 INTT046
$CKSAVE S 4 01.12 93/10/13 99/05/12 12:06 41 25 0 INTT022
$CKSAVE1 S 4 01.11 06/08/16 06/08/25 15:54 59 41 0 INTT068
$ENDMSG S 4 01.07 93/10/13 99/12/01 00:52 69 17 0 INTT022
$GENBRK S 4 01.00 03/12/01 03/12/01 15:51 28 204 0 INTT021
$GENFLD S 4 01.00 03/12/01 03/12/01 15:56 63 204 0 INTT021
$GENLEV S 4 01.00 03/12/01 03/12/01 15:46 41 204 0 INTT021
$GENPRN S 4 01.00 03/12/01 03/12/01 15:47 32 204 0 INTT021
F1=HELP F2=SPLIT F3=END F4=VIEW F5=RFIND F6=RCHANGE
F7=UP F8=DOWN F9=SWAP F10=LEFT F11=RIGHT F12=RETRIEVE

```

Figure 6-9 SCLM Edit using an MSL

The SCLMPARM command shows the window in Figure 6-10 on page 147.



```

- IPT -
Command      ==>
GROUPS: 1=DEV1025 2=INT420 3=TEST420 4=RLSE420 5=
Project: SOF AUD Group: DEV1025 Type: ASM
Specify/verify SCLM parameter for this library:
Lock         ==> YES (Yes, No)
Alternate    ==> SOF TA420 (Alternate project name)
Change code  ==> V4R2M0
Language     ==> HLASM (Default language)
Auth. code   ==>
F1=HELP      F2=SPLIT    F3=END      F4=VIEW      F5=RFIND     F6=RCHANGE
F7=UP        F8=DOWN     F9=SWAP     F10=LEFT     F11=RIGHT

```

Figure 6-10 SCLMPARM window populated with data

The SCLMPARM window provides the default values that SCLM uses for new members. You might need to update the alternate project name to direct SCLM to the correct libraries. The examples in this chapter use the data shown in Figure 6-10.

In Figure 6-11, the DEF E action sets the default action to EDIT. The 7 initiates an edit session with concatenated libraries. ISPF-PT classified the data sets on line 7 as ISPF concatenated libraries.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
- IPT- OLIST (B) ----- Personal list ----- Row 5 to 13 of 13
Command      ==> def e; 7
Hotbar: REFRESH FLIP UTIL FILLVOL CLRVOL SET CUT UPDATE
Open list    ==> MYLIST (or BLANK for reference list)
TSO PARMS    ==>
Command      Member Numbr Data Set Names / Objects Class
-----
5 |----- SOF AUD LIBRARIES -----
6 <SOF AUD.DEV1025.ASM SCLM
7 =SOF AUD DEV1025 INT420 TEST420 RLSE420 ASM ISPF
8 'SOF AUD.DEV1025*' LIST
9 'SOF AUD.INT420*' LIST
10 'SOF AUD.TEST420*' LIST
11 'SOF AUD.RLSE420*' LIST
12 :LISTW SOF AUD OLIST
13 'SOF AUD.PROJ*' LIST
----- END OF LIST -----
F1=HELP      F2=SPLIT    F3=END      F4=VIEW      F5=RFIND     F6=RCHANGE
F7=UP        F8=DOWN     F9=SWAP     F10=LEFT     F11=RIGHT    F12=RETRIEVE

```

Figure 6-11 ISPF concatenated library – Used as SCLM libraries

Both lines 6 and line 7 in Figure 6-11 yield the same results. ISPF-PT determined that these libraries are SCLM libraries by the S in the lib field, which we circled in Figure 6-12 on page 148.

```

File Display Library SCLM Settings Menu Utilities Test Help Exit
-----
-IPT--EDIT L1----- SOF AUD.DEVI025.ASM ----- "A" will display assist
COMMAND ===>
HOTBAR: REFRESH FLIP SORT CHA TAILOR COMPRESS EXPDIR INFO SET
LIB2=INT420 LIB3=TEST420 LIB4=RLSE420
NAME RENAME LIB VV.MM CREATED CHANGED SIZE INIT MOD USERID
=NOTE>Enter the "SCLMPARM" command to change the following defaults:
=NOTE>Language:HLASM Change code:V4R2M0 Auth code:
=NOTE>Using alternate project SOFTA420
$AUDFORM S 4 01.10 96/02/26 06/04/24 12:50 97 89 0 INTT065
$AUDFSTK S 4 01.00 93/05/04 99/05/12 12:06 18 18 0 V20SHIP
$AUDGSTK S 4 01.07 93/05/04 02/06/11 21:20 61 29 0 INTT025
$AUDINIT S 4 01.05 95/03/14 03/12/04 11:45 66 38 0 INTT021
$CALL S 4 01.09 98/08/24 99/05/12 12:06 71 59 0 INTT046
$CKSAVE S 4 01.12 93/10/13 99/05/12 12:06 41 25 0 INTT022
$CKSAVE1 S 4 01.11 06/08/16 06/08/25 15:54 59 41 0 INTT068
$ENDMSG S 4 01.07 93/10/13 99/12/01 00:52 69 17 0 INTT022
$GENBRK S 4 01.00 03/12/01 03/12/01 15:51 28 204 0 INTT021
$GENFLD S 4 01.00 03/12/01 03/12/01 15:56 63 204 0 INTT021
$GENLEV S 4 01.00 03/12/01 03/12/01 15:46 41 204 0 INTT021
$GENPRN S 4 01.00 03/12/01 03/12/01 15:47 32 204 0 INTT021
F1=HELP F2=SPLIT F3=END F4=VIEW F5=RFIND F6=RCHANGE
F7=UP F8=DOWN F9=SWAP F10=LEFT F11=RIGHT F12=RETRIEVE

```

Figure 6-12 SCLM edit using an MSL

If your session is SCLM enabled and SCLM libraries are used, ISPF-PT invokes the SCLM support. These functions include:

- ▶ SCLMPARM command
- ▶ K command
- ▶ Notes, as shown in Figure 6-12
- ▶ SPROF command
- ▶ Many more functions

In Figure 6-13, the libraries that are specified in line 7 do not match the SCLM hierarchy. The INT420 library is missing.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (E) ----- Personal list ----- Row 5 to 14 of 14
Command ===> 7 SCROLL ==> CSR
Hotbar: REFRESH FLIP UTIL FILLVOL CLRVOL SET CUT UPDATE
Open list ==> MYLIST (or BLANK for reference list)
TSO PARMS ==>
Command Member Numbr Data Set Names / Objects Class
-----
5 |----- LCM2 SUPPORT -----
6 <SOF AUD.DEVI025.ASM SCLM
7 =SOF AUD.DEVI025 TEST420 RLSE420 ASM ISPF
8 'SOF AUD.DEVI025*' LIST
9 'SOF AUD.INT420*' LIST
10 'SOF AUD.TEST420*' LIST
11 'SOF AUD.RLSE420*' LIST
12 :LISTW SOF AUD OLIST
13 'SOF AUD.PROJ*' LIST
14 JCL
----- END OF LIST -----
F1=HELP F2=SPLIT F3=END F4=VIEW F5=RFIND F6=RCHANGE
F7=UP F8=DOWN F9=SWAP F10=LEFT F11=RIGHT F12=RETRIEVE

```

Figure 6-13 ISPF concatenated library – Used as SCLM libraries

ISPF-PT notifies you if the libraries in the ISPF concatenation do not match the SCLM hierarchy. If a library is incorrect or missing or if the alternate project definition is incorrect, you will see the window in Figure 6-14. Choose option 1 or 2 to continue.

```

-IPT--L1 ----- EDIT - CONFIRM PANEL -----
OPTION ==> 1
IQIM145  CORRECT SCLM CONCATENATION SHOULD HAVE TEST420 INSTEAD OF RLSE420

The concatenation order you specified does not match the SCLM hierarchy.

Select one of the following options and press ENTER or press END to cancel.
 1 - Use the correct SCLM hierarchy.
 2 - Use the groups you specified
+-----+
| ISPF LIBRARY:                                     |
| Project      ==> SOFTAUD                           |
| Specified Group ==> DEVI025   ==> TEST420   ==> RLSE420   ==> |
| SCLM Hierarchy ==> DEVI025   ==> INT420     ==> TEST420   ==> RLSE420   |
| Type         ==> ASM                               |
+-----+

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

```

Figure 6-14 ISPF-PT SCLM Confirmation window

## 6.3 ISPF-PT SCLM walk through

In this section, which includes Figure 6-15 on page 150 through Figure 6-51 on page 167, we use ISPF-PT and SCLM to show you how to:

- ▶ Create a new assembler program
- ▶ Migrate members to SCLM
- ▶ Update the project SCLM ARCHDEF members
- ▶ Build and Promote the SCLM ARCHDEF members
- ▶ Remove obsolete program

### 6.3.1 Creating a new assembler program

In this section, we show you how to create a new assembler program and how to assemble it.

1. Edit the SCLM hierarchy shown Figure 6-15 on page 150.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (E) ----- Personal list ----- Row 1 to 14 of 14
Command ==> 6 SCROLL ==> CSR
Hotbar: REFRESH FLIP UTIL FILLVOL CLRVOL SET CUT UPDATE
Open list ==> MYLIST (or BLANK for reference list)
TSO PARMS ==>
Command Member Numbr Data Set Names / Objects Class
-----
1 |----- PERSONAL LIBRARIES -----
2 'INTT125.CLIST'
3 'INTT125*' LIST
4 JCL
5 |----- LCM2 SUPPORT -----
6 <SOFTAUD.DEVI025.ASM SCLM
7 =SOFTAUD DEVI025 TEST420 RLSE420 ISPF
8 'SOFTAUD.DEVI025*' LIST
9 'SOFTAUD.INT420*' LIST
10 'SOFTAUD.TEST420*' LIST
11 'SOFTAUD.RLSE420*' LIST
12 :LISTW SOFTAUD OLIST
13 'SOFTAUD.PROJ*' LIST
14 JCL
----- END OF LIST -----

```

Figure 6-15 ISPF-PT SCLM walk through

One way to locate the items in DEVI025 and INT420 is to use the SORT LIB statement, as shown in Figure 6-16.

```

File Display Library SCLM Settings Menu Utilities Test Help Exit
-----
-IPT--EDIT L1----- SOFTAUD.DEVI025.ASM ----- "A" will display assist
COMMAND ==> sort lib SCROLL ==> CSR
HOTBAR: REFRESH FLIP SORT CHA TAILOR COMPRESS EXPDIR INFO SET
LIB2=INT420 LIB3=TEST420 LIB4=RLSE420
NAME RENAME LIB VV.MM CREATED CHANGED SIZE INIT MOD USERID
=NOTE>Enter the "SCLMPARM" command to change the following defaults:
=NOTE>Language:HLASM Change code:V4R2M0 Auth code:
=NOTE>Using alternate project SOFTA420
$AUDFORM S 4 01.10 96/02/26 06/04/24 12:50 97 89 0 INTT065
$AUDFSTK S 4 01.00 93/05/04 99/05/12 12:06 18 18 0 V20SHIP
$AUDGSTK S 4 01.07 93/05/04 02/06/11 21:20 61 29 0 INTT025
$AUDINIT S 4 01.05 95/03/14 03/12/04 11:45 66 38 0 INTT021
$CALL S 4 01.09 98/08/24 99/05/12 12:06 71 59 0 INTT046
$CKSAVE S 4 01.12 93/10/13 99/05/12 12:06 41 25 0 INTT022
$CKSAVE1 S 4 01.11 06/08/16 06/08/25 15:54 59 41 0 INTT068
$ENDMSG S 4 01.07 93/10/13 99/12/01 00:52 69 17 0 INTT022
$GENBRK S 4 01.00 03/12/01 03/12/01 15:51 28 204 0 INTT021
$GENFLD S 4 01.00 03/12/01 03/12/01 15:56 63 204 0 INTT021
$GENLEV S 4 01.00 03/12/01 03/12/01 15:46 41 204 0 INTT021
$GENPRN S 4 01.00 03/12/01 03/12/01 15:47 32 204 0 INTT021
$GENRGN S 4 01.00 03/12/01 03/12/03 14:24 32 204 0 INTT021
$HASHLIB S 4 01.02 93/05/04 99/05/12 12:06 38 25 0 INTT025

```

Figure 6-16 ISPF-PT SCLM walk through

2. Select the program TSTSVC26, as shown in Figure 6-17 on page 151. We are going to change the name of this module to TSTSVC30, which we demonstrate in this walk through.

File Display Library SCLM Settings Menu Utilities Test Help Exit									
-----									
-IPT--EDIT L1----- SOF AUD.DEVI025.ASM -----					-ROW 00001 OF 00343-				
COMMAND ==>					SCROLL ==> CSR				
HOTBAR: REFRESH FLIP SORT CHA TAILOR COMPRESS EXPDIR INFO SET									
*SORT*					LIB2=INT420		LIB3=TEST420		LIB4=RLSE420
NAME	RENAME	LIB	VV.MM	CREATED	CHANGED	SIZE	INIT	MOD	USERID
TESTLONG		S 1	01.02	04/08/19	07/09/06 09:46	176	150	0	INTT125
TESTSCST		S 1	01.01	03/06/24	03/06/25 12:57	430	429	0	INTT025
S TSTSVC26		S 2	01.01	07/09/06	07/09/06 10:28	84	84	0	INTT125
AUCMBASE		S 3	01.01	07/03/15	07/03/26 08:56	861	861	0	INTT070
AUCMSENU		S 3	01.01	07/03/15	07/03/26 08:56	2082	2082	0	INTT070
AUDCMPID		S 3	02.45	01/05/29	07/08/30 16:54	4762	3918	0	INTT068
AUDCMPKB		S 3	01.94	01/10/02	07/08/30 16:55	4556	3358	0	INTT068
AUDCMPSK		S 3	01.42	01/08/28	07/08/30 16:56	3662	3620	0	INTT068
AUDCROSS		S 3	01.99	93/05/04	07/05/10 14:35	868	847	0	INTT011
AUDDIMON		S 3	01.99	01/12/09	07/06/22 12:00	1719	1605	0	INTT068
AUDDIPTB		S 3	01.99	01/12/10	07/08/21 09:22	2767	1847	0	INTT068
AUDDIST		S 3	02.99	97/10/28	07/08/30 17:00	2456	593	0	INTT068
AUDDIUNI		S 3	01.62	06/05/03	07/03/06 10:23	262	126	0	INTT068
AUDDIXML		S 3	01.99	06/05/30	07/08/21 11:23	2089	1306	0	INTT068
AUDEXPBM		S 3	04.05	04/01/05	06/11/17 11:51	2934	1282	0	INTT070
AUDFBOIL		S 3	03.64	99/10/06	07/04/25 15:00	3918	2363	0	INTT065
AUDFILTM		S 3	02.99	97/12/18	07/08/30 17:04	5936	3858	0	INTT068

Figure 6-17 ISPF-PT SCLM walk through

Figure 6-18 demonstrates the SCLM command SPROF.

```
File Edit Settings Build SCLM Menu Utilities Compilers Test Help

-IPT- EDIT SOF AUD.DEVI025.ASM(TSTSVC26) - 01.01          Columns 00001 00072
Command ==> sprof                                         Scroll ==> CSR
***** Top of Data *****
000001 TSTSVC26 TITLE 'Test LOCATE SVC 26'
000002 *****
000003 *      Execution JCL
000004 *
000005 *      =====
000006 * //      EXEC PGM=IPIUTIL,PARM='<dsname>'
000007 *****
000008 *
000009 *      *****
000010 *      * PROLOG
000011 *      *****
000012 TSTSVC26 RMODE ANY
000013 TSTSVC26 AMODE 31
000014 TSTSVC26 CSECT ,
000015         SAVE  (14,12),,*
000016         LR    R12,R15
000017         USING TSTSVC26,R12

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

Figure 6-18 ISPF-PT SCLM walk through

The SPROF command, which we show in Figure 6-19 on page 152, provides the actual language and change code for the member. The SCLMPARM window, Figure 6-19 on page 152, provides the default values that are used for new members.

```

-IPT-
Command ==> _____

Specify desired values then press the ENTER key.
or press the END key to leave values unchanged.
Language ==> HLASM
Change code ==> V4R2M0 (Use "=" to retrieve
                        last entry)

File:SOFTAUD.DEVI025.ASM(TSTSVC26)
Alternate project:SOFTA420

```

Figure 6-19 ISPF-PT SCLM walk through

- Copy the member TSTSVC26 to the member TSTSVC30, as shown in Figure 6-20.

File Display Library SCLM Settings Menu Utilities Test Help Exit									
-----									
-IPT--EDIT L1----- SOFTAUD.DEVI025.ASM					-----ROW 00001 OF 00343-				
COMMAND ==>					SCROLL ==> CSR				
HOTBAR: REFRESH FLIP SORT CHA TAILOR COMPRESS EXPDIR INFO SET									
*SORT*					LIB2=INT420		LIB3=TEST420		LIB4=RLSE420
NAME	RENAME	LIB	VV.MM	CREATED	CHANGED	SIZE	INIT	MOD	USERID
TESTLONG		S 1	01.02	04/08/19	07/09/06 09:46	176	150	0	INTT125
TESTSCSI		S 1	01.01	03/06/24	03/06/25 12:57	430	429	0	INTT025
c TSTSVC26	TSTSVC30	S 2	01.01	07/09/06	07/09/06 10:28	84	84	0	INTT125
AUCMBASE		S 3	01.01	07/03/15	07/03/26 08:56	861	861	0	INTT070
AUCMSENU		S 3	01.01	07/03/15	07/03/26 08:56	2082	2082	0	INTT070
AUDCMPID		S 3	02.45	01/05/29	07/08/30 16:54	4762	3918	0	INTT068
AUDCMPKB		S 3	01.94	01/10/02	07/08/30 16:55	4556	3358	0	INTT068
AUDCMPSK		S 3	01.42	01/08/28	07/08/30 16:56	3662	3620	0	INTT068
AUDCROSS		S 3	01.99	93/05/04	07/05/10 14:35	868	847	0	INTT011
AUDDIMON		S 3	01.99	01/12/09	07/06/22 12:00	1719	1605	0	INTT068
AUDDIPTB		S 3	01.99	01/12/10	07/08/21 09:22	2767	1847	0	INTT068
AUDDIST		S 3	02.99	97/10/28	07/08/30 17:00	2456	593	0	INTT068
AUDDIUNI		S 3	01.62	06/05/03	07/03/06 10:23	262	126	0	INTT068
AUDDIXML		S 3	01.99	06/05/30	07/08/21 11:23	2089	1306	0	INTT068
AUDEXPRM		S 3	04.05	04/01/05	06/11/17 11:51	2934	1282	0	INTT070
AUDFBOIL		S 3	03.64	99/10/06	07/04/25 15:00	3918	2363	0	INTT065
AUDFILTM		S 3	02.99	97/12/18	07/08/30 17:04	5936	3858	0	INTT068

Figure 6-20 ISPF-PT SCLM walk through

- Select TSTSVC30, as shown in Figure 6-21.

File Display Library SCLM Settings Menu Utilities Test Help Exit									
-----									
-IPT--EDIT L1----- SOFTAUD.DEVI025.ASM					-----ROW 00003 OF 00344-				
COMMAND ==>					SCROLL ==> CSR				
HOTBAR: REFRESH FLIP SORT CHA TAILOR COMPRESS EXPDIR INFO SET									
*SORT* LIB2=INT420 LIB3=TEST420 LIB4=RLSE420									
NAME	RENAME	LIB	VV.MM	CREATED	CHANGED	SIZE	INIT	MOD	USERID
S TSTSVC30		S 1	01.01	07/09/06	07/09/06 10:28	84	84	0	INTT125
TSTSVC26	-CREATED	S 2	01.01	07/09/06	07/09/06 10:28	84	84	0	INTT125
AUCMBASE		S 3	01.01	07/03/15	07/03/26 08:56	861	861	0	INTT070
AUCMSENU		S 3	01.01	07/03/15	07/03/26 08:56	2082	2082	0	INTT070
AUDCMPID		S 3	02.45	01/05/29	07/08/30 16:54	4762	3918	0	INTT068
AUDCMPKB		S 3	01.94	01/10/02	07/08/30 16:55	4556	3358	0	INTT068
AUDCMPSK		S 3	01.42	01/08/28	07/08/30 16:56	3662	3620	0	INTT068
AUDCROSS		S 3	01.99	93/05/04	07/05/10 14:35	868	847	0	INTT011
AUDDIMON		S 3	01.99	01/12/09	07/06/22 12:00	1719	1605	0	INTT068
AUDDIPTB		S 3	01.99	01/12/10	07/08/21 09:22	2767	1847	0	INTT068
AUDDIST		S 3	02.99	97/10/28	07/08/30 17:00	2456	593	0	INTT068
AUDDIUNI		S 3	01.62	06/05/03	07/03/06 10:23	262	126	0	INTT068
AUDDIXML		S 3	01.99	06/05/30	07/08/21 11:23	2089	1306	0	INTT068
AUDEXPRM		S 3	04.05	04/01/05	06/11/17 11:51	2934	1282	0	INTT070
AUDFBOIL		S 3	03.64	99/10/06	07/04/25 15:00	3918	2363	0	INTT065
AUDFILTM		S 3	02.99	97/12/18	07/08/30 17:04	5936	3858	0	INTT068
AUDFLTVP		S 3	01.31	99/12/24	07/08/30 17:50	841	390	0	INTT068

Figure 6-21 ISPF-PT SCLM walk through

- As shown in Figure 6-22, change all occurrences of TSTSVC26 to TSTSVC30, and save the member.

```

File Edit Settings Build SCLM Menu Utilities Compilers Test Help
-IPT- EDIT SOF AUD DEV I025.ASM(TSTSVC26) - 01.01 Columns 00001 00072
Command ==> c all TSTSVC26 TSTSVC30 Scroll ==> CSR
***** Top of Data *****
000001 TSTSVC26 TITLE 'Test LOCATE SVC 26'
000002 *****
000003 * Execution JCL
000004 *
000005 * =====
000006 * // EXEC PGM=IPIUTIL,PARM='<dsname>'
000007 *****
000008 *
000009 * *****
000010 * * PROLOG *
000011 * *****
000012 TSTSVC26 RMODE ANY
000013 TSTSVC26 AMODE 31
000014 TSTSVC26 CSECT ,
000015 SAVE (14,12),,*
000016 LR R12,R15
000017 USING TSTSVC26,R12
F1=HELP F2=SPLIT F3=END F4=VIEW F5=RFIND F6=RCHANGE
F7=UP F8=DOWN F9=SWAP F10=LEFT F11=RIGHT F12=RETRIEVE

```

Figure 6-22 ISPF-PT SCLM walk through

- We can invoke the SCLM build function directly from ISPF-PT. Use the K command with the parameter “BUILD” in the RENAME field, as shown in Figure 6-23, to assemble and link the new program.

```

File Display Library SCLM Settings Menu Utilities Test Help Exit
-IPT--EDIT L1----- SOF AUD DEV I025.ASM ----- TSTSVC30 saved & parsed
COMMAND ==> SCROLL ==> CSR
HOTBAR: REFRESH FLIP SORT CHA TAILOR COMPRESS EXPDIR INFO SET
*SORT* LIB2=INT420 LIB3=TEST420 LIB4=RLSE420
NAME RENAME LIB VV.MM CREATED CHANGED SIZE INIT MOD USERID
k TSTSVC30 build 1 01.09 07/09/06 07/09/12 08:32 84 84 0 INTT125
AUCMBASE S 3 01.01 07/03/15 07/03/26 08:56 861 861 0 INTT070
AUCMSENU S 3 01.01 07/03/15 07/03/26 08:56 2082 2082 0 INTT070
AUDCMPID S 3 02.45 01/05/29 07/08/30 16:54 4762 3918 0 INTT068
AUDCMPKB S 3 01.94 01/10/02 07/08/30 16:55 4556 3358 0 INTT068
AUDCMPSK S 3 01.42 01/08/28 07/08/30 16:56 3662 3620 0 INTT068
AUDCROSS S 3 01.99 93/05/04 07/05/10 14:35 868 847 0 INTT011
AUDDIMON S 3 01.99 01/12/09 07/06/22 12:00 1719 1605 0 INTT068
AUDDIPTB S 3 01.99 01/12/10 07/08/21 09:22 2767 1847 0 INTT068
AUDDIST S 3 02.99 97/10/28 07/08/30 17:00 2456 593 0 INTT068
AUDDIUNI S 3 01.62 06/05/03 07/03/06 10:23 262 126 0 INTT068
AUDDIXML S 3 01.99 06/05/30 07/08/21 11:23 2089 1306 0 INTT068
AUDEXPRM S 3 04.05 04/01/05 06/11/17 11:51 2934 1282 0 INTT070
AUDFBOIL S 3 03.64 99/10/06 07/04/25 15:00 3918 2363 0 INTT065
AUDFILTM S 3 02.99 97/12/18 07/08/30 17:04 5936 3858 0 INTT068
F1=HELP F2=SPLIT F3=END F4=VIEW F5=RFIND F6=RCHANGE
F7=UP F8=DOWN F9=SWAP F10=LEFT F11=RIGHT F12=RETRIEVE

```

Figure 6-23 ISPF-PT SCLM walk through

- The action from Figure 6-23 invokes the SCLM Build – Entry panel. To assemble the program in online, use the EX command, as shown in Figure 6-24 on page 154.

Menu SCLM Utilities Jobcard Test Workstation Build Help					
SCLM Build - Entry Panel					
Command ==> <u>ex</u>					
Build input:					
Project . . .	SOFTAUD	Alternate -	SOFTA420		
Group . . .	DEVI025				
Type . . .	ASM	Enter "/" to select option			
Member . . .	TSTSVC30	/ Error Listings only			
		- Workstation Build			
Mode . . .	1	1. Conditional	Scope . . .	4	1. Limited
	2	2. Unconditional		2	2. Normal
	3	3. Forced		3	3. Subunit
	4	4. Report		4	4. Extended
Output control:					
Ex Sub	1 2	1. Terminal	Process . . .	2	1. Execute
Messages . . .	1 2	2. Printer		2	2. Submit
Report . . .	1 2	3. Data set	Printer . . .	H	
Listings . . .	1 2	4. None	Volume . . .		
F1=HELP	F2=SPLIT	F3=END	F4=VIEW	F5=RFIND	F6=RCHANGE
F7=UP	F8=DOWN	F9=SWAP	F10=LEFT	F11=RIGHT	F12=RETRIEVE

Figure 6-24 ISPF-PT SCLM walk through

**Hint:** If you want to build the project ARCHDEF, change the TYPE to ARCHDEF, and enter the correct member name on the window in Figure 6-24.

## 6.3.2 Migrating members to SCLM

In this section, we demonstrate how to migrate members using ISPF-PT.

Figure 6-25 shows how to Edit the JCL library. This library has four members that we want to migrate to SCLM.

File Edit Find Display Populate Settings Menu Util Test Help Exit					
-IPT- OLIST (E) ----- Personal list ----- Row 1 to 14 of 14					
Command ==> <u>14</u> SCROLL ==> CSR					
Hotbar: REFRESH FLIP UTIL FILLVOL CLRVOL SET CUT UPDATE					
Open list ==> MYLIST (or BLANK for reference list)					
TSO PARMS ==>					
Command	Member	Numbr	Data Set Names / Objects	Class	
-----					
	1		!----- PERSONAL LIBRARIES -----		
	2		'INTT125.CLIST'		
	3		'INTT125*'	LIST	
	4		JCL		
	5		!----- LCM2 SUPPORT -----		
-E	6		<SOFTAUD.DEVI025.ASM	SCLM	
	7		=SOFTAUD DEVI025 TEST420 RLSE420 ASM	ISPF	
	8		'SOFTAUD.DEVI025*'	LIST	
	9		'SOFTAUD.INT420*'	LIST	
	10		'SOFTAUD.TEST420*'	LIST	
	11		'SOFTAUD.RLSE420*'	LIST	
	12		:LISTW SOFTAUD	OLIST	
	13		'SOFTAUD.PROJ*'	LIST	
	14		JCL		
----- END OF LIST -----					

Figure 6-25 ISPF-PT SCLM walk through



Figure 6-26 demonstrates how to use the COPY command to copy all four members.

```

File Display Library SCLM Settings Menu Utilities Test Help Exit
-----
-IPT--EDIT L1----- INTT125.JCL -----ROW 00001 OF 00004
COMMAND ==> copy TST* SCROLL ==> CSR
HOTBAR: REFRESH FLIP SORT CHA TAILOR COMPRESS EXPDIR INFO SET
ON VOLUME USR033
NAME RENAME LIB VV.MM CREATED CHANGED SIZE INIT MOD USERID
TSTJIM30 1 01.00 07/09/07 07/09/07 12:45 10 10 0 INTT125
TSTJIM31 1 01.00 07/09/07 07/09/07 14:13 10 10 0 INTT125
TSTJIM32 1 01.00 07/09/07 07/09/07 14:13 10 10 0 INTT125
TSTJIM33 1 01.00 07/09/07 07/09/07 14:13 10 10 0 INTT125
--END--

```

Figure 6-26 ISPF-PT SCLM walk through

As shown in Figure 6-27, the target library is our SCLM development JCL library.

```

-IPT----- COPY PANEL -----(FROM INTT125.JCL) -----
COMMAND ==>
Specify "TO" data set below (4 members will be processed):

PROJECT ==> softaud
GROUP ==> devi025
TYPE ==> JCL

Or other Partitioned, Sequential Data Set or @H (History List):
DATA SET NAME ==> -
VOLUME SERIAL ==> - (If not catalogued)
PASSWORD ==> - (If password protected)

COPY OPTIONS:
REPLACE like-named library members ==> 0 (Y=Yes, N=No, 0=Target is Older)
Disposition for sequential target ==> OLD (OLD or MOD)
Note: If statistics are not available, replace option 0 (older) is
treated as option Y (Yes).

Press ENTER to copy END key to cancel.

```

Figure 6-27 ISPF-PT SCLM walk through

Figure 6-28 on page 156 displays the confirmation panel.

```

-IPT--EDIT----- COPY PANEL -----
COMMAND ==> _

You have requested that 4 members be COPIED to SOF AUD.DEVI025.JCL.

Indicate a new member name in the RENAME column, or leave it blank to
use the same member name.

      Press ENTER to process the list, or the END key to cancel.
      (Use the PF keys for scrolling to see the entire list.)

      NAME      RENAME      LIB VV.MM  CREATED      CHANGED      SIZE  INIT  MOD  USERID
TSTJIM30              1 01 00 07/09/07 07/09/07 12:45    10    10    0  INTT125
TSTJIM31              1 01 00 07/09/07 07/09/07 14:13    10    10    0  INTT125
TSTJIM32              1 01 00 07/09/07 07/09/07 14:13    10    10    0  INTT125
TSTJIM33              1 01 00 07/09/07 07/09/07 14:13    10    10    0  INTT125

```

Figure 6-28 ISPF-PT SCLM walk through

As shown in Figure 6-29, the copy process was successful for all members.

```

File Display Library SCLM Settings Menu Utilities Test Help Exit
-----
-IPT--EDIT L1----- INTT125.JCL -----ROW 00001 OF 00004
COMMAND ==> _                                SCROLL ==> CSR
HOTBAR: REFRESH FLIP      SORT CHA TAILOR  COMPRESS EXPDIR  INFO  SET
                                         ON VOLUME USR033

      NAME      RENAME      LIB VV.MM  CREATED      CHANGED      SIZE  INIT  MOD  USERID
TSTJIM30 -COPIED      1 01.00 07/09/07 07/09/07 12:45    10    10    0  INTT125
TSTJIM31 -COPIED      1 01.00 07/09/07 07/09/07 14:13    10    10    0  INTT125
TSTJIM32 -COPIED      1 01.00 07/09/07 07/09/07 14:13    10    10    0  INTT125
TSTJIM33 -COPIED      1 01.00 07/09/07 07/09/07 14:13    10    10    0  INTT125
      --END--

```

Figure 6-29 ISPF-PT SCLM walk through

Edit the SCLM ASM library, switch to the TYPE JCL, and SORT the members in LIB order, as shown in Figure 6-30 on page 157.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (E) ----- Personal list ----- Row 1 to 14 of 14
Command === 6:type jcl;sort lib SCROLL ==> CSR
Hotbar: REFRESH FLIP UTIL FILLVOL CLRVOL SET CUT UPDATE
Open list ==> MYLIST (or BLANK for reference list)
TSO PARMS ==>
Command Member Numbr Data Set Names / Objects Class
-----
1 |----- PERSONAL LIBRARIES -----
2 'INTT125.CLIST'
3 'INTT125*' LIST
4 JCL
5 |----- LCM2 SUPPORT -----
6 <SOFTAUD.DEVI025.ASM SCLM
7 =SOFTAUD DEVI025 TEST420 RLSE420 ASM ISPF
8 'SOFTAUD.DEVI025*' LIST
9 'SOFTAUD.INT420*' LIST
10 'SOFTAUD.TEST420*' LIST
11 'SOFTAUD.RLSE420*' LIST
12 :LISTW SOFTAUD OLIST
13 'SOFTAUD.PROJ*' LIST
14 JCL PDSE
-----
-E
-E
END OF LIST -----

```

Figure 6-30 ISPF-PT SCLM walk through

Note that the LIB information for the TSTJIM\* members shown in Figure 6-31. Because there is no “S” under the LIB column, these members do not have SCLM accounting information available. Use the SC command to migrate all four members. The SCLMPARM window default values are applied during the MIGRATE process.

```

File Display Library SCLM Settings Menu Utilities Test Help Exit
-----
-IPT--EDIT L1----- SOFTAUD DEVI025.JCL -----ROW 00001 OF 00049-
COMMAND ==> sc tstjim* migrate SCROLL ==> CSR
HOTBAR: REFRESH FLIP SORT CHA TAILOR COMPRESS EXPDIR INFO SET
*SORT* LIB2=INT420 LIB3=TEST420 LIB4=RLSE420
NAME RENAME LIB VV.MM CREATED CHANGED SIZE INIT MOD USERID
TSTJIM30 1 01.00 07/09/07 07/09/07 12:45 10 10 0 INTT125
TSTJIM31 1 01.00 07/09/07 07/09/07 14:13 10 10 0 INTT125
TSTJIM32 1 01.00 07/09/07 07/09/07 14:13 10 10 0 INTT125
TSTJIM33 1 01.00 07/09/07 07/09/07 14:13 10 10 0 INTT125
TSTSVC26 S 2 01.04 07/09/06 07/09/07 14:18 7 5 0 INTT125
ALOCDATA S 4 03.41 93/10/01 06/09/26 10:47 61 32 0 INTT065
ANALYZER S 4 01.10 02/04/08 06/09/26 10:48 144 130 0 INTT065
AUDISTAT S 4 01.01 03/12/04 04/08/06 07:50 20 7 0 INTT070
AUDITLCM S 4 03.37 93/10/01 06/05/02 17:48 38 22 0 INTT011
AUDKBUTL S 4 01.17 99/03/08 06/08/15 20:37 50 70 0 INTT065
AUDUTIL S 4 01.13 01/11/15 06/05/02 17:50 42 8 0 INTT011
COLLECTR S 4 01.03 03/12/12 06/08/18 12:30 45 1 0 INTT009
COMPARE S 4 01.09 01/05/29 04/08/06 07:50 81 55 0 INTT070
COMPARKB S 4 01.06 01/10/03 06/08/16 14:43 66 56 0 INTT065
COMPRSKB S 4 01.08 01/08/29 06/08/16 14:43 76 80 0 INTT065
COPYAUTH S 4 03.10 93/10/01 06/10/02 16:28 3 2 0 INTT015
COPYSURV S 4 01.03 93/11/23 99/05/12 12:35 2 2 0 INTT007

```

Figure 6-31 ISPF-PT SCLM walkthrough

As shown in Figure 6-32 on page 158, the SCLM Migrate function completed successfully.

File Display Library SCLM Settings Menu Utilities Test Help Exit									
-----									
-IPT--EDIT L1----- SOFPAUD.DEVI025.TCL -----ROW 00001 OF 00049-									
COMMAND ==> type archdef;filter lib 2 SCROLL ==> CSR									
HOTBAR: REFRESH FLIP SORT CHA TAILOR COMPRESS EXPDIR INFO SET									
*SORT* LIB2=INT420 LIB3=TEST420 LIB4=RLSE420									
NAME	RENAME	LIB	VV.MM	CREATED	CHANGED	SIZE	INIT	MOD	USERID
TSTJIM30	-MIGRATE	S	1	01.00	07/09/07	12:45	10	10	0 INTT125
TSTJIM31	-MIGRATE	S	1	01.00	07/09/07	14:13	10	10	0 INTT125
TSTJIM32	-MIGRATE	S	1	01.00	07/09/07	14:13	10	10	0 INTT125
TSTJIM33	-MIGRATE	S	1	01.00	07/09/07	14:13	10	10	0 INTT125
TSTSVC26		S	2	01.04	07/09/06	14:18	7	5	0 INTT125
ALOCDATA		S	4	03.41	93/10/01	06/09/26	10:47	61	32 0 INTT065
ANALYZER		S	4	01.10	02/04/08	06/09/26	10:48	144	130 0 INTT065
AUDISTAT		S	4	01.01	03/12/04	04/08/06	07:50	20	7 0 INTT070
AUDITLCM		S	4	03.37	93/10/01	06/05/02	17:48	38	22 0 INTT011
AUDKBUTL		S	4	01.17	99/03/08	06/08/15	20:37	50	70 0 INTT065
AUDUTIL		S	4	01.13	01/11/15	06/05/02	17:50	42	8 0 INTT011
COLLECTR		S	4	01.03	03/12/12	06/08/18	12:30	45	1 0 INTT009
COMPARE		S	4	01.09	01/05/29	04/08/06	07:50	81	55 0 INTT070
COMPARKB								56	0 INTT065
COMPRSKB								80	0 INTT065
COPYAUTH								2	0 INTT015
COPYSURV		S	4	01.03	93/11/23	99/05/12	12:35	2	2 0 INTT007

Figure 6-32 ISPF-PT SCLM walk through

**Hint:** If you copy a member into a library and you want to reset statistics, such as create time and date, use the “Z” MSL command to invoke the MSL Statistics Settings Panel.

### 6.3.3 Updating the project SCLM ARCHDEF members

In this section, we show you how to update the SCLM ARCHDEF members.

1. Select the ARCHDEF members shown in Figure 6-33.

File Display Library SCLM Settings Menu Utilities Test Help Exit									
-----									
-IPT--EDIT L1----- SOFPAUD.DEVI025.ARCHDEF -----ROW 00001 OF 00004-									
COMMAND ==> SCROLL ==> CSR									
HOTBAR: REFRESH FLIP SORT CHA TAILOR COMPRESS EXPDIR INFO SET									
*FILTER* LIB2=INT420 LIB3=TEST420 LIB4=RLSE420									
NAME	RENAME	LIB	VV.MM	CREATED	CHANGED	SIZE	INIT	MOD	USERID
\$\$\$STALL		S	2	01.00	07/09/06	10:54	5	5	0 INTT125
\$ \$TSTASM		S	2	01.00	07/09/06	10:56	4	4	0 INTT125
\$ \$TSTJCL		S	2	01.00	07/09/06	10:58	4	4	0 INTT125
TSTSVC26		S	2	01.02	07/09/06	10:32	10	10	0 INTT125
--END--									
-----									
F1=HELP	F2=SPLIT	F3=END	F4=VIEW	F5=RFIND	F6=RCHANGE				
F7=UP	F8=DOWN	F9=SWAP	F10=LEFT	F11=RIGHT	F12=RETRIEVE				

Figure 6-33 ISPF-PT SCLM walk through

2. Change the TSTSVC26 ARCHDEF to TSTSVC30, as shown in Figure 6-34.

```

File Edit Settings Build SCLM Menu Utilities Compilers Test Help
-IPT- EDIT SOFATAUD.DEVI025.ARCHDEF($TSTASM) - 01.00 Columns 00001 00072
Command ==> Scroll ==> CSR
***** Top of Data *****
==MSG> IQIX607 SPIFFY issued an SCLM lock for this member.
==MSG> SPIFFY will parse the member when you terminate the edit session
==MSG> -Warning- The UNDO command is not available until you change
==MSG> your edit profile using the command RECOVERY ON.
==MSG> IQIS002 SCLM is using alternate project "SOFATA420" with this file.
000001 *
000002 * HL Archetecture member - All ASM based modules of TST component
000003 *
000004 INCL TSTSVC30 ARCHDEF
***** Bottom of Data *****

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

```

Figure 6-34 ISPF-PT SCLM walk through

3. Update the \$TSTJCL members with the four JCL library members, as shown in Figure 6-35.

```

File Edit Settings Build SCLM Menu Utilities Compilers Test Help
-IPT- EDIT SOFATAUD.DEVI025.ARCHDEF($TSTJCL) - 01.01 Columns 00001 00072
Command ==> Scroll ==> CSR
***** Top of Data *****
==MSG> IQIX607 SPIFFY issued an SCLM lock for this member.
==MSG> SPIFFY will parse the member when you terminate the edit session
==MSG> -Warning- The UNDO command is not available until you change
==MSG> your edit profile using the command RECOVERY ON.
==MSG> IQIS002 SCLM is using alternate project "SOFATA420" with this file.
000001 *
000002 * HL Archetecture member - All JCL members of TST component
000003 *
000004 INCLD TSTSVC30 JCL
000005 INCLD TSTSVC31 JCL
000006 INCLD TSTSVC32 JCL
000007 INCLD TSTSVC33 JCL
***** Bottom of Data *****

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

```

Figure 6-35 ISPF-PT SCLM walk through

4. Create a new TSTSVC30 ARCHDEF member, as shown in Figure 6-36.

```

File Display Library SCLM Settings Menu Utilities Test Help Exit
-----
-IPT--EDIT L1----- SOFTAUD.DEVI025.ARCHDEF ----- $TSTJCL saved & parsed
COMMAND ==>
HOTBAR: REFRESH FLIP SORT CHA TAILOR COMPRESS EXPDIR INFO SET
*FILTER* 109 HIDDEN LIB2=INT420 LIB3=TEST420 LIB4=RLSE420
NAME RENAME LIB VV.MM CREATED CHANGED SIZE INIT MOD USERID
$TSTJCL SAVED S 1 01.01 07/09/06 07/09/10 15:45 7 4 0 INTT125
C TSTSVC26 TSTSVC30 S 2 01.02 07/09/06 07/09/06 10:32 10 10 0 INTT125
--END--

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

```

Figure 6-36 ISPF-PT SCLM walk through

5. Select the new ARCHDEF member, as shown in Figure 6-37.

```

File Display Library SCLM Settings Menu Utilities Test Help Exit
-----
-IPT--EDIT L1----- SOFTAUD.DEVI025.ARCHDEF -----ROW 00004 OF 00006-
COMMAND ==>
HOTBAR: REFRESH FLIP SORT CHA TAILOR COMPRESS EXPDIR INFO SET
*FILTER* *SORT*108 HIDDEN LIB2=INT420 LIB3=TEST420 LIB4=RLSE420
NAME RENAME LIB VV.MM CREATED CHANGED SIZE INIT MOD USERID
TSTSVC26 -COPIED S 2 01.02 07/09/06 07/09/06 10:32 10 10 0 INTT125
S TSTSVC30 S 1 01.02 07/09/06 07/09/06 10:32 10 10 0 INTT125
TESTLONG S 1 01.01 04/08/19 07/08/17 10:58 11 11 0 INTT025
--END--

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

```

Figure 6-37 ISPF-PT SCLM walk through

6. Change the TSTSVC26 to TSTSVC30, as shown Figure 6-38 on page 161.

```

File Edit Settings Build SCLM Menu Utilities Compilers Test Help
-IPT- EDIT SOFATAUD.DEVI025.ARCHDEF(TSTSVC30) - 01.04 Columns 00001 00072
Command ==> c all TSTSVC26 TSTSVC30 Scroll ==> CSR
***** Top of Data *****
==MSG> IQIX604 This is an SCLM locked member.
==MSG> SPIFFY will parse the member when you terminate the edit session
==MSG> -Warning- The UNDO command is not available until you change
==MSG> your edit profile using the command RECOVERY ON.
==MSG> IQIS002 SCLM is using alternate project "SOFATA420" with this file.
000001 *
000002 * First-level architecture definition for TSTSVC26 executable
000003 *
000004 LKED LE370
000005 LOAD TSTSVC26 LLIB
000006 LMAP TSTSVC26 LMAP
000007 PARM AMODE=31,RMODE=ANY,REUS,XREF,NCAL,AC=0
000008 *
000009 INCLD TSTSVC26 ASM
000010 CMD ENTRY TSTSVC26
***** Bottom of Data *****

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

```

Figure 6-38 ISPF-PT SCLM walk through

### 6.3.4 Building and promoting the SCLM ARCHDEF members

In this section, we show you how to build and promote the SCLM ARCHDEF members using ISPF-PT.

1. The ARCHDEF updates are complete. Enter the "K" ISPF-PT line command to invoke the SCLM window, shown in Figure 6-39.

```

File Display Library SCLM Settings Menu Utilities Test Help Exit
-----
-IPT--EDIT L1----- SOFATAUD.DEVI025.ARCHDEF -----ROW 00001 OF 00006-
COMMAND ==> SCROLL ==> CSR
HOTBAR: REFRESH FLIP SORT CHA TAILOR COMPRESS EXPDIR INFO SET
*FILTER* *SORT*108 HIDDEN LIB2=INT420 LIB3=TEST420 LIB4=RLSE420
NAME RENAME LIB VV.MM CREATED CHANGED SIZE INIT MOD USERID
k TSTSVC30 build S 1 01.09 07/09/06 07/09/12 08:56 10 10 0 INTT125
k $TSTASM build S 1 01.10 07/09/06 07/09/12 08:24 4 4 0 INTT125
k $TSTJCL build_ S 1 01.07 07/09/06 07/09/12 08:24 7 4 0 INTT125
$TSTALL S 2 01.00 07/09/06 07/09/06 10:54 5 5 0 INTT125
TSTSVC26 S 2 01.02 07/09/06 07/09/06 10:32 10 10 0 INTT125
TESTLONG S 1 01.01 04/08/19 07/08/17 10:58 11 11 0 INTT025
--END--

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

```

Figure 6-39 ISPF-PT SCLM walk through

As shown in Figure 6-40 on page 162, the SCLM Build Enter Panel builds the ARCHDEF member TSTSVC30.

Menu SCLM Utilities Jobcard Test Workstation Build Help					
SCLM Build - Entry Panel					
Command ==> <u>ex</u>					
Build input:					
Project . . .		SOFTAUD Alternate - SOFTA420			
Group . . .		DEVI025			
Type . . .		ARCHDEF			
Member . . .		TSTSVC30			
		Enter "/" to select option			
		/ Error Listings only			
		- Workstation Build			
Mode . . . <u>1</u>		1. Conditional		Scope . . . <u>4</u>	
		2. Unconditional		1. Limited	
		3. Forced		2. Normal	
		4. Report		3. Subunit	
				4. Extended	
Output control:					
		Ex Sub		Process . . . <u>2</u>	
Messages . . . <u>1</u> <u>2</u>		1. Terminal		1. Execute	
Report . . . <u>1</u> <u>2</u>		2. Printer		2. Submit	
Listings . . . <u>1</u> <u>2</u>		3. Data set		Printer . . . <u>H</u>	
		4. None		Volume . . .	
F1=HELP		F2=SPLIT		F3=END	
F7=UP		F8=DOWN		F4=VIEW	
				F5=RFIND	
				F6=RCHANGE	
				F10=LEFT	
				F11=RIGHT	
				F12=RETRIEVE	

Figure 6-40 ISPF-PT SCLM walk through

As shown in Figure 6-41, the SCLM Build Enter Panel builds the ARCHDEF member \$TSTASM.

Menu SCLM Utilities Jobcard Test Workstation Build Help					
SCLM Build - Entry Panel					
Command ==> <u>ex</u>					
Build input:					
Project . . .		SOFTAUD Alternate - SOFTA420			
Group . . .		DEVI025			
Type . . .		ARCHDEF			
Member . . .		\$TSTASM			
		Enter "/" to select option			
		/ Error Listings only			
		- Workstation Build			
Mode . . . <u>1</u>		1. Conditional		Scope . . . <u>4</u>	
		2. Unconditional		1. Limited	
		3. Forced		2. Normal	
		4. Report		3. Subunit	
				4. Extended	
Output control:					
		Ex Sub		Process . . . <u>2</u>	
Messages . . . <u>1</u> <u>2</u>		1. Terminal		1. Execute	
Report . . . <u>1</u> <u>2</u>		2. Printer		2. Submit	
Listings . . . <u>1</u> <u>2</u>		3. Data set		Printer . . . <u>H</u>	
		4. None		Volume . . .	
F1=HELP		F2=SPLIT		F3=END	
F7=UP		F8=DOWN		F4=VIEW	
				F5=RFIND	
				F6=RCHANGE	
				F10=LEFT	
				F11=RIGHT	
				F12=RETRIEVE	

Figure 6-41 ISPF-PT SCLM walk through

As shown in Figure 6-42 on page 163, the SCLM Build Enter Panel builds the ARCHDEF member \$TSTJCL.



```

Menu  SCLM  Utilities  Jobcard  Test  Workstation Build  Help
-----
SCLM Build - Entry Panel

Command ==> ex

Build input:
Project . . : SOFTAUD  Alternate - SOFTA420
Group . . . DEVI025
Type . . . . ARCHDEF
Member . . . $TSTJCL

Enter "/" to select option
/ Error Listings only
- Workstation Build

Mode . . . 1 1. Conditional
                2. Unconditional
                3. Forced
                4. Report
Scope . . . 4 1. Limited
                2. Normal
                3. Subunit
                4. Extended

Output control:
Ex Sub
Messages . . 1 2 1. Terminal
Report . . . 1 2 2. Printer
Listings . . 1 2 3. Data set
                4. None
Process . . . 2 1. Execute
                2. Submit
Printer . . . H
Volume . . .

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

```

Figure 6-42 ISPF-PT SCLM walk through

The ARCHDEFS builds completed successfully. As shown above in Figure 6-43, next to the ARCHDEF \$TSTASM, we entered the “K” command with the parameter “PROMOTE” in the RENAME.

- For the second ARCHDEF, \$TSTJCL, enter the “K” command without any parameters.

```

File  Display  Library  SCLM  Settings  Menu  Utilities  Test  Help  Exit
-----
-IPT--EDIT L1----- SOFTAUD.DEVI025.ARCHDEF -----ROW 00001 OF 00006-
COMMAND ==>
HOTBAR: REFRESH  FLIP      SORT CHA TAILOR  COMPRESS EXPDIR  INFO  SET
*FILTER*
NAME  RENAME  LIB  VV.MM  CREATED  CHANGED  SIZE  INIT  MOD  USERID
TSTSVC30 -BUILD $ 1 01.11 07/09/06 07/09/12 09:12 10 10 0 INTT125
k $TSTASM promote $ 1 01.11 07/09/06 07/09/12 09:11 4 4 0 INTT125
k $TSTJCL -BUILD $ 1 01.08 07/09/06 07/09/12 09:11 7 4 0 INTT125
$TSTALL $ 2 01.00 07/09/06 07/09/06 10:54 5 5 0 INTT125
TSTSVC26 $ 2 01.02 07/09/06 07/09/06 10:32 10 10 0 INTT125
TESTLONG $ 1 01.01 04/08/19 07/08/17 10:58 11 11 0 INTT025
--END--

IQIM259 SCLM BUILD issued on 1 members

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

```

Figure 6-43 ISPF-PT SCLM walk through

The parameter PROMOTE invokes the SCLM Promote Panel that is depicted in Figure 6-44 on page 164. Using the EX primary command, the project \$TSTASM is promoted successfully to the INT420 library.

```

Menu  SCLM  Utilities  Jobcard  Workstation  Promote  Help

SCLM Promote - Entry Panel

Command ==> ex

Promote input:
Project . . . : SOFTAUD           Alternate - SOFTA420
From group . . DEVI025
Type . . . . . ARCHDEF           Enter "/" to select option
Member . . . . $TSTASM           _ Workstation Promote

Mode . . . 1 1. Conditional       Scope . . . 3 1. Normal
                2. Unconditional   2. Subunit
                3. Report           3. Extended

Output control:
Ex Sub
Messages . . 1 2 1. Terminal       Process . . 2 1. Execute
Report . . . 1 2 2. Printer         2. Submit
                3. Data set       Printer . . H
                4. None           Volume . . .
F1=HELP    F2=SPLIT  F3=END    F4=VIEW    F5=RFIND   F6=RCHANGE
F7=UP      F8=DOWN   F9=SWAP   F10=LEFT  F11=RIGHT  F12=RETRIEVE

```

Figure 6-44 ISPF-PT SCLM walk through

3. The Figure 6-45 shows the Promote messages. To see the Promote report (not shown), enter a Y in the value that we circled in Figure 6-44.

```

-IPT- - SCLM PROCESS MESSAGES ----- LINE 00000000 COL 001 080
COMMAND ==>
Display SCLM report (if available) ==> N (Y=Yes, N=No)  SCROLL ==> PAGE
Library :SOFTAUD.DEVI025.ARCHDEF($TSTASM)
Language:ARCHDEF  Change code:V4R2M0  Auth. code:      Alternate:SOFTA420
----- Press the END key to exit -----
***** Top of Data *****
FLM51000 - PROMOTE PROCESSOR INITIATED - 17:25:11 ON 2007/09/10
FLM52000 - INITIATING VERIFICATION PHASE - 17:25:11 ON 2007/09/10
FLM55000 - INITIATING COPY PHASE - 17:25:11 ON 2007/09/10
FLM57000 - INITIATING PURGE PHASE - 17:25:13 ON 2007/09/10
FLM57001 - INITIATING PURGE FROM GROUP: DEVI025
FLM58000 - PROMOTE PROCESSOR COMPLETED - 17:25:13 ON 2007/09/10
***** Bottom of Data *****

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

```

Figure 6-45 ISPF-PT SCLM walk through

Figure 6-46 on page 165 depicts the request to promote the second ARCHDEF, \$TSTJCL.

```

-IPT- - SCLM PROCESS MESSAGES ----- LINE 00000000 COL 001 080
                                     SCLM COMMAND PROMPT
      SCLM_Menu Menu Utilities Test Exit
-----
Command      ==> _____

Select the function to execute on the selected member(s):
SCLM function ==> 8 (1=Build, 2=Delete, 3=Info, 4=Lock,
                    5=Parse, 6=Promote, 7=Migrate, 8=Unlock)
For Build, Delete, and Promote:
  Execution type ==> 1 (1=Immediate (online), 2=Use SCLM dialog)

Current library:  SOFTAUD.DEVI025.ARCHDEF
Current member :  $TSTJCL

Specify/verify SCLM parameter for this library:
Change code      ==> V4R2M0
Language         ==> ARCHDEF (Default language)
Auth. code       ==> _____

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

```

Figure 6-46 ISPF-PT SCLM walk through

Figure 6-47 shows the successful promotion of the \$TSTJCL ARCHDEF.

```

-IPT- - SCLM PROCESS MESSAGES ----- LINE 00000000 COL 001 080
COMMAND ==> _____
Display SCLM report (if available) ==> N (Y=Yes, N=No)
Library :SOFTAUD.DEVI025.ARCHDEF($TSTJCL)
Language:ARCHDEF Change code:V4R2M0 Auth. code:      Alternate:SOFTA420
----- Press the END key to exit -----
***** Top of Data *****
FLM51000 - PROMOTE PROCESSOR INITIATED - 17:26:11 ON 2007/09/10
FLM52000 - INITIATING VERIFICATION PHASE - 17:26:11 ON 2007/09/10
FLM55000 - INITIATING COPY PHASE - 17:26:12 ON 2007/09/10
FLM57000 - INITIATING PURGE PHASE - 17:26:12 ON 2007/09/10
FLM57001 - INITIATING PURGE FROM GROUP: DEVI025
FLM58000 - PROMOTE PROCESSOR COMPLETED - 17:26:12 ON 2007/09/10
***** Bottom of Data *****

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

```

Figure 6-47 ISPF-PT SCLM walk through

### 6.3.5 Removing the obsolete program

In this section, we show you how to remove the obsolete program.

We completed the Migrate, Build, and Promotion changes for our project. Figure 6-48 on page 166 depicts an easy way to switch to the ASM hierarchy and view a subset of members using the TAILOR command. The TAILOR command will display only members that were changed in the last 60 days.

```

File Display Library SCLM Settings Menu Utilities Test Help Exit
-----
-IPT--EDIT L1----- SOF AUD_DEVIO25.ARCHDEF -----ROW 00001 OF 00006-
COMMAND ==> type asm;tailor SCROLL ==> CSR
HOTBAR: REFRESH FLIP SORT CHA TAILOR COMPRESS EXPDIR INFO SET
*FILTER* *SORT*108 HIDDEN LIB2=INT420 LIB3=TEST420 LIB4=RLSE420
  NAME RENAME LIB VV.MM CREATED CHANGED SIZE INIT MOD USERID
$TSTASM -PROMOTE S 2 01.09 07/09/06 07/09/10 17:17 4 4 0 INTT125
$TSTJCL -PROMOTE S 2 01.06 07/09/06 07/09/10 17:17 7 4 0 INTT125
TSTSVC30 -SAVED S 2 01.08 07/09/06 07/09/10 17:17 10 10 0 INTT125
$$TSTALL S 2 01.00 07/09/06 07/09/06 10:54 5 5 0 INTT125
TSTSVC26 S 2 01.02 07/09/06 07/09/06 10:32 10 10 0 INTT125
TESTLONG S 1 01.01 04/08/19 07/08/17 10:58 11 11 0 INTT025
--END--

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

```

Figure 6-48 ISPF-PT SCLM walk through

As shown in Figure 6-49, enter a D to delete the TSTSVC26 ASM member.

```

File Display Library SCLM Settings Menu Utilities Test Help Exit
-----
-IPT--EDIT L1----- SOF AUD_DEVIO25.ASM -----ROW 00001 OF 00019-
COMMAND ==> SCROLL ==> CSR
HOTBAR: REFRESH FLIP SORT CHA TAILOR COMPRESS EXPDIR INFO SET
*FILTER* *SORT*325 HIDDEN LIB2=INT420 LIB3=TEST420 LIB4=RLSE420
  NAME RENAME LIB VV.MM CREATED CHANGED SIZE INIT MOD USERID
TSTSVC30 S 2 01.06 07/09/06 07/09/10 16:51 84 84 0 INTT125
AUDFLTYP S 3 01.32 99/12/24 07/09/10 14:26 841 390 0 INTT068
d TSTSVC26 S 2 01.01 07/09/06 07/09/06 10:28 84 84 0 INTT125
TESTLONG S 1 01.02 04/08/19 07/09/06 09:46 176 150 0 INTT125
AUDPBYP S 3 01.37 99/11/10 07/08/31 11:40 976 877 0 INTT065
AUDIDENT S 3 03.34 99/12/04 07/08/31 11:38 3624 2488 0 INTT065
AUDXPORT S 3 01.99 94/01/19 07/08/30 17:26 900 423 0 INTT068
AUDRPORT S 3 02.99 98/09/09 07/08/30 17:24 2573 1755 0 INTT068
AUDRIFID S 3 01.88 00/05/30 07/08/30 17:22 3841 2684 0 INTT068
AUDRIFUT S 3 02.25 00/01/17 07/08/30 17:22 3015 210 0 INTT068
AUDRIFCO S 3 01.69 00/02/06 07/08/30 17:20 1570 1393 0 INTT068
AUDRIFCI S 3 01.11 01/06/15 07/08/30 17:18 1579 1345 0 INTT068
AUDFILTM S 3 02.99 97/12/18 07/08/30 17:04 5936 3858 0 INTT068
AUDDIST S 3 02.99 97/10/28 07/08/30 17:00 2456 593 0 INTT068
AUDCMPSK S 3 01.42 01/08/28 07/08/30 16:56 3662 3620 0 INTT068

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

```

Figure 6-49 ISPF-PT SCLM walk through

If your settings request a delete confirmation, ISPF-PT prompts you prior to deleting the member, as shown in Figure 6-50 on page 167. The value of Y, circled in Figure 6-49, directs ISPF-PT to delete the SCLM statistics.

```

-IPT----- DELETE MEMBER CONFIRMATION: SOFTAUD.INT420.ASM-----
COMMAND ==> _

+-----+
| Delete SCLM accounting information?==> Y (Y=Yes,N=No) |
| Press ENTER to delete this member or the END key to cancel. |
+-----+

NAME RENAME LIB VV.MM CREATED .CHANGED. SIZE INIT MOD USERID
TSTSVC26 S 2 01.01 07/09/06 07/09/06 10:28 84 84 0 INTT125

LANGUAGE VER. CHANGE-CODE AUTH.-CODE BUILD-MAP PROMOTE-ID STATEMENTS INCLUDES
HLASM 1 V4R2M0 INTT125 83 0

***** TOP OF DATA *****
TSTSVC26 TITLE 'Test LOCATE SVC'
*****
* Execution JCL
* =====
*
* // EXEC PGM=IPIUTIL,PARM='<dsname>'
*****
*
* *****
F1=HELP F2=SPLIT F3=END F4=VIEW F5=RFIND F6=RCHANGE
F7=UP F8=DOWN F9=SWAP F10=LEFT F11=RIGHT F12=RETRIEVE

```

Figure 6-50 ISPF-PT SCLM walk through

Figure 6-51 confirms the member TSTSVC26 deletion.

```

File Display Library SCLM Settings Menu Utilities Test Help Exit
-----
-IPT--EDIT L1----- SOFTAUD.DEVI025.ASM ----- --MEMBER TSTSVC26 DELETED
COMMAND ==> _ SCROLL ==> CSR
HOTBAR: REFRESH FLIP SORT CHA TAILOR COMPRESS EXPDIR INFO SET
*FILTER* *SORT*325 HIDDEN LIB2=INT420 LIB3=TEST420 LIB4=RLSE420
NAME RENAME LIB VV.MM CREATED .CHANGED. SIZE INIT MOD USERID
AUDFLTVP S 3 01.32 99/12/24 07/09/10 14:26 841 390 0 INTT068
TESTLONG S 1 01.02 04/08/19 07/09/06 09:46 176 150 0 INTT125
AUDPBYV S 3 01.37 99/11/10 07/08/31 11:40 976 877 0 INTT065
AUDIDENT S 3 03.34 99/12/04 07/08/31 11:38 3624 2488 0 INTT065
AUDXPORT S 3 01.99 94/01/19 07/08/30 17:26 900 423 0 INTT068
AUDRPORT S 3 02.99 98/09/09 07/08/30 17:24 2573 1755 0 INTT068
AUDRIFID S 3 01.88 00/05/30 07/08/30 17:22 3841 2684 0 INTT068
AUDRIFUT S 3 02.25 00/01/17 07/08/30 17:22 3015 210 0 INTT068
AUDRIFCO S 3 01.69 00/02/06 07/08/30 17:20 1570 1393 0 INTT068
AUDRIFCI S 3 01.11 01/06/15 07/08/30 17:18 1579 1345 0 INTT068
AUDFILTM S 3 02.99 97/12/18 07/08/30 17:04 5936 3858 0 INTT068
AUDDIST S 3 02.99 97/10/28 07/08/30 17:00 2456 593 0 INTT068
AUDCMPSK S 3 01.42 01/08/28 07/08/30 16:56 3662 3620 0 INTT068
AUDCMPKB S 3 01.94 01/10/02 07/08/30 16:55 4556 3358 0 INTT068
AUDCMPID S 3 02.45 01/05/29 07/08/30 16:54 4762 3918 0 INTT068
F1=HELP F2=SPLIT F3=END F4=VIEW F5=RFIND F6=RCHANGE
F7=UP F8=DOWN F9=SWAP F10=LEFT F11=RIGHT F12=RETRIEVE

```

Figure 6-51 ISPF-PT SCLM walk through

Archived

# TSO Command Shell

In this chapter, we review the ISPF Productivity Tool functions that the TSO Command Shell provides.

The TSO Command Shell provides for two sets of command lists:

- ▶ History Command List
- ▶ Permanent Command List

Both sets of command lists save up to 999 commands. The TSO shell commands are retained after you log off from TSO and are available when you logon again. You can execute any of the 999 history commands from option 6. You can execute any of the 999 permanent commands from any ISPF panel.

## 7.1 Validating the TSO Command Shell options

In this section, we provide information about the ISPF-PT TSO Command Shell. However, if your options defined incorrectly, you will not see the same windows that we describe in this chapter.

To validate the TSO Shell options:

1. Enter ISET as a primary command.
2. Select the TSO shell options. You should see the window in Figure 7-1 on page 170.

```
-IPT- -----TSO shell options-----
COMMAND ==>

IBMIPT TSO command shell options:

Use IBMIPT TSO shell ==> Y (Y=Yes - Use the IBMIPT TSO shell)
                        (N=No - Use the standard ISPF TSO shell)

If IBMIPT shell active:

Initial screen      ==> H (H=History list, P=Permanent list)
Automatic filling   ==> Y (Y=Yes - Automatically fill permanent list, N=No)
Output line number ==> 7 (Number of screen line where the output of TSO
                        commands is displayed)

Press ENTER for options menu, END to exit, CANCEL for installation defaults.

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

Figure 7-1 ISET - TSO shell options

3. Be sure to set the **Use IBMIPT TSO Shell** to Y, which we circled in Figure 7-1.

**Note:** The Automatic filling option, circled in Figure 7-1 automatically enters your TSO Shell commands into the Permanent Command List, in the next available “empty slot”.

## 7.2 IPT History Commands List

Figure 7-2 on page 171 shows the History Command List. Up to 999 entries are stored automatically. When you enter a new command, the new command is **added** as the first entry, and the remaining commands are pushed down the list.



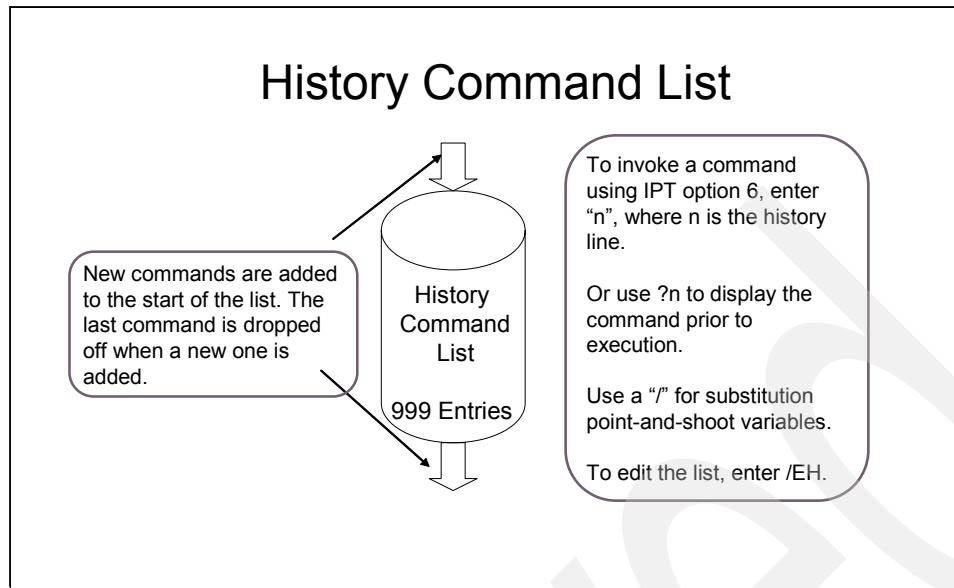


Figure 7-2 ISPF-PT History Commands

### 7.2.1 Using the TSO Command Shell with the History Command List

When you are in the TSO Command Shell, you can:

- ▶ Invoke a command by entering **n** on the TSO Command Shell, where **n** is the command number.
- ▶ Invoke a command by entering **?n** on the TSO Command Shell, which gives you an opportunity to modify the command prior to execution.
- ▶ Use a slash (/) as a variable. This variable is resolved with a point-and-shoot action. If the variable is not available, ISPF-PT prompts you for the variable.
- ▶ Update the History Command List with the **/EH** command.

## 7.3 ISPF-PT Permanent Commands List

Figure 7-3 on page 172 depicts the Permanent Command List. Up to 999 entries are stored automatically.

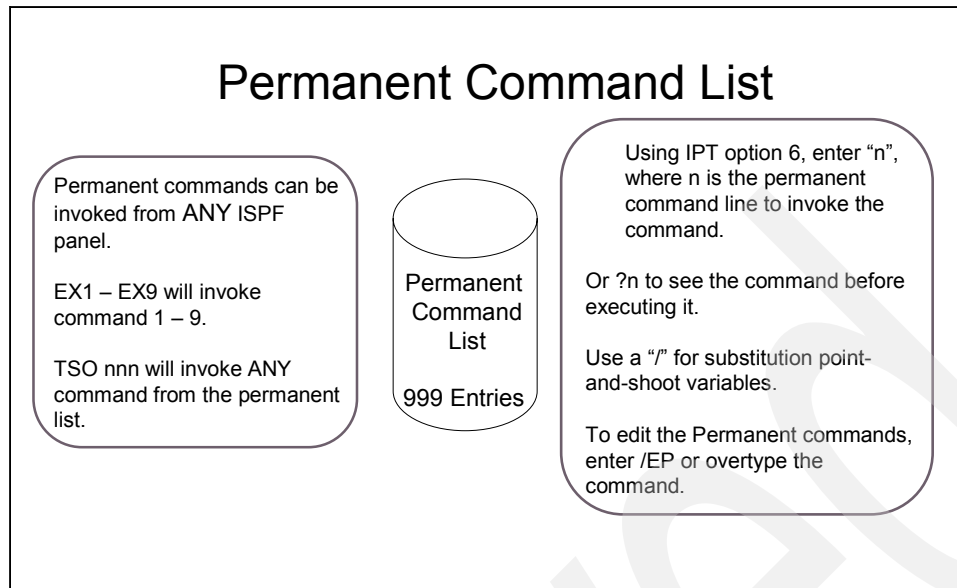


Figure 7-3 ISPF-PT History commands

### 7.3.1 Using the TSO Command Shell with the Permanent Command List

When you are in the TSO Command Shell, you can:

- ▶ Invoke a command by entering **n** on the TSO Command Shell, where **n** is the command number.
- ▶ Invoke a command by entering **?n** on the TSO Command Shell, which gives you an opportunity to modify the command prior to execution.
- ▶ Use a slash (/) as a variable. This variable is resolved with a point-and-shoot action. If the variable is not available, ISPF-PT prompts you for the variable.
- ▶ Update the Permanent Command List with the **/EH** command.
- ▶ No matter where you are in ISPF, you can invoke a permanent command by:
  - Typing **EXn**, where **n** is 1-9, representing one of the first nine permanent commands
  - Typing **TSO n**. **n** can be 1-999

**Note:** It is best to use TSO n for allocation commands, such as ALLOC, ALTLIB, or LIBDEF.

## 7.4 TSO Command Shell demonstration

ISPF-PT Option 6 displays the window in Figure 7-4 on page 173. From this window, you can enter:

<b>n</b>	Executes any of the 999 commands
<b>?n</b>	Displays any of the 999 commands prior to execution
<b>/IS</b>	Navigates to the ISPF Command Shell
<b>/EH</b>	Edits the History Command List
<b>/P</b>	Displays the Permanent Command List

<b>/EP</b>	Edits the Permanent Command List
<b>/SAV</b>	Saves the Command List shown
<b>/RES</b>	Restores the Command List shown

**Note:** The /SAV command saves the permanent or history list. However, when you exit ISPF, the History Command List and Permanent Command List are automatically saved.

Enter 2 to invoke the HOMETEST command, as shown in Figure 7-4.

```

-IPT-                                TSO COMMAND SHELL                                Row 1 to 11 of 115
(F18)=Standard SPF/TSO, PF10/F10=Set Linenum
Command ==> 2

Enter: TSO command, CLIST, REXX EXEC, or ISPEXEC statement.      Scroll => CSR
/IS standard ISPF/TSO, /P permanent
/EH edit history, /EP edit permanent, /SAV save, /RES restore
<c><entry number> where <c> is ? or / or omitted.

----- HISTORY COMMAND LIST -----
1  CCCA                                ->
2  hometest                           ->
3  racf                               ->
4  QEDIT                             ->
5  ALTLIB ACT DA('DNET424.EXEC') APPL(EXEC) ->
6  SELECT PGM(IQITSO) PARM(1) NEWAPPL(ISR) ->
7  ex16                               ->
8  omvs                               ->
9  obrowse                           ->
10 lista                             ->
11 ALLOC FILE(INPUT) DA('ACCT.TEST') SHR ->
F1=HELP    F2=SPLIT    F3=END      F4=RETURN    F5=RPFND    F6=RCHG
F7=UP      F8=DOWN     F9=SWAP     F10=LEFT    F11=RIGHT   F12=RETRIEVE

```

Figure 7-4 TSO Command Shell

Results of the HOMETEST command are shown in Figure 7-5.

```

-IPT-                                TSO COMMAND SHELL                                Row 1 to 11 of 112
*** HISTORY command list edited. ***
Command ==> 2

Enter: TSO command, CLIST, REXX EXEC, or ISPEXEC statement.      Scroll => CSR
EZA0619I Running IBM MVS TCP/IP CS V1R8 TCP/IP Configuration Tester

EZA9431I FTP.DATA file not found. Using hardcoded default values.

EZA0602I TCP Host Name is: DEMOMVS

EZA0605I Using Name Server to Resolve DEMOMVS
EZA0611I The following IP addresses correspond to TCP Host Name: DEMOMVS
EZA0612I 9.39.64.151

EZA0614I The following IP addresses are the HOME IP addresses defined in PROFILE
E
.TCPIP:
EZA0615I 9.39.64.151
EZA0615I 9.39.64.151
EZA0615I 127.0.0.1

***

```

Figure 7-5 Results of HOMETEST command

The ?n command, shown in Figure 7-6, gives you an opportunity to modify the data prior to execution.

```

-IPT-                                TSO COMMAND SHELL                                Row 1 to 11 of 115
IQIP628  HOMETEST executed. Return code is zero.
Command ==> ?5

Enter: TSO command, CLIST, REXX EXEC, or ISPEXEC statement.      Scroll => CSR
/IS standard ISPF/TSO, /P permanent
/EH edit history, /EP edit permanent, /SAV save, /RES restore
<c><entry number> where <c> is ? or / or omitted.

----- HISTORY COMMAND LIST -----
1  hometest                                ->
2  CCCA                                    ->
3  racf                                    ->
4  OEDIT                                  ->
5  ALTLIB ACT DA('DNET424.EXEC') APPL(EXEC) ->
6  SELECT PGM(IQITSO) PARM(1) NEWAPPL(ISR) ->
7  ex16                                    ->
8  omvs                                    ->
9  obrowse                                ->
10 lista                                  ->
11 ALLOC FILE(INPUT) DA('ACCT.TEST') SHR   ->
F1=HELP      F2=SPLIT      F3=END      F4=RETURN      F5=RPFND      F6=RCHG
F7=UP        F8=DOWN       F9=SWAP      F10=LEFT      F11=RIGHT     F12=RETRIEVE

```

Figure 7-6 Display and modify command prior to execution

You can modify the ALTLIB command prior to execution, as shown in Figure 7-7.

```

-IPT-                                TSO COMMAND SHELL                                Row 5 to 15 of 115
*** Press ENTER to execute the displayed command. ***
Command ==> ALTLIB ACT DA('DNET424.EXEC') APPL(EXEC)

Enter: TSO command, CLIST, REXX EXEC, or ISPEXEC statement.      Scroll => CSR
/IS standard ISPF/TSO, /P permanent
/EH edit history, /EP edit permanent, /SAV save, /RES restore
<c><entry number> where <c> is ? or / or omitted.

----- HISTORY COMMAND LIST -----
5  ALTLIB ACT DA('DNET424.EXEC') APPL(EXEC) ->
6  SELECT PGM(IQITSO) PARM(1) NEWAPPL(ISR) ->
7  ex16                                    ->
8  omvs                                    ->
9  obrowse                                ->
10 lista                                  ->
11 ALLOC FILE(INPUT) DA('ACCT.TEST') SHR   ->
12 ex                                      ->
13 HOMETEST                                ->
14 RPFND                                    ->
15 RCHG                                    ->
F1=HELP      F2=SPLIT      F3=END      F4=RETURN      F5=RPFND      F6=RCHG
F7=UP        F8=DOWN       F9=SWAP      F10=LEFT      F11=RIGHT     F12=RETRIEVE

```

Figure 7-7 Command available for modification

You can edit a single line from the History Command List by entering /n, where n is one of the History Command List line numbers. Use the /EH command to use ISPF to edit all of the History Command List lines, as shown in Figure 7-8 on page 175.

```

-IPT-                                TSO COMMAND SHELL                                Row 1 to 11 of 115
*** HISTORY command list left unchanged. ***
Command ==> /eh

Enter: TSO command, CLIST, REXX EXEC, or ISPEXEC statement.      Scroll => CSR
/IS standard ISPF/TSO, /P permanent
/EH edit history, /EP edit permanent, /SAV save, /RES restore
<c><entry number> where <c> is ? or / or omitted.

----- HISTORY COMMAND LIST -----
1  hometest ->
2  CCCA ->
3  racf ->
4  OEDIT ->
5  ALTLIB ACT DA('DNET424.EXEC') APPL(EXEC) ->
6  SELECT PGM(IQITSO) PARM(1) NEWAPPL(ISR) ->
7  ex16 ->
8  omvs ->
9  obrowse ->
10 lista ->
11 ALLOC FILE(INPUT) DA('ACCT.TEST') SHR ->
F1=HELP      F2=SPLIT      F3=END      F4=RETURN      F5=RPFND      F6=RCHG
F7=UP        F8=DOWN      F9=SWAP      F10=LEFT      F11=RIGHT     F12=RETRIEVE

```

Figure 7-8 Edit History Command List

The resulting Edit window for the History Command Shell is shown in Figure 7-9.

```

EDIT      .....TSO_Command_Shell....HISTORY-LIST      Columns 00001 00072
Command ==> Scroll ==> CSR
***** Top of Data *****
000001 hometest
000002 CCCA
000003 racf
000004 OEDIT
000005 ALTLIB ACT DA('DNET424.EXEC') APPL(EXEC)
000006 SELECT PGM(IQITSO) PARM(1) NEWAPPL(ISR)
000007 ex16
000008 omvs
000009 obrowse
000010 lista
000011 ALLOC FILE(INPUT) DA('ACCT.TEST') SHR
000012 ex
000013 HOMETEST
000014 RPFND
000015 RCHG
000016 SDSF ST DNET424*
000017 trsmain
000018 HELP TRSMAIN
000019 receive inda(/)
F1=Help      F2=Split      F3=Exit      F4=IPT View  F5=Rfind      F6=Rchange
F7=Up        F8=Down      F9=Swap      F10=Left    F11=Right     F12=Retrieve

```

Figure 7-9 Edit window for the History Command Shell

You can modify, insert, or delete the History Commands using standard ISPF editor commands.

You can view the Permanent Command List from any panel. Using option 6, you can enter /P, or you can enter EX? from *any* panel. An example is shown in Figure 7-10 on page 176.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (E) ----- LEVEL DNET424.ADLAB ----- "A" will display assist
Command ==> ex? SCROLL ==> CSR
Hotbar: OPRINT REFRESH CLRVOL FILLVOL UPDATE CUT FLIP VALIDATE
*TEMPORARY LIST*

Command Member Numbr Data Set Names / Objects Volume
-----
1 'DNET424.ADLAB.ADATA' DMPU20
2 'DNET424.ADLAB.ADATA.BIN' DMPU17
3 'DNET424.ADLAB.CNTL' DMPU29
4 'DNET424.ADLAB.CNTL.BIN' DMPU27
5 'DNET424.ADLAB.COBOL' DMPU26
6 'DNET424.ADLAB.COBOL.BIN' DMPU21
7 'DNET424.ADLAB.COPYLIB' DMPU15
8 'DNET424.ADLAB.COPYLIB.BIN' DMPU13
9 'DNET424.ADLAB.COPYLIBI' DMPU23
10 'DNET424.ADLAB.COPYLIBI.BIN' DMPU27
11 'DNET424.ADLAB.CUST.F4000' DMPU22
12 'DNET424.ADLAB.CUST.F4000.SPAC' DMPU06
13 'DNET424.ADLAB.CUST.ONEREC' DMPU18

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

```

Figure 7-10 Viewing the Permanent Command List

You can modify the commands that are listed in Figure 7-11 on this window, or you can modify them using the /EP command.

The command in line 1 invokes the SDSF ST command for my userID, DNET424.

The command in line 2, tso -tso invokes the ISPF command shell.

```

-IPT- TSO COMMAND SHELL Row 1 to 11 of 999
*** PERMANENT command list edited. ***
Command ==> _

Enter: TSO command, CLIST, REXX EXEC, or ISPEXEC statement. Scroll => CSR
/IS standard ISPF/TSO, /H history
/EH edit history, /EP edit permanent, /SAV save, /RES restore
<c><entry number> where <c> is ? or / or omitted.

----- PERMANENT COMMAND LIST -----
1 SDSF ST DNET424* ->
2 tso -tso ->
3 receive inda(/) ->
4 racf ->
5 CCCA ->
6 hometest ->
7 ISRDDN ->
8 obrowse ->
9 OEDIT ->
10 omvs ->
11 ISHELL ->

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

```

Figure 7-11 Permanent Command List

Put your cursor under data set number 2. The command RECEIVE INDA(/) is invoked by EX3 for the data set DNET424.ADLAB.ADATA.BIN, as shown in Figure 7-12.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (E) ----- LEVEL DNET424.ADLAB ----- RECEIVE RC(12)
Command ==> ex3 SCROLL ==> CSR
Hotbar: OPRINT REFRESH CLRVOL FILLVOL UPDATE CUT FLIP VALIDATE
*TEMPORARY LIST*

Command Member Numbr Data Set Names / Objects Volume
-----
1 'DNET424.ADLAB.ADATA' DMPU20
2 'DNET424.ADLAB.ADATA.BIN' DMPU17
3 'DNET424.ADLAB.CNTL' DMPU29
4 'DNET424.ADLAB.CNTL.BIN' DMPU27
5 'DNET424.ADLAB.COBOL' DMPU26
6 'DNET424.ADLAB.COBOL.BIN' DMPU21
7 'DNET424.ADLAB.COPYLIB' DMPU15
8 'DNET424.ADLAB.COPYLIB.BIN' DMPU13
9 'DNET424.ADLAB.COPYLIBI' DMPU23
10 'DNET424.ADLAB.COPYLIBI.BIN' DMPU27
11 'DNET424.ADLAB.CUST.F4000' DMPU22
12 'DNET424.ADLAB.CUST.F4000.SPAC' DMPU06
13 'DNET424.ADLAB.CUST.ONEREC' DMPU18

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

```

Figure 7-12 Permanent Command List with slash

The EX3 command receives the file successfully, as shown in Figure 7-13.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (E) ----- LEVEL DNET424.ADLAB ----- RECEIVE RC(12)
Command ==> ex3 SCROLL ==> CSR
Hotbar: OPRINT REFRESH CLRVOL FILLVOL UPDATE CUT FLIP VALIDATE
*TEMPORARY LIST*

INMR901I Dataset DNET424.ADLAB.ADATA from DNET424 on DEMOMVS
INMR154I The incoming data set is a 'DATA LIBRARY'.
INMR906A Enter restore parameters or 'DELETE' or 'END' +

```

Figure 7-13 Results of RECEIVE INDA(/) Command

## 7.5 Recalling commands

You can recall commands from any ISPF panel by using either of the following commands:

### EX?

This command will display the Permanent Command List. This is the same as Option 6, with the /P command. However, when you exit from the EX? command, you are returned to the original ISPF window.

<b>EX=</b>	This command recalls the last History Command and presents the History Command List.
<b>TSO =</b>	This command is identical to the <b>EX=</b> command described above. The <b>EX=</b> command does not have an imbedded space, but the <b>TSO =</b> command has an imbedded space.
<b>TSO ?</b>	This command presents the History Command List. However, it does not recall the last command that was executed.





## First time logon to the ISPF Productivity Tool

In this chapter, we describe the process of using the ISPF Productivity Tool (ISPF-PT) for the first time.

## 8.1 ISPF-PT LOGON procedure

The ISPF Productivity Tool requires a different LOGON procedure. Consult your z/OS System Programmer team to determine the correct LOGON procedure name to use for the ISPF Productivity Tool. Specify the LOGON procedure name on the TSO/E LOGON panel, as shown in Figure 8-1.

```
----- TSO/E LOGON -----

Enter LOGON parameters below:          RACF LOGON parameters:

Userid  ==> DNET424
Password ==>
Procedure ==> IPT
Acct Nmbr ==> 12345678
Size    ==> 120000
Perform ==>
Command ==>

Enter an 'S' before each option desired below:
      -Nomail      -Nonotice      -Reconnect      -OIDcard

PF1/PF13 ==> Help    PF3/PF15 ==> Logoff    PA1 ==> Attention    PA2 ==> Reshow
You may request specific help information by entering a '?' in any entry field
```

Figure 8-1 ISPF-PT LOGON procedure

We use the ISPF-PT LOGON procedure named IPT. Authorization might be required for the IPT procedure name. The LOGON procedure name is retained until the next time you log on to ISPF.

## 8.2 ISPF-PT persistent tables

The ISPF Productivity Tool maintains a significant amount of persistent tables, such as clipboards, Object Lists, MSL settings, and ISPF-PT settings. This data is retained after you logoff from ISPF. During your ISPF-PT session, most persistent tables are stored in a PDS or PDSE library that is allocated to the DD "IPITBLIB".

The ISPF-PT customization wizard defines how persistent tables are stored. Persistent tables might be:

- ▶ Defined during your first logon
- ▶ Pre-allocated by the z/OS system programmer
- ▶ No persistent tables defined

### 8.2.1 ISPF-PT persistent tables defined with first logon

The ISPF-PT customization wizard recommends that the allocation of persistent tables occurs during your first logon. If this option is set, you will see the window shown in Figure 8-2 on page 181 when you first log on to ISPF-PT. This window offers one of three choices:

- ▶ Specify an existing library for persistent tables

- Allocate a new library for persistent tables
- Operate without a persistent tables library

```

-IPT----- Persistent Table Management -----
COMMAND ==> _

  IBMIPT recommends a dedicated library (PDSE) for storing its
  persistent tables. DD(IPITBLIB) is currently not allocated or may
  be associated with an unusable dataset.
  * You may either specify an existing library or dynamically allocate a
  new library to be used for this purpose.
  * After successfully allocating a valid library, its name is stored in
  your profile and will be automatically used from now on, as long as
  DD(IPITBLIB) is not explicitly included in your LOGON procedure.
  * The library name first suggested by IBMIPT is either your already
  allocated permanent OLIST library, or if DD(PLSTLIBW) is missing,
  a name and volume derived from your PROFILE library name.
  Note: you may choose to operate without saving any IBMIPT persistent
  tables by declining to allocate DD(IPITBLIB).

  Main line commands:  D Disable    S Select (default)
  Press ENTER to process or F3      to cancel.

      LIBRARY NAME --> 'DNET424.ISPF.IPITBLIB'

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

```

Figure 8-2 Persistent table management window

ISPF-PT recommends a dedicated PDSE library to store persistent tables. ISPF-PT uses the naming convention “userid.ISPF.IPITBLIB” for the DD IPITBLIB allocation. To allocate a PDSE with this naming convention, press Enter. Note the data set name type “LIBRARY” in Figure 8-3.

```

Menu  RefList  Utilities  Help
-----
Allocate New Data Set
Command ==>
Data Set Name . . . : DNET424.ISPF.IPITBLIB
Management class . . . (Blank for default management class)
Storage class . . . (Blank for default storage class)
Volume serial . . . (Blank for system default volume) **
Device type . . . (Generic unit or device address) **
Data class . . . (Blank for default data class)
Space units . . . BLOCK (BLKS, TRKS, CYLS, KB, MB, BYTES
or RECORDS)
Average record unit _ (M, K, or U)
Primary quantity . . 320 (In above units)
Secondary quantity . 160 (In above units)
Directory blocks . . 50 (Zero for sequential data set) *
Record format . . . FB
Record length . . . 80
Block size . . . 6160
Data set name type . LIBRARY (LIBRARY, HFS, PDS, LARGE, BASIC, *
EXTREQ, EXTPREF or blank)
Expiration date . . . (YY/MM/DD, YYYY/MM/DD)

```

Figure 8-3 IPITBLIB allocation

Figure 8-3 defines the persistent table library as a PDSE. PDSE libraries have significant benefits, which we describe in 3.5, “Partitioned Data Set Extended” on page 86. Your installation may require data, such as Management Class or Storage Class. Complete the necessary fields, and press Enter to allocate to the library. The message in Figure 8-4 on page 182 confirms a successful persistent table library allocation.

```

Menu Utilities Compilers Options Status Help
z/OS Primary Option Menu IPITBLIB now available
Option ==>
0 Settings      Terminal and user parameters      User ID . . : DNET424
1 View          Display source data or listings      Time. . . . : 10:34
2 Edit          Create or change source data    Terminal. . : 3278
3 Utilities     Perform utility functions          Screen. . . : 1
4 Foreground    Interactive language processing      Language. . : ENGLISH
5 Batch         Submit job for language processing  Appl ID . . : ISP
6 Command       Enter TSO or Workstation commands   TSO logon : SPIFFY
7 Dialog Test   Perform dialog testing                TSO prefix: DNET424
P IBM Products  IBM program products                  System ID : DEMOMVS
10 SCLM         SW Configuration Library Manager      MVS acct. : 12345678
11 Workplace    ISPF Object/Action Workplace          Release . . : ISPF 5.8
12 z/OS System  z/OS system programmer applications
13 z/OS User    z/OS user applications

Enter X to Terminate using log/list defaults

```

Figure 8-4 IPITBLIB allocation

## 8.2.2 ISPF-PT persistent tables defined by the system programmer

The system programmer may pre-define your IPT Persistent Library for you. If this task is completed, no action is required on your part to use all ISPF-PT functions.

## 8.2.3 No ISPF-PT persistent tables defined by customization

If the Elect Persistent Table Use is set to N during the ISPF-PT customization, you might receive a message, such as \*\*\* HISTORY command list not saved. DD(IPITBLIB) unavailable. \*\*\*. To activate your persistent tables, use the ISET command, and enter L to navigate to the persistent table library options panel, as shown in Figure 8-5 and Figure 8-6 on page 183.

```

-IPT- -----Setting IBMIPT Defaults-----
COMMAND ==> 1
Select options by number, name, with cursor selection, or with line commands:
  IBMIPT is running under ISPF version 5.8

- A - ALL          - Select all the below displayed options
- M - MSL          - Member Selection List options
- P - OLIST        - Object list options
- G - GLOBAL        - Global edit and Findtext options
- R - PRINT         - Print options
- D - DSLIST        - DSLIST options
- T - TSO           - TSO shell options
- E - EDIT          - Edit, Browse and View options
- I - INTERFACE     - Specify user interface options
- N - DIAGNOSE      - Diagnose ISPF errors
- L - LIBRARY       - Persistent table library options

Make your selection and press the ENTER key or press the END key to exit

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

```

Figure 8-5 Setting ISPF-PT defaults

To allocate a Persistent Table library, enter an A, as shown in Figure 8-6.

```
-IPT- -----Persistent Table Library Options-----
COMMAND ==>

IBMPT uses DD(IPITBLIB) for specifying a personal library as a
repository of persistent tables, such as CUT/PASTE clipboards and
the TSO-SHELL command lists. This library may be dynamically allocated
or disabled by electing one of the following actions:

  A Allocate an existing or a new library for this purpose.

  B Backup the current table library and allocate space for a new one.

  T Temporarily disable its use for this session. At the next IBMPT startup
    you will be prompted again to elect how to use this facility.

  P Permanently disable its use. This facility will be disabled until you
    explicitly use the SET command to change this status.

  A <=== Elect persistent table library action.

Press ENTER for options menu, END to exit, CANCEL for installation defaults.
F1=HELP      F2=SPLIT      F3=END      F4=IPT View  F5=RFIND    F6=RCHANGE
F7=UP        F8=DOWN       F9=SWAP     F10=LEFT   F11=RIGHT   F12=RETRIEVE
```

Figure 8-6 Persistent table library options

Archived

# Customizing your ISPF keys for Point-and-Shoot

In this chapter, we show you how to define a Function Key to view files using the ISPF-PT Point-and-Shoot functions.

You can use the standard ISPF function keys to minimize the keystrokes that are required when you use the ISPF-PT Point-and-Shoot functions. In the following sections, we show you how to define the PF4 key initiate and the VIEW primary command when using SDSF. You can choose to define a PF key for a Browse or Edit command or any of the ISPF-PT shortcuts that the IPT? command shows. You can choose to define any function key to view the file.

## 9.1 Customizing ISPF-PT with View using PF4

Using ISPF-PT, you can view a file by typing `VIEW filename` on any primary window. You can also use the Point-and-Shoot capabilities to identify the file by placing your cursor on the file. If you define a PF key for the VI command, you do not have to type the VIEW command to view the file contents.

ISPF has different KEYLIST files that are used to define function keys for various windows. Defining a VIEW command on one window might not be applicable to other windows. The VIEW command is useful when:

- ▶ Viewing printed output using SDSF
- ▶ Using from an MSL

To define the Function Key you want to use:

1. Navigate to the window you want to set the PF keys, for example, SDSF.
2. Enter KEYS.
3. Update the PF key command.
4. Optionally, update the PF key label.
5. Exit from the PF Key Definition and Labels window.

### 9.1.1 Defining PF4 for the ISPF-PT View command for SDSF

In this section, we show you how to define a PF key to view files with SDSF. Because different KEYLISTS are used for an Object List and Member Selection Lists, repeat the following steps when you view the Object List and again when you view a Member Selection List.

Figure 9-1 depicts a job in the SDSF output queue.

Display Filter View Print Options Help									
-----									
SDSF OUTPUT ALL CLASSES ALL FORMS				LINES 16,878		LINE 1-7 (7)			
COMMAND INPUT ==> keys				SCROLL ==> CSR					
PREFIX=DNET424* DEST=(ALL) OWNER=* SORT=JobID/A SYSNAME=									
NP	JOBNAME	JobID	Owner	Prt	C	Forms	Dest	Tot-Rec	
DNET424A	JOB07078	DNET424		144	A	STD	LOCAL	334	
DNET424A	JOB07079	DNET424		144	A	STD	LOCAL	335	
DNET424A	JOB07082	DNET424		112	A	STD	LOCAL	7,781	
DNET424A	JOB07095	DNET424		112	A	STD	LOCAL	7,727	
DNET424B	JOB07114	DNET424		144	A	STD	LOCAL	334	
DNET424C	JOB07121	DNET424		144	A	STD	LOCAL	2	
DNET424C	JOB07121	DNET424		144	A	STD	LOCAL	365	
F1=HELP			F2=SPLIT		F3=END		F4=IPT View		F5=IFIND
F7=UP			F8=DOWN		F9=SWAP		F10=LEFT		F11=RIGHT
									F12=RETRIEVE

Figure 9-1 SDSF Queue – Invocation of PF Key Definition and Labels panel

The KEYS command displays the PF Key Definition and Label window, as shown in Figure 9-2. Change the PF4 setting to VI. Optionally, you can add a Label. PF3 to return to the SDSF list of output Jobs.

```

PF Key Definitions and Labels - Primary Keys
-----
Command ==>
Number of PF Keys . . . 24
Terminal type . . : 3278
More: +

PF1 . . . HELP
PF2 . . . SPLIT
PF3 . . . END
PF4 . . . VI
PF5 . . . IFIND
PF6 . . . BOOK
PF7 . . . UP
PF8 . . . DOWN
PF9 . . . SWAP
PF10 . . . LEFT
PF11 . . . RIGHT
PF12 . . . RETRIEVE

PF1 label . . . IPT View
PF4 label . . .
PF7 label . . .
PF10 label . . .
PF2 label . . .
PF5 label . . .
PF8 label . . .
PF11 label . . .
PF3 label . . .
PF6 label . . .
PF9 label . . .
PF12 label . . .

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

```

Figure 9-2 PF Key Definition and Label panel



Select a job from the output queue, as shown in Figure 9-3.

Display Filter View Print Options Help									
SDSF OUTPUT ALL CLASSES ALL FORMS					LINES 16,878		LINE 1-7 (7)		
COMMAND INPUT ==>					SCROLL ==> CSR				
PREFIX=DNET424* DEST=(ALL) OWNER=* SORT=JobID/A SYSNAME=									
NP	JOBNAME	JobID	Owner	Prty	C	Forms	Dest	Tot-Rec	
	DNET424A	JOB07078	DNET424	144	A	STD	LOCAL	334	
	DNET424A	JOB07079	DNET424	144	A	STD	LOCAL	335	
	DNET424A	JOB07082	DNET424	112	A	STD	LOCAL	7,781	
	DNET424A	JOB07095	DNET424	112	A	STD	LOCAL	7,727	
S	DNET424B	JOB07114	DNET424	144	A	STD	LOCAL	334	
	DNET424C	JOB07121	DNET424	144	A	STD	LOCAL	2	
	DNET424C	JOB07121	DNET424	144	A	STD	LOCAL	365	
F1=HELP F2=SPLIT F3=END F4=IPT View F5=IFIND F6=BOOK									
F7=UP F8=DOWN F9=SWAP F10=LEFT F11=RIGHT F12=RETRIEVE									

Figure 9-3 SDSF – Select Job from the Output Queue

The Find command locates the file and puts the cursor under the file name, as shown in Figure 9-4.

Display Filter View Print Options Help									
SDSF OUTPUT DISPLAY DNET424B JOB07114 DSID					2 LINE 0		COLUMNS 02- 81		
COMMAND INPUT ==> f cust2.ksds					SCROLL ==> CSR				
***** TOP OF DATA *****									
JES 2 JOB LOG -- SYSTEM M VSA -- N O D E									
09.47.21 JOB07114 ---- WEDNESDAY, 22 AUG 2007 ----									
09.47.21 JOB07114 IRR010I USERID DNET424 IS ASSIGNED TO THIS JOB.									
09.47.21 JOB07114 ICH70001I DNET424 LAST ACCESS AT 08:46:04 ON WEDNESDAY, AUGU									
09.47.21 JOB07114 \$HASP373 DNET424B STARTED - INIT 2 - CLASS A - SYS MVSA									
09.47.21 JOB07114 IEF403I DNET424B - STARTED - TIME=09.47.21									
09.47.21 JOB07114 - --TIMINGS (MINS.)--									
09.47.21 JOB07114 -JOBNAME STEPNAME PROCSTEP RC EXCP CPU SRB CLOCK									
09.47.21 JOB07114 -DNET424B CUSTKSDS CHECKV 00 45 .00 .00 .00									
09.47.21 JOB07114 -DNET424B CUSTKSDS ALLOCV FLUSH 0 .00 .00 .00									
09.47.21 JOB07114 -DNET424B CUSTKSDS COPYV FLUSH 0 .00 .00 .00									
09.47.21 JOB07114 -DNET424B CUSTKSDS CHECKV 00 48 .00 .00 .00									
09.47.21 JOB07114 -DNET424B CUSTKSDS ALLOCV FLUSH 0 .00 .00 .00									
09.47.21 JOB07114 -DNET424B CUSTKSDS COPYV FLUSH 0 .00 .00 .00									
09.47.21 JOB07114 -DNET424B VERIFY 00 37 .00 .00 .00									
09.47.21 CAZ0072I Sampling started for request number 00000513 by SSID									
F1=HELP F2=SPLIT F3=END F4=IPT View F5=IFIND F6=BOOK									
F7=UP F8=DOWN F9=SWAP F10=LEFT F11=RIGHT F12=RETRIEVE									

Figure 9-4 Locating a file in the SDSF output

Figure 9-5 shows where the file DNET424.ADLAB.CUST2.KSDS is located. Use PF4 to View the file.

```

Display Filter View Print Options Help
-----
SDSF OUTPUT DISPLAY DNET424B JOB07114 DSID      3 LINE  CHARS 'CUST2.KSDS' FOUN
COMMAND INPUT ==>                                SCROLL ==> CSR
BSTITUTION JCL - DSN=DNET424.ADLAB.CUST2.KSDS, DISP=(NEW,CATLG,DELETE), SPACE=(CYL
YLEN=13,RECOrg=KS
EXEC PGM=IDCAMS,COND=(0,EQ,CHECKV)
DD DSN=&SYSUID..&FILNAM,DISP=SHR
BSTITUTION JCL - DSN=DNET424.ADLAB.CUST2.KSDS,DISP=SHR
DD DSN=&SYSUID..ADLAB.FILES(CUST2),DISP=SHR
BSTITUTION JCL - DSN=DNET424.ADLAB.FILES(CUST2),DISP=SHR
DD SYSOUT=*

IN      DD *

CATE ADLAB.CUST2.WORK.KSDS
*****
EXEC ALLOCVS,FILNAM='ADLAB.CUST2.WORK.KSDS',
XSPACE='CYL,(1,1)',
XLRECL=2048,XKEYOFF=0,
XKEYLEN=13,XRECOrg=KS
F1=HELP      F2=SPLIT      F3=END      F4=IPT View  F5=IFIND      F6=BOOK
F7=UP        F8=DOWN      F9=SWAP     F10=LEFT   F11=RIGHT   F12=RETRIEVE

```

Figure 9-5 SDSF output: VSAM file located

**Note:** ISPF-PT does not resolve Symbolic information, for example, selecting &SYSUID.&FILNAM does not display the file.

ISPF-PT is customized to invoke the File Manager product when you browse or view a VSAM file, as shown in Figure 9-6.

```

Process Options Help
-----
Browse          DNET424.ADLAB.CUST2.KSDS                      Rec 0 of 100
Command ==>                                           Scroll PAGE
Type KSDS      Key          + RBA          Format CHAR
Col 1
<==+==1==>+---2---+---3---+---4---+---5---+---6---+---7---+
**** Top of data ****
01001C      Lynn, Amanda      .....Spirit Lake      Musician
01001PAG00487Acoustic guitar      2004-05-13..2006-03-30
01001PVN00048Violin      2004-03-25..2004-03-26
02200C      Graham, Anna      ../.*..Atwon      Cryptographer
02200PBG00459Bass Guitar      1996-04-30..2003-06-14
02200PDS00099Starter Drum Set      2001-09-04..2002-10-05
02200PEG00057Electric Guitar      2003-07-31..2004-03-30
02200PEP00090Electric Piano      2005-05-27..2005-09-27
02200PMC00400Microphone      2003-03-15..2004-01-13
02200PPA00760Amplifier 11+      1999-04-08..2001-10-25
02202C      Major, Art      ....%..Harmon      College student
02202PCL00857Cello      1998-06-30..2000-02-30
02202PDB00297Double Bass      2003-09-25..2006-01-04
02202PVN00956Violin      1997-08-14..2003-00-31
F1=Help      F2=Zoom      F3=Exit      F4=Expand      F5=RFind      F7=Up
F8=Down      F9=Swap      F10=Left   F11=Right   F12=Retrieve

```

Figure 9-6 File Manager invoked with PF4 – View

A prior FIND command positioned the cursor on the second qualifier of the file. ISPF-PT recognizes this file name and appends the user DNET424 prefix to the file name when using PF4, as shown in Figure 9-7 on page 189.

```

Display Filter View Print Options Help
-----
SDSF OUTPUT DISPLAY DNET424B JOB07114 DSID      3 LINE CHARS '(CUS' FOUND
COMMAND INPUT ==>
//OLDFILE DD DSN=&SYSUID. ADLAB.FILES(CUST2), DISP=SHR
//SYSPRINT DD SYSOUT=*
//*
//      PEND
//*
//*
//*
//*      ALLOCATE ADLAB.CUST2.KSDS
//*      *****
3 //CUSTKSDS EXEC ALLOCVS, FILNAM='ADLAB.CUST2.KSDS',
//      XSPACE='CYL, (1,1)',
//      XLRECL=2048, XKEYOFF=0,
//      XKEYLEN=13, XRECORD=KS
4 ++ALLOCVS PROC FILNAM='NONAME', XSPACE='CYL, (1,1)',
++      XLRECL=2048, XKEYOFF=0, XKEYLEN=10, XRECORD=KS
+++
+++ CHECK TO SEE IF FILE IS ALREADY ALLOCATED
F1=HELP  F2=SPLIT  F3=END  F4=IPT View  F5=IFIND  F6=PF4
F7=UP    F8=DOWN  F9=SWAP F10=LEFT  F11=RIGHT F12=

```

Figure 9-7 SDSF output: Sequential file located

ISPF-PT displays the file DNET424.ADLAB.FILES(CUST2), as shown in Figure 9-8.

```

File Edit Edit_Settings Menu Utilities Compilers Test Help
-----
-IPT- VIEW DNET424.ADLAB.FILES(CUST2) - 01.00 Columns 00001 00072
Command ==> Scroll ==> CSR
***** Top of Data *****
000001 01001C Lynn, Amanda f8 Spirit Lake Musician
000002 01001PAG00487Acoustic guitar 2004-05-13 2006-03-30
000003 01001PVN00048Violin 2004-03-25 2004-03-26
000004 02200C Graham, Anna / * Atwon Cryptographer
000005 02200PBG00459Bass Guitar 1996-04-30 2003-06-14
000006 02200PDS00099Starter Drum Set 2001-09-04 2002-10-05
000007 02200PEG00057Electric Guitar 2003-07-31 2004-03-30
000008 02200PEP00090Electric Piano 2005-05-27 2005-09-27
000009 02200PMC00400Microphone 2003-03-15 2004-01-13
000010 02200PPA00760Amplifier 11+ 1999-04-08 2001-10-25
000011 02202C Major, Art a% Harmon College student
000012 02202PCL00857Cello 1998-06-30 2000-02-30
000013 02202PDB00297Double Bass 2003-09-25 2006-01-04
000014 02202PVN00956Violin 1997-08-14 2003-00-31
000015 03003C Prentice, Anna Ñ Laramie New hire
000016 03003PFH00785French Horn 2005-07-05 2006-12-27
000017 03003POB00138Oboe 2006-06-30 2006-01-15
F1=Help F2=Split F3=Exit F4=IPT View F5=Rfind F6=Rchange
F7=Up F8=Down F9=Swap F10=Left F11=Right F12=Retrieve

```

Figure 9-8 IPT View invoked

## 9.1.2 Using the Point-and-Shoot commands with a Member Selection List

In this section, PF4 invokes the VIEW command. Select the \$INDEX member from the DNET424.MASTER.ADLAB.INSTALL data set. This member contains a list of important members within this PDS, as shown in Figure 9-9 on page 190.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (B) ----- LEVEL DNET424.MASTER ----- Row 28 to 40 of 41
Command ===> 28 V $index SCROLL ==> CSR
Hotbar: OPRINT REFRESH CLRVOL FILLVOL UTIL UPDATE CUT FLIP
*TEMPORARY LIST*

TSO PARMS ==>
Command Member Numbr Data Set Names / Objects Class
-----
28 'DNET424.MASTER.ADLAB.INSTALL' PDSE
29 'DNET424.MASTER.ADLAB.JCL'
30 'DNET424.MASTER.ADLAB.LISTING'
31 'DNET424.MASTER.ADLAB.LOAD'
32 'DNET424.MASTER.ADLAB.LOAD2'
33 'DNET424.MASTER.ADLAB.PSBLIB'
34 'DNET424.MASTER.ADLAB.PSBSOURC'
35 'DNET424.MASTER.ADLAB.SOURCE' PDSE
36 'DNET424.MASTER.ADLAB.SOURCE2' PDSE
37 'DNET424.MASTER.ADLAB.SYSDEBUG'
38 'DNET424.MASTER.ADLAB.TEMPLATE'
39 'DNET424.MASTER.ADLAB.VIEW'
40 'DNET424.MASTER.DBGT00L.SAVEBPS' PDSE

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

```

Figure 9-9 View the \$INDEX member

With ISPF-PT, you can view data sets or Members using the VIEW command. When you are in an MSL, or a Hierarchy, you can view members by putting your cursor on the member name and pressing PF4 to invoke the VIEW command, as shown in Figure 9-10.

```

File Edit Edit_Settings Menu Utilities Compilers Test Help
-----
-IPT- VIEW DNET424.MASTER.ADLAB.INSTALL($INDEX) - 01.27 Columns 00001 00072
Command ==> Scroll ==> CSR
***** Top of Data *****
000001 TOOL LE EXITS LINKED IN. THIS JOB IS OPTIONAL.
000002 $$README POINTS TO LABINSTALLATIONINSTRUCTIONS.DOC
000003 $$SET IS UPDATED BY THE INSTALLER - POINTS TO LOCAL FILES
000004 $$VER VERSION OF THIS INSTALL PDS
000005 $CREATE JCL TO CREATE THE ADLAB.INSTALL.XMIT INSTALL FILE
000006 $INDEX DESCRIPTIONS OF MEMBERS OF THIS PDS
000007 $JOB CARD SAMPLE JOB CARD
000008 $PREPAPA JOB TO RUN PRIOR TO APA LABS TO SET UP FILES
000009 $PREPCII JOB TO RUN TO COMPILE SAMPLE COBOL II PROGRAMS
000010 $PREPCOS JOB TO RUN TO COMPILE SAMPLE OS/VS COBOL PROGRAMS
000011 $PREPDT JOB TO RUN PRIOR TO DEBUG TOOL LABS TO SET UP FILES
000012 $PREPDTA JOB TO RUN PRIOR TO DEBUG TOOL ASSEMBLER LABS TO SET UP
000013 $PREPDTC JOB TO RUN PRIOR TO DEBUG TOOL CICS LABS TO SET UP FILES
000014 $PREPDTE JOB TO RUN TO CREATE COPIES OF LE MODULES WITH THE DEBUG
000015 $PREPFA JOB TO RUN PRIOR TO FAULT ANALYZER LABS TO SET UP FILES
000016 $PREPFM JOB TO RUN PRIOR TO FILE MGR / IMS LABS TO SET UP FILES
000017 $PREPFM JOB TO RUN PRIOR TO FILE MGR LABS TO SET UP FILES
F1=Help F2=Split F3=Exit F4=IPT View F5=Rfind F6=
F7=Up F8=Down F9=Swap F10=Left F11=Right F12=

```

Figure 9-10 Using the Point-and-Shoot VIEW command within an MSL member

Member viewed with PF4 is shown in Figure 9-11.

```

File Edit Edit_Settings Menu Utilities Compilers Test Help
-IPT- VIEW DNET424.MASTER.ADLAB.INSTALL($PREPAPA) - 01.20 Columns 00001 00072
Command ==> Scroll ==> CSR
***** ***** Top of Data *****
000001 //DNET424A JOB (JIM),'FA UTILITY',
000002 //          REGION=OK,NOTIFY=DNET424,
000003 //          MSGCLASS=A,
000004 //          CLASS=A
000005 /**      - - - ADD A JOB CARD HERE - - -
000006 /**
000007 /**      Prepare Files and Programs needed for the      ***
000008 /**      Application Performance Analyzer workshop      *****
000009 /**
000010 /**
000011 /**      I N S T R U C T I O N S :
000012 /**
000013 /**      1) Add a JOB card. *****
000014 /**          Make sure that the job will run on the same z/OS system
000015 /**          that participants will be using during the class.
000016 /**
000017 /**      2) Modify the following "JCLLIB" statement
F1=Help      F2=Split      F3=Exit      F4=IPT View      F5=Rfind      F6=Rchange
F7=Up        F8=Down      F9=Swap      F10=Left      F11=Right      F12=Retrieve

```

Figure 9-11 Member viewed with PF4 key

Archived



## **Installing ISPF Productivity Tool V5.9 out-of-the box on the z/OS environment**

In this chapter, we describe the installation procedures for the ISPF Productivity Tool on z/OS environment.

IBM delivers ISPF-PT V5.9 in tapes (CDBPO - Custom-Built Product Delivery Offering) to you at the address specified during the ordering process.

## 10.1 Installation requirements for ISPF-PT V5.9

The following system software is required for the installation and operation of the ISPF Productivity Tool Release 5.9:

- ▶ z/OS Version 1 Release 4 or later operating system
- ▶ ISPF Version 5 Release 2 or later

Installing ISPF Productivity Tool does not require an IPL or authorized mode. While you invoke the ISPF Productivity Tool, use libraries that are pointed to by the TSO LOGON procedure or CLIST.

Hence, we can install ISPF-PT on a system-wide basis or only for selected programmers. The ISPF Productivity Tool code is fully reentrant. Most of the ISPF Productivity Tool code resides above the 16-megabyte line. Mostly, ISPF Productivity Tool acquires working storage above the line.

## 10.2 Contents of the ISPF-PT product package

In this section, we cover the contents of the ISPF-PT product package.

### 10.2.1 Custom-built product delivery offering tapes

A custom-built product delivery offering tape (CBPDO) is usually for a particular feature (CICS, database system, MVS, or NCP). These features correspond to the SRELs to which products are applicable. A CBPDO tape contains the PTFs, HOLDDATA, and preventive service planning (PSP) upgrade files to bring your system up to a service level that you select.

The CBPDO tape is a standard label (SL) tape, with files arranged as shown in Table 10-1.

Table 10-1 Format of a CBPDO tape

File number	Processed by SMP/E	Contents
	No	Documents
	No	Installation RIMs
	Yes	HOLDDATA for exception SYSMODs
	No	Program directories and PSP information
	Yes	SMPMCS file (MCS statements for SYSMODs on the tape), PTFs, and cover letters
6 – 14	Yes	RELFILES for products on the tape

CBPDO contains product and services on a single logical tape. It contains the following installation materials.

- ▶ Sample jobs to receive products and service
- ▶ Program directories for the products you ordered
- ▶ Installation manuals for the products you ordered



- ▶ ISPF-PT V5.9 CDBPO package delivered to customer contains these items:
  - Relative file (RELFILE) tape
  - Object code only (OCO)
  - Source code (optional)
- ▶ Program directory (installation manual)

## Relative file tape

A *relative file* tape, or *RELFILE* tape, is a standard label tape made up of two or more files. It contains a file of the MCS for one or more functions and one or more relative files that contain unloaded source data sets and unloaded link-edited data sets that contain executable modules. The relative files might also contain other data, such as sample procedures. These unloaded partitioned data sets are in a format for installation on a z/OS system or subsystem by SMP/E.

The product distribution tape contains the following files in JES2 XMIT format, as shown in Table 10-2. The file attributes are those of the resulting files when loaded onto disk. Use the TSO RECEIVE command to process.

Table 10-2 Contents of product distribution tape

File	Library
Relative File 1	AIQIINST library: JCL and other members required to install and maintain IPT File name: IBM.HIQI590.F1.XMIT RECFM=FB, LRECL=80, BLKSIZE=8800
Relative File 2	AIQICLIB CLIST library: CLISTs used by IPT in fixed-length format File name: IBM.HIQI590.F2.XMIT RECFM=FB, LRECL=80, BLKSIZE=8800
Relative File 3	AIQICLBV library: same as File 2, but in variable-length format File name: IBM.HIQI590.F3.XMIT RECFM=VB, LRECL=251, BLKSIZE=8800
Relative File 4	AIQILOAD library: IPT load modules for private use File name: IBM.HIQI590.F4.XMIT RECFM=U, LRECL=0, BLKSIZE=6144
Relative File 5	AIQILPA library: IPT load modules for LPA File name: IBM.HIQI590.F5.XMIT RECFM=U, LRECL=0, BLKSIZE=6144
Relative File 6	AIQIMLIB IPT message library File name: IBM.HIQI590.F6.XMIT RECFM=FB, LRECL=80, BLKSIZE=8800
Relative File 7	AIQIPLIB IPT panel library File name: IBM.HIQI590.F7.XMIT RECFM=FB, LRECL=80, BLKSIZE=8800
Relative File 8	AIQISLIB IPT skeleton library File name: IBM.HIQI590.F8.XMIT RECFM=FB, LRECL=80,
Relative File 9	AIQITLIB IPT ISPF table library File name: IBM.HIQI590.F9.XMIT RECFM=FB, LRECL=80,

## Program directory (installation manual)

The program directory (installation manual) is a document shipped with each release of a product. Its primary purpose is to document the installation of the product.

The program directory is part of the informal documentation of the product. It does not contain changes that are not related to installation or detailed information that is available in a technical newsletter or replacement publication.

## Contents of the Program Directory

The program directory model explains what information is to be included in the program directory. The program directory performs the following functions:

- ▶ Describes all of the machine-readable material and publications
- ▶ Documents which systems, concurrent programs, and machines are required
- ▶ Provides details on how to install the product
- ▶ Documents the support that is available for the product
- ▶ Identifies program and service levels when communicating with personnel
- ▶ Identifies the resources needed to install the program and the impact of its use on an existing data processing system.

## 10.3 ISPF Productivity Tool installation

You must install and maintain the delivered product code using SMP/E, as described in the Program Directory for IBM ISPF Productivity Tool V5.9.0, program number 5698-A81, FMID HIQI590. Table 10-3 and Table 10-4 list the target and distribution libraries (data sets) used by ISPF Productivity Tool and their contents.

Table 10-3 *Distribution data set descriptions*

Library	Description
AIQICLBV	IPT CLIST library (variable blocked)
AIQICLIB	IPT CLIST library (fixed blocked)
AIQIINST	IPT installation and maintenance sample JCL library
AIQILOAD	IPT Linklist load library
AIQILPA	IPT LPA load library
AIQIMLIB	IPT ISPF-message library
AIQIPLIB	IPT ISPF-panel library
AIQISLIB	IPT ISPF-skeleton library
AIQITLIB	IPT ISPF-table library

Table 10-4 *Target data set descriptions*

Library	Description
SIQICLBV	IPT CLIST library (variable blocked)
SIQICLIB	IPT CLIST library (fixed blocked)
SIQIINST	IPT installation and maintenance sample JCL library
SIQILOAD	IPT Linklist load library
SIQILPA	IPT LPA load library
SIQIMLIB	IPT ISPF-message library
SIQIPLIB	IPT ISPF-panel library
SIQISLIB	IPT ISPF-skeleton library

Library	Description
SIQICLBV	IPT CLIST library (variable blocked)
SIQITLIB	IPT ISPF-table library

### 10.3.1 Sample Jobs

Table 10-5 shows sample installation jobs that are part of the product. It is useful during installation of ISPF Productivity Tool.

*Table 10-5 Sample Installation Jobs*

Job Name	Job Type	Description	RELFILE
IQIJRECV	RECEIVE	Sample RECEIVE job	IBM.HIQI590.F1
IQIJALLO	ALLOCATE	Sample job to allocate target and distribution libraries	IBM.HIQI590.F1
IQIJDDDF	DDDEF	Sample job to define SMP/E DDDEFs	IBM.HIQI590.F1
IQIJAPP	APPLY	Sample APPLY job	IBM.HIQI590.F1
IQIJACC	ACCEPT	Sample ACCEPT job	IBM.HIQI590.F1

The chosen mainframe enterprise has multiple LPARs running different z/OS releases and we wanted to use the same IPT target libraries on all LPARs regardless of their z/OS level. The ISPF Productivity Tool is independent of z/OS and ISPF release. It dynamically enables or disables any ISPF Productivity Tool functions that are dependent on certain z/OS or ISPF releases. We consider maintaining a dedicated SMP/E global zone that serves as single point of ISPF Productivity Tool maintenance for all of the z/OS images of your enterprise, which is termed as the ISPF-PT single point of maintenance option.

To access the sample installation jobs:

1. Perform an SMP/E RECEIVE, and then copy the jobs from the RELFILES to a work data set for editing and submission. See Table 10-5 to find the appropriate RELFILE data set.
2. Copy the files #3, #5, and #6 to #14, which we mentioned in Table 10-1 on page 194, to DASD on the LPAR.
3. The job in Figure 10-1 on page 198 shows the JCL statements to use to copy the contents of RELFILE 1 into DASD. Use either the //TAPEIN or the //FILEIN DD statement, depending on the distribution medium, and comment out or delete the other statement.
4. Add a job card, and change the lowercase parameters to uppercase values to meet your site's requirements before you submit. Figure 10-1 on page 198 is an example.

```
//STEP1      EXEC PGM=IEBCOPY
//SYSPRINT   DD SYSOUT=*
//TAPEIN     DD DSN=IBM.HIQI590.F1,UNIT=tunit,
//           VOL=SER=volser,LABEL=(x,SL) ,
//           DISP=(OLD,KEEP)
//FILEIN     DD DSN=IBM.HIQI590.F1,UNIT=SYSALLDA,DISP=SHR,
//           VOL=SER=filevol
//OUT        DD DSNAME=jcl-library-name,
//           DISP=(NEW,CATLG,DELETE),
//           VOL=SER=dasdvol,UNIT=SYSALLDA,
//           SPACE=(TRK,(5,1,3))
//SYSUT3     DD UNIT=SYSALLDA,SPACE=(CYL,(1,1))
//SYSIN DD *
//           COPY INDD=xxxxIN,OUTDD=OUT
/*
```

Figure 10-1 Sample job

In Figure 10-1, update the statements as follows:

- ▶ If using TAPEIN:
  - **tunit** is the unit value that matches the product tape.
  - **volser** is the volume serial matching the product tape.
  - **x** is the tape file number where the data set name is on the tape.
  - Refer to the documentation provided by CBPDO to see where IBM.HIQI590.F1 is on the tape.
- ▶ If using FILEIN:
  - **filevol** is the volume serial of the DASD device where the downloaded files reside.
- OUT
  - **jcl-library-name** is the name of the output data set where the sample jobs are stored.
  - **dasdvol** is the volume serial of the DASD device where the output data set will reside.
- SYSIN
  - **xxxxIN** is either TAPEIN or FILEIN depending on your input DD statement.

### 10.3.2 Allocating SMP/E global zone CSI libraries

Edit and submit sample job IQIJALLO to allocate the SMP/E target and distribution libraries for the ISPF Productivity Tool. Consult the instructions in the sample job for more information.

We used the JCL to delete, define, and prime the VSAM files for Global Zone CSI, as shown in Figure 10-2 on page 199.

```

//INTT025G JOB (1),'GLOBAL ZONE',CLASS=B,
// NOTIFY=&SYSUID,
// TYPRUN=HOLD,
// MSGCLASS=H,MSGLEVEL=(1,1)
//DELDEF1 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
//*DELETE IQI.V5R9M0.GLOBAL.CSI /* <==2 */

DEFINE CLUSTER(NAME(IQI.V5R9M0.GLOBAL.CSI) /* <==2 */ -
              FREESPACE(20, 5) -
              KEYS(24 0) -
              RECORDSIZE(24 143) -
              SHAREOPTIONS(2) -
              UNIQUE -
              VOLUME(Z18SMP)) /* <==1 */ -
DATA(NAME(IQI.V5R9M0.GLOBAL.CSI.DATA) /* <==2 */ -
      CONTROLINTERVALSIZE(4096) -
      CYLINDER(10 1)) -
INDEX(NAME(IQI.V5R9M0.GLOBAL.CSI.INDEX) /* <==2 */ -
       CONTROLINTERVALSIZE(1024) -
       TRACK(15 1) -
       IMBED)
/*
//PRIMCSI2 EXEC PGM=IDCAMS
//SMPCSI DD DSN=IQI.V5R9M0.GLOBAL.CSI,DISP=SHR /* <==2 */
//ZPOOL DD DSN=SYS1.MACLIB(GIMZPOOL),DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
      REPRO OUTFILE(SMPCSI) INFILE(ZPOOL)
//

```

Figure 10-2 JCL

Expected return codes and messages: The job is considered successful if a return code of 0 is received.

### 10.3.3 Allocating SMP/E target and distribution zone libraries

We used the JCL to delete, define, and prime VSAM target and distribution libraries, as shown in Figure 10-3 on page 200.

```

//INTT025Z JOB (1),'T/D ZONES',CLASS=B,
// NOTIFY=&SYSUID,
// MSGCLASS=O,MSGLEVEL=(1,1)
//DELDEF1 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DELETE IQI.V5R9M0.TZONE.CSI /* <==2,3 */
DELETE IQI.V5R9M0.DZONE.CSI /* <==2,4 */

DEFINE CLUSTER(NAME(IQI.V5R9M0.TZONE.CSI) /* <==2,3 */ -
FREESPACE(20, 5) -
KEYS(24 0) -
RECORDSIZE(24 143) -
SHAREOPTIONS(2) -
UNIQUE -
VOLUME(Z18R52)) /* <==1 */ -
DATA(NAME(IQI.V5R9M0.TZONE.CSI.DATA) /* <==2,3 */ -
CONTROLINTERVALSIZE(4096) -
CYLINDER(10 1)) -
INDEX(NAME(IQI.V5R9M0.TZONE.CSI.INDEX) /* <==2,3 */ -
CONTROLINTERVALSIZE(1024) -
TRACK(15 1) -
IMBED)

DEFINE CLUSTER(NAME(IQI.V5R9M0.DZONE.CSI) /* <==2,4 */ -
FREESPACE(20, 5) -
KEYS(24 0) -
RECORDSIZE(24 143) -
SHAREOPTIONS(2) -
UNIQUE -
VOLUME(Z18D52)) /* <==1 */ -
DATA(NAME(IQI.V5R9M0.DZONE.CSI.DATA) /* <==2,4 */ -
CONTROLINTERVALSIZE(4096) -
CYLINDER(10 1)) -
INDEX(NAME(IQI.V5R9M0.DZONE.CSI.INDEX) /* <==2,4 */ -
CONTROLINTERVALSIZE(1024) -
TRACK(15 1) -
IMBED)
/*
//PRIMCSI2 EXEC PGM=IDCAMS
//SMPCSI DD DSN=IQI.V5R9M0.TZONE.CSI,DISP=SHR /* <==2,3 */
//ZPOOL DD DSN=SYS1.MACLIB(GIMZPOOL),DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
REPRO OUTFILE(SMPCSI) INFILE(ZPOOL)
/*
//PRIMCSI3 EXEC PGM=IDCAMS
//SMPCSI DD DSN=IQI.V5R9M0.DZONE.CSI,DISP=SHR /* <==2,4 */
//ZPOOL DD DSN=SYS1.MACLIB(GIMZPOOL),DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
REPRO OUTFILE(SMPCSI) INFILE(ZPOOL)
/*
//

```

Figure 10-3 JCL

Expected return codes and messages: The job is considered successful if a return code of 0 is received.

#### **10.3.4 Allocating SMP/E temporary libraries for global zone**

We used the JCL to allocate the needed SMP/E temporary libraries, while setting up a new global zone for installation and maintenance, as shown in Figure 10-4 on page 202 and Figure 10-5 on page 203.

```

//INTT025A JOB (1),'GLOBAL ZONE LIBS',CLASS=B,
// NOTIFY=&SYSUID,
// MSGCLASS=O,MSGLEVEL=(1,1)
//SMPALLOC PROC SMPHLQ=,SMPSLQ=,SMPUNIT=,SMPVOL=
//ALLOC EXEC PGM=IEFBR14
//SYSPRINT DD SYSOUT=*
//*****
//* SMP/E TEMP LIBRARIES FOR USE WITH TIVOLI 390 PRODUCTS **
//*****
//SMPMTS DD DSN=&SMPHLQ..&SMPSLQ.SMPMTS,
// SPACE=(8800,(200,40,10)),
// DISP=(NEW,CATLG,DELETE),
// UNIT=&SMPUNIT,
// VOL=SER=&SMPVOL,
// DCB=(LRECL=80,RECFM=FB,BLKSIZE=0)
//*
//SMPPTS DD DSN=&SMPHLQ..&SMPSLQ.SMPPTS,
//***** SPACE=(8800,(200,40,10)),
// SPACE=(8800,(400,80,40)),
// DISP=(NEW,CATLG,DELETE),
// UNIT=&SMPUNIT,
// VOL=SER=&SMPVOL,
// DCB=(LRECL=80,RECFM=FB,BLKSIZE=0)
//*
//SMPLTS DD DSN=&SMPHLQ..&SMPSLQ.SMPLTS,
// SPACE=(6144,(400,80,10)),
// DISP=(NEW,CATLG,DELETE),
// UNIT=&SMPUNIT,
// VOL=SER=&SMPVOL,
// DCB=(RECFM=U,BLKSIZE=32760)
//*
//SMPSCDS DD DSN=&SMPHLQ..&SMPSLQ.SMPSCDS,
// SPACE=(8800,(200,40,10)),
// DISP=(NEW,CATLG,DELETE),
// UNIT=&SMPUNIT,
// VOL=SER=&SMPVOL,
// DCB=(LRECL=80,RECFM=FB,BLKSIZE=0)
//*
//SMPSTS DD DSN=&SMPHLQ..&SMPSLQ.SMPSTS,
// SPACE=(8800,(200,40,10)),
// DISP=(NEW,CATLG,DELETE),
// UNIT=&SMPUNIT,
// VOL=SER=&SMPVOL,
// DCB=(LRECL=80,RECFM=FB,BLKSIZE=0)
//*
//SMPLOG DD DSN=&SMPHLQ..&SMPSLQ.SMPLOG,
// SPACE=(8800,(100,20)),
// DISP=(NEW,CATLG,DELETE),
// UNIT=&SMPUNIT,
// VOL=SER=&SMPVOL,
// DCB=(LRECL=510,RECFM=VB,BLKSIZE=0)
//*

```

Figure 10-4 JCL - Part 1 of 2



```

// SMPLOGA DD DSN=&SMPHLQ. .&SMPSLQ.SMPLOGA,
//          SPACE=(8800,(100,20)),
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=&SMPUNIT,
//          VOL=SER=&SMPVOL,
//          DCB=(LRECL=510,RECFM=VB,BLKSIZE=0)
// *
//          PEND
// ALLOCATE EXEC SMPALLOC,
//          SMPHLQ=IQI.V5R9M0,    <==1 SMP/E HIGH LEVEL QUALIFIER
//          SMPSLQ=,              <==2 OPTIONAL SECOND LEVEL QUAL.
//          SMPUNIT=SYSALLDA,     <==3 UNIT NAME OF ALLOCATION DISK
//          SMPVOL=Z18SMP        <==4 VOLUME NAME OF ALLOC DISK
// *
//

```

Figure 10-5 JCL - Part 2 of 2

### 10.3.5 Allocating libraries for target and distribution zone

We used the JCL to allocate files for the target and distribution libraries for the IBM ISPF Productivity Tool, as shown in Figure 10-6 on page 204, Figure 10-7 on page 205, and Figure 10-8 on page 206.

```

//INTT025A JOB (1), 'GLOBAL ZONE LIBS', CLASS=B,
// NOTIFY=&SYSUID,
// MSGCLASS=O, MSGLEVEL= (1,1)
//IQIRALOC PROC TGTHLQ=, DSTHLQ=, TUNIT=, DUNIT=, TVOLID1=, TVOLID2=, DVOLID=
//ALLOC EXEC PGM=IEFBR14
//*
//* ALLOCATE TARGET LIBRARIES
//*
//SIQIINST DD DSN=&TGTHLQ.SIQIINST,
//          UNIT=&TUNIT,
//          VOL=SER=&TVOLID2,
//          SPACE=(TRK, (5, 1, 3)),
//          DCB=(RECFM=FB, LRECL=80, BLKSIZE=0),
//          DISP=(NEW, CATLG)
//*
//SIQICLIB DD DSN=&TGTHLQ.SIQICLIB,
//          UNIT=&TUNIT,
//          VOL=SER=&TVOLID2,
//          SPACE=(TRK, (16, 1, 10)),
//          DCB=(RECFM=FB, LRECL=80, BLKSIZE=0),
//          DISP=(NEW, CATLG)
//*
//SIQICLBV DD DSN=&TGTHLQ.SIQICLBV,
//          UNIT=&TUNIT,
//          VOL=SER=&TVOLID2,
//          SPACE=(TRK, (16, 1, 10)),
//          DCB=(RECFM=VB, LRECL=251, BLKSIZE=0),
//          DISP=(NEW, CATLG)
//*
//SIQILOAD DD DSN=&TGTHLQ.SIQILOAD,
//          UNIT=&TUNIT,
//          VOL=SER=&TVOLID1,
//          SPACE=(TRK, (35, 2, 10)),
//          DCB=(RECFM=U, LRECL=0, BLKSIZE=32760),
//          DISP=(NEW, CATLG)
//*
//SIQILPA DD DSN=&TGTHLQ.SIQILPA,
//          UNIT=&TUNIT,
//          VOL=SER=&TVOLID1,
//          SPACE=(TRK, (125, 5, 40)),
//          DCB=(RECFM=U, LRECL=0, BLKSIZE=32760),
//          DISP=(NEW, CATLG)
//*
//SIQIMLIB DD DSN=&TGTHLQ.SIQIMLIB,
//          UNIT=&TUNIT,
//          VOL=SER=&TVOLID2,
//          SPACE=(TRK, (3, 1, 5)),
//          DCB=(RECFM=FB, LRECL=80, BLKSIZE=0),
//          DISP=(NEW, CATLG)
//*
```

Figure 10-6 JCL - Part 1 of 3

```

//SIQIPLIB DD DSN=&TGTHLQ.SIQIPLIB,
//          UNIT=&TUNIT,
//          VOL=SER=&TVOLID2,
//          SPACE=(TRK,(110,10,230)),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=0),
//          DISP=(NEW,CATLG)
//*
//SIQISLIB DD DSN=&TGTHLQ.SIQISLIB,
//          UNIT=&TUNIT,
//          VOL=SER=&TVOLID2,
//          SPACE=(TRK,(1,1,1)),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=0),
//          DISP=(NEW,CATLG)
//*
//SIQITLIB DD DSN=&TGTHLQ.SIQITLIB,
//          UNIT=&TUNIT,
//          VOL=SER=&TVOLID2,
//          SPACE=(TRK,(6,2,25)),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=0),
//          DISP=(NEW,CATLG)
//*
//*   ALLOCATE DISTRIBUTION LIBRARIES
//*
//AIQIINST DD DSN=&DSTHLQ.AIQIINST,
//          UNIT=&DUNIT,
//          VOL=SER=&DVOLID,
//          SPACE=(TRK,(5,1,3)),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=0),
//          DISP=(NEW,CATLG)
//*
//AIQICLIB DD DSN=&DSTHLQ.AIQICLIB,
//          UNIT=&TUNIT,
//          VOL=SER=&TVOLID2,
//          SPACE=(TRK,(16,1,10)),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=0),
//          DISP=(NEW,CATLG)
//*
//AIQICLBV DD DSN=&DSTHLQ.AIQICLBV,
//          UNIT=&TUNIT,
//          VOL=SER=&TVOLID2,
//          SPACE=(TRK,(16,1,10)),
//          DCB=(RECFM=VB,LRECL=251,BLKSIZE=0),
//          DISP=(NEW,CATLG)
//*
//AIQILOAD DD DSN=&DSTHLQ.AIQILOAD,
//          UNIT=&TUNIT,
//          VOL=SER=&TVOLID1,
//          SPACE=(TRK,(35,2,10)),
//          DCB=(RECFM=U,LRECL=0,BLKSIZE=32760),
//          DISP=(NEW,CATLG)
//*
```

Figure 10-7 JCL - Part 2 of 3

```

//AIQILPA DD DSN=&DSTHLQ.AIQILPA,
//          UNIT=&TUNIT,
//          VOL=SER=&TVOLID1,
//          SPACE=(TRK,(125,5,40)),
//          DCB=(RECFM=U,LRECL=0,BLKSIZE=32760),
//          DISP=(NEW,CATLG)
//*
//AIQIMLIB DD DSN=&DSTHLQ.AIQIMLIB,
//          UNIT=&TUNIT,
//          VOL=SER=&TVOLID2,
//          SPACE=(TRK,(3,1,5)),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=0),
//          DISP=(NEW,CATLG)
//*
//AIQIPLIB DD DSN=&DSTHLQ.AIQIPLIB,
//          UNIT=&TUNIT,
//          VOL=SER=&TVOLID2,
//          SPACE=(TRK,(110,10,230)),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=0),
//          DISP=(NEW,CATLG)
//*
//AIQISLIB DD DSN=&DSTHLQ.AIQISLIB,
//          UNIT=&TUNIT,
//          VOL=SER=&TVOLID2,
//          SPACE=(TRK,(1,1,1)),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=0),
//          DISP=(NEW,CATLG)
//*
//AIQITLIB DD DSN=&DSTHLQ.AIQITLIB,
//          UNIT=&TUNIT,
//          VOL=SER=&TVOLID2,
//          SPACE=(TRK,(6,2,25)),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=0),
//          DISP=(NEW,CATLG)
//*
//          PEND
//ALLOCATE EXEC IQIRALOC,
//          TGTHLQ='IQI.V5R9M0.', <== TGT LIB HIGH LEVEL QUAL.
//          TUNIT=SYSALLDA, <== TGT LIB UNIT TYPE
//          TVOLID1=Z18R52, <== TGT LIB VOLUME 1
//          TVOLID2=Z18R52, <== TGT LIB VOLUME 2
//          DSTHLQ='IQI.V5R9M0.', <== DIST LIB HIGH LEVEL QUAL.
//          DUNIT=SYSALLDA, <== DIST LIB UNIT TYPE
//          DVOLID=Z18D52 <== DIST LIB VOLSER
//
//

```

Figure 10-8 JCL - Part 3 of 3

### 10.3.6 Defining global, target, and distribution zone options and creating DDDEF entries

Edit and submit sample job IQIJDDDF to create DDDEF entries for the SMP/E target and distribution libraries for the ISPF Productivity Tool. Consult the instructions in the sample job for more information.

We used the JCL to define the global, target, and distribution zone options to SMP/E as well as define the DDDEFs for SMP/E libraries, as shown in Figure 10-9 through Figure 10-15 on page 213.

```
//INTT025D JOB (1),'DEFINE SMP ZONES',
// NOTIFY=&SYSUID,
// MSGCLASS=0,MSGLEVEL=(1,1),CLASS=B
//CZONES1 EXEC PGM=GIMSMP,PARM='DATE=U',REGION=0M
//SMPCSI DD DSN=IQI.V5R9M0.GLOBAL.CSI,DISP=SHR <==1
//SMPLOG DD DSN=IQI.V5R9M0.SMPLOG,DISP=SHR <==2
//SMPPTS DD DSN=IQI.V5R9M0.SMPPTS,DISP=SHR <==2
//SMPOUT DD SYSOUT=*
//SMPLIST DD SYSOUT=*
//SMRPT DD SYSOUT=*
//SMPSNAP DD DUMMY
//SYSDUMP DD DUMMY
//SMPCNTL DD *
SET BOUNDARY(GLOBAL)
.
UCLIN
.
ADD GLOBALZONE
SREL(Z038)
OPTIONS(DEFOPT) /* <==7 OPTIONS ENTRY NAME */
ZONEINDEX((TZONE,IQI.V5R9M0.TZONE.CSI,TARGET), /* <==3,4 */
(DZONE,IQI.V5R9M0.DZONE.CSI,DLIB)) /* <==3,5 */
.
ADD OPTIONS(DEFOPT) /* <==7 OPTIONS ENTRY NAME */
DSSPACE(300,500,900)
DSPREFIX(IQI.V5R9M0) /* <==6 PREFIX FOR SMPTLIBS */
NOPURGE
NOREJECT
SAVEMTS
SAVESTS
/* NUCID(1) */
PAGELEN(60)
PEMAX(9999)
RETRYDDN(ALL)
.
ENDUCL
SET BOUNDARY(TZONE) /* <==3 */
.
UCLIN
.
ADD TARGETZONE(TZONE) /* <==3 */
OPTIONS(DEFOPT) /* <==7 OPTIONS ENTRY NAME */
SREL(Z038)
RELATED(DZONE) /* <==3 */
.
ENDUCL
SET BOUNDARY(DZONE) /* <==3 */
.
```

Figure 10-9 JCL - Part 1 of 7

```

UCLIN
.
  ADD DLIBZONE (DZONE)                                /* <==3 */
    OPTIONS (DEFOPT)                                /* <==7 OPTIONS ENTRY NAME */
    SREL (Z038)
    RELATED (TZONE)                                /* <==3 */
.
ENDUCL
.
SET BOUNDARY (GLOBAL)
.
LIST
  ALLZONES
.
//*****
//* Create DDDEFs for SMP/E libraries
//*****
//DDDEF EXEC PGM=GIMSMP,COND=(0,LT),REGION=0M
//SMPCSI DD DSN=IQI.V5R9M0.GLOBAL.CSI,DISP=SHR <==1
//SMPLOG DD DSN=IQI.V5R9M0.SMPLOG,DISP=SHR <==2
//SMPPTS DD DSN=IQI.V5R9M0.SMPPTS,DISP=SHR <==2
//SMPOUT DD SYSOUT=*
//SMPLIST DD SYSOUT=*
//SMRPT DD SYSOUT=*
//SMPCNTL DD *
SET BOUNDARY (GLOBAL)
.
  UCLIN.
    ADD DDDEF (SMPDEBUG)
      SYSOUT (*) .
    ADD DDDEF (SMPLIST)
      SYSOUT (*) .
    ADD DDDEF (SMPLOG)
      DATASET (IQI.V5R9M0.SMPLOG) /* <==2 */
      UNIT (SYSALLDA)
      VOLUME (Z18SMP) /* <==8 */
      WAITFORDSN
      SHR.
    ADD DDDEF (SMPLOGA)
      DATASET (IQI.V5R9M0.SMPLOGA) /* <==2 */
      UNIT (SYSALLDA)
      VOLUME (Z18SMP) /* <==8 */
      WAITFORDSN
      SHR.
    ADD DDDEF (SMPOUT)
      SYSOUT (*) .
    ADD DDDEF (SMPPTS)
      DATASET (IQI.V5R9M0.SMPPTS) /* <==2 */
      UNIT (SYSALLDA)
      VOLUME (Z18SMP) /* <==8 */
      WAITFORDSN
      SHR.
    ADD DDDEF (SMPPUNCH)
      SYSOUT (B) .
      SYSOUT (*) .

```

Figure 10-10 JCL - Part 2 of 7

```

ADD DDDEF (SMPSNAP)
  SYSOUT(*) .
ADD DDDEF (SMPTLIB)
  UNIT(SYSALLDA)
  VOLUME(Z18SMP) .                               /* <==8 */
ADD DDDEF (SYSPRINT)
  SYSOUT(*) .
ADD DDDEF (SYSUT1)
  UNIT(SYSALLDA)
  SPACE(2 ,1 )
  CYL.
ADD DDDEF (SYSUT2)
  UNIT(SYSALLDA)
  SPACE(2 ,1 )
  CYL.
ADD DDDEF (SYSUT3)
  UNIT(SYSALLDA)
  SPACE(2 ,1 )
  CYL.
ADD DDDEF (SYSUT4)
  UNIT(SYSALLDA)
  SPACE(2 ,1 )
  CYL.
ENDUCL
.
SET BOUNDARY (TZONE)                               /* <==3 */
.
UCLIN.
  ADD DDDEF (MACLIB)
    DATASET (SYS1.MACLIB)                           /* <==9 */
    UNIT(SYSALLDA)
    VOLUME(Z18R52)                                   /* <==10 */
    WAITFORDSN
    SHR.
  ADD DDDEF (SMPDEBUG)
    SYSOUT(*) .
  ADD DDDEF (SMPLIST)
    SYSOUT(*) .
  ADD DDDEF (SMPLOG)
    DATASET (IQI.V5R9M0.SMPLOG)                       /* <==2 */
    UNIT(SYSALLDA)
    VOLUME(Z18SMP)                                   /* <==8 */
    WAITFORDSN
    SHR.
  ADD DDDEF (SMPLOGA)
    DATASET (IQI.V5R9M0.SMPLOGA)                       /* <==2 */
    UNIT(SYSALLDA)
    VOLUME(Z18SMP)                                   /* <==8 */
    WAITFORDSN
    SHR.

```

Figure 10-11 JCL - Part 3 of 7

```

ADD DDDEF (SMPLTS)
    DATASET (IQI.V5R9M0.SMPLTS)                /* <==2 */
    UNIT (SYSALLDA)
    VOLUME (Z18SMP)                            /* <==8 */
    WAITFORDSN
    SHR.
ADD DDDEF (SMPMTS)
    DATASET (IQI.V5R9M0.SMPMTS)
    UNIT (SYSALLDA)
    VOLUME (Z18SMP)                            /* <==8 */
    WAITFORDSN
    SHR.
ADD DDDEF (SMPPTS)
    DATASET (IQI.V5R9M0.SMPPTS)
    UNIT (SYSALLDA)
    VOLUME (Z18SMP)                            /* <==8 */
    WAITFORDSN
    SHR.
ADD DDDEF (SMPPUNCH)
    SYSOUT (B) .
ADD DDDEF (SMPRPT)
    SYSOUT (*) .
ADD DDDEF (SMPSCDS)
    DATASET (IQI.V5R9M0.SMPSCDS)
    UNIT (SYSALLDA)
    VOLUME (Z18SMP)                            /* <==8 */
    WAITFORDSN
    SHR.
ADD DDDEF (SMPSNAP)
    SYSOUT (*) .
ADD DDDEF (SMPSTS)
    DATASET (IQI.V5R9M0.SMPSTS)
    UNIT (SYSALLDA)
    VOLUME (Z18SMP)                            /* <==8 */
    WAITFORDSN
    SHR.
ADD DDDEF (SMPWRK1)
    UNIT (SYSALLDA)
    SPACE (10 , 5 )
    DIR (250)
    CYL.
ADD DDDEF (SMPWRK2)
    UNIT (SYSALLDA)
    SPACE (10 , 5 )
    DIR (250)
    CYL.
ADD DDDEF (SMPWRK3)
    UNIT (SYSALLDA)
    SPACE (10 , 5 )
    DIR (250)
    CYL.

```

Figure 10-12 JCL - Part 4 of 7



```

ADD DDDEF (SMPWRK4)
    UNIT (SYSALLDA)
    SPACE (10 , 5 )
    DIR (250)
    CYL.
ADD DDDEF (SMPWRK6)
    UNIT (SYSALLDA)
    SPACE (10 , 5 )
    DIR (250)
    CYL.
ADD DDDEF (SYSLIB)
    CONCAT (SMPMTS
            MACLIB) .
ADD DDDEF (SYSPRINT)
    SYSOUT (*).
ADD DDDEF (SYSUT1)
    UNIT (SYSALLDA)
    SPACE (5 , 1 )
    CYL.
ADD DDDEF (SYSUT2)
    UNIT (SYSALLDA)
    SPACE (5 , 1 )
    CYL.
ADD DDDEF (SYSUT3)
    UNIT (SYSALLDA)
    SPACE (5 , 1 )
    CYL.
ADD DDDEF (SYSUT4)
    UNIT (SYSALLDA)
    SPACE (5 , 1 )
    CYL.
ENDUCL
.
SET BOUNDARY (DZONE) /* <==3 */
.
UCLIN.
ADD DDDEF (MACLIB)
    DATASET (SYS1.MACLIB) /* <==9 */
    UNIT (SYSALLDA)
    VOLUME (Z18R52) /* <==10 */
    WAITFORDSN
    SHR.
ADD DDDEF (SMPDEBUG)
    SYSOUT (*).
ADD DDDEF (SMPLIST)
    SYSOUT (*).
ADD DDDEF (SMPLOG)
    DATASET (IQI.V5R9M0.SMPLOG) /* <==2 */
    UNIT (SYSALLDA)
    VOLUME (Z18SMP) /* <==8 */
    WAITFORDSN
    SHR.

```

Figure 10-13 JCL - Part 5 of 7

```

ADD DDDEF (SMPLOGA)
    DATASET (IQI.V5R9M0.SMPLOGA)          /* <==2 */
    UNIT (SYSALLDA)
    VOLUME (Z18SMP)                       /* <==8 */
    WAITFORDSN
    SHR.
ADD DDDEF (SMPLTS)
    DATASET (IQI.V5R9M0.SMPLTS)           /* <==2 */
    UNIT (SYSALLDA)
    VOLUME (Z18SMP)                       /* <==8 */
    WAITFORDSN
    SHR.
ADD DDDEF (SMPMTS)
    DATASET (IQI.V5R9M0.SMPMTS)           /* <==2 */
    UNIT (SYSALLDA)
    VOLUME (Z18SMP)                       /* <==8 */
    WAITFORDSN
    SHR.
ADD DDDEF (SMPPTS)
    DATASET (IQI.V5R9M0.SMPPTS)           /* <==2 */
    UNIT (SYSALLDA)
    VOLUME (Z18SMP)                       /* <==8 */
    WAITFORDSN
    SHR.
ADD DDDEF (SMP PUNCH)
    SYSOUT (B) .
ADD DDDEF (SMPRPT)
    SYSOUT (*) .
ADD DDDEF (SMPSCDS)
    DATASET (IQI.V5R9M0.SMPSCDS)          /* <==2 */
    UNIT (SYSALLDA)
    VOLUME (Z18SMP)                       /* <==8 */
    WAITFORDSN
    SHR.
ADD DDDEF (SMPSNAP)
    SYSOUT (*) .
ADD DDDEF (SMPSTS)
    DATASET (IQI.V5R9M0.SMPSTS)           /* <==2 */
    UNIT (SYSALLDA)
    VOLUME (Z18SMP)                       /* <==8 */
    WAITFORDSN
    SHR.
ADD DDDEF (SMPWRK1)
    UNIT (SYSALLDA)
    SPACE (10 ,5 )
    DIR (250)
    CYL .
ADD DDDEF (SMPWRK2)
    UNIT (SYSALLDA)
    SPACE (10 ,5 )
    DIR (250)
    CYL .

```

Figure 10-14 JCL - Part 6 of 7

```

ADD DDDEF (SMPWRK3)
    UNIT (SYSALLDA)
    SPACE(10 ,5 )
    DIR(250)
    CYL.
ADD DDDEF (SMPWRK4)
    UNIT (SYSALLDA)
    SPACE(10 ,5 )
    DIR(250)
    CYL.
ADD DDDEF (SMPWRK6)
    UNIT (SYSALLDA)
    SPACE(10 ,5 )
    DIR(250)
    CYL.
ADD DDDEF (SYSLIB)
    CONCAT (SMPMTS
            MACLIB) .
ADD DDDEF (SYSPRINT)
    SYSOUT (*) .
ADD DDDEF (SYSUT1)
    UNIT (SYSALLDA)
    SPACE(5 ,1 )
    CYL.
ADD DDDEF (SYSUT2)
    UNIT (SYSALLDA)
    SPACE(5 ,1 )
    CYL.
ADD DDDEF (SYSUT3)
    UNIT (SYSALLDA)
    SPACE(5 ,1 )
    CYL.
ADD DDDEF (SYSUT4)
    UNIT (SYSALLDA)
    SPACE(5 ,1 )
    CYL.
ENDUCL.

```

//

Figure 10-15 JCL - Part 7 of 7

Expected return codes and messages: The job is considered successful if a return code of 0 is received.

### Sample job to define the SMP/E DDDEF zone entries

Next is the sample job we used to define the SMP/E DDEF zone entries, as shown in Figure 10-16 on page 214, Figure 10-17 on page 215, and Figure 10-18 on page 216.

```

//INTT025D JOB (1),'GLOBAL ZONE LIBS',CLASS=B,
// NOTIFY=&SYSUID,
// MSGCLASS=O,MSGLEVEL=(1,1)
    UNIT(SYSALLDA)
    WAITFORDSN
    SHR .
ADD DDDEF(SIQILOAD)
    DA(IQI.V5R9M0.SIQILOAD) /* <=== 3 TARGET LIBRARY */
    UNIT(SYSALLDA)
    WAITFORDSN
    SHR .
ADD DDDEF(SIQILPA)
    DA(IQI.V5R9M0.SIQILPA) /* <=== 3 TARGET LIBRARY */
    UNIT(SYSALLDA)
    WAITFORDSN
    SHR .
ADD DDDEF(SIQIMLIB)
    DA(IQI.V5R9M0.SIQIMLIB) /* <=== 3 TARGET LIBRARY */
    UNIT(SYSALLDA)
    WAITFORDSN
    SHR .
ADD DDDEF(SIQIPLIB)
    DA(IQI.V5R9M0.SIQIPLIB) /* <=== 3 TARGET LIBRARY */
    UNIT(SYSALLDA)
    WAITFORDSN
    SHR .
ADD DDDEF(SIQISLIB)
    DA(IQI.V5R9M0.SIQISLIB) /* <=== 3 TARGET LIBRARY */
    UNIT(SYSALLDA)
    WAITFORDSN
    SHR .
ADD DDDEF(SIQITLIB)
    DA(IQI.V5R9M0.SIQITLIB) /* <=== 3 TARGET LIBRARY */
    UNIT(SYSALLDA)
    WAITFORDSN
    SHR .
ADD DDDEF(AIQIINST)
    DA(IQI.V5R9M0.AIQIINST) /* <=== 3 DIST. LIBRARY */
    UNIT(SYSALLDA)
    WAITFORDSN
    SHR .
ADD DDDEF(AIQICLIB)
    DA(IQI.V5R9M0.AIQICLIB) /* <=== 3 DIST. LIBRARY */
    UNIT(SYSALLDA)
    WAITFORDSN
    SHR .
ADD DDDEF(AIQICLBV)
    DA(IQI.V5R9M0.AIQICLBV) /* <=== 3 DIST. LIBRARY */
    UNIT(SYSALLDA)
    WAITFORDSN
    SHR .

```

Figure 10-16 Sample job - Part 1 of 3

```

ADD DDDEF(AIQILOAD)
  DA(IQI.V5R9M0.AIQILOAD) /* <=== 3 DIST. LIBRARY */
  UNIT(SYSALLDA)
  WAITFORDSN
  SHR .
ADD DDDEF(AIQILPA)
  DA(IQI.V5R9M0.AIQILPA) /* <=== 3 DIST. LIBRARY */
  UNIT(SYSALLDA)
  WAITFORDSN
  SHR .
ADD DDDEF(AIQIMLIB)
  DA(IQI.V5R9M0.AIQIMLIB) /* <=== 3 DIST. LIBRARY */
  UNIT(SYSALLDA)
  WAITFORDSN
  SHR .
ADD DDDEF(AIQIPLIB)
  DA(IQI.V5R9M0.AIQIPLIB) /* <=== 3 DIST. LIBRARY */
  UNIT(SYSALLDA)
  WAITFORDSN
  SHR .
ADD DDDEF(AIQISLIB)
  DA(IQI.V5R9M0.AIQISLIB) /* <=== 3 DIST. LIBRARY */
  UNIT(SYSALLDA)
  WAITFORDSN
  SHR .
ADD DDDEF(AIQITLIB)
  DA(IQI.V5R9M0.AIQITLIB) /* <=== 3 DIST. LIBRARY */
  UNIT(SYSALLDA)
  WAITFORDSN
  SHR .
ENDUCL .
/*
//DDDEF2 EXEC PGM=GIMSMP,REGION=0M
//SMPCSI DD DSN=IQI.V5R9M0.GLOBAL.CSI,DISP=SHR /* <=== NOTE 2 */
//SMPCNTL DD *
SET BDY(DZONE) /* <=== NOTE 3 */
UCLIN .
  ADD DDDEF(AIQIINST)
  DA(IQI.V5R9M0.AIQIINST) /* <=== 3 DIST. LIBRARY */
  UNIT(SYSALLDA)
  WAITFORDSN
  SHR .
  ADD DDDEF(AIQICLIB)
  DA(IQI.V5R9M0.AIQICLIB) /* <=== 3 DIST. LIBRARY */
  UNIT(SYSALLDA)
  WAITFORDSN
  SHR .
  ADD DDDEF(AIQICLBV)
  DA(IQI.V5R9M0.AIQICLBV) /* <=== 3 DIST. LIBRARY */
  UNIT(SYSALLDA)
  WAITFORDSN
  SHR .

```

Figure 10-17 Sample job- Part 2 of 3

```

ADD DDDEF (AIQILOAD)
  DA (IQI.V5R9M0.AIQILOAD)      /* <=== 3 DIST. LIBRARY */
  UNIT (SYSALLDA)
  WAITFORDSN
  SHR .
ADD DDDEF (AIQILPA)
  DA (IQI.V5R9M0.AIQILPA)      /* <=== 3 DIST. LIBRARY */
  UNIT (SYSALLDA)
  WAITFORDSN
  SHR .
ADD DDDEF (AIQIMLIB)
  DA (IQI.V5R9M0.AIQIMLIB)     /* <=== 3 DIST. LIBRARY */
  UNIT (SYSALLDA)
  WAITFORDSN
  SHR .
ADD DDDEF (AIQIPLIB)
  DA (IQI.V5R9M0.AIQIPLIB)     /* <=== 3 DIST. LIBRARY */
  UNIT (SYSALLDA)
  WAITFORDSN
  SHR .
ADD DDDEF (AIQISLIB)
  DA (IQI.V5R9M0.AIQISLIB)     /* <=== 3 DIST. LIBRARY */
  UNIT (SYSALLDA)
  WAITFORDSN
  SHR .
ADD DDDEF (AIQITLIB)
  DA (IQI.V5R9M0.AIQITLIB)     /* <=== 3 DIST. LIBRARY */
  UNIT (SYSALLDA)
  WAITFORDSN
  SHR .
ENDUCL .
//

```

Figure 10-18 Sample job - Part 3 of 3

### Sample job to delete and define the SMP/E DDDEF zone entries

We used the following JCL to replace DDDEFs for SMP/E libraries, as shown in Figure 10-19 on page 217.

```

//INTT025D JOB (1),'REPLACE DDDEFS',CLASS=B,
// NOTIFY=&SYSUID,
// MSGCLASS=O,MSGLEVEL=(1,1)
//DDDEF      EXEC PGM=GIMSMP,COND=(0,LT),REGION=0M
//SMPCSI     DD DSN=IQI.V5R9M0.GLOBAL.CSI,DISP=SHR      <==1
//SMPLOG     DD DSN=IQI.V5R9M0.SMPLOG,DISP=SHR          <==2
//SMPPTS     DD DSN=IQI.V5R9M0.SMPPTS,DISP=SHR          <==2
//SMPOUT     DD SYSOUT=*
//SMPLIST    DD SYSOUT=*
//SMPRPT     DD SYSOUT=*
//SMPCNTL    DD *
SET BOUNDARY(GLOBAL)
.
  UCLIN.
    DEL DDDEF (SMPTLIB).
    ADD DDDEF (SMPTLIB)
      UNIT(SYSALLDA)
      VOLUME(Z18SMP).
  ENDUCL
/* <==8 */
.
//

```

Figure 10-19 Sample job

### 10.3.7 Performing SMP/E RECEIVE

If you obtained the ISPF Productivity Tool as part of a CBPDO, use the RCVPDO job found in the CBPDO RIMLIB data set to RECEIVE the ISPF Productivity Tool FMIDs and any service, HOLDDATA, or preventive service planning information included on the CBPDO tape. For more information, refer to the documentation included with the CBPDO.

You can also choose to edit and submit sample job IQIJRECV to perform the SMP/E RECEIVE for the ISPF Productivity Tool. Consult the instructions in the sample job for more information.

We used the JCL to RECEIVE ISPF Productivity Tool as, shown in Figure 10-20 on page 218.

```

//INTT025R JOB (1),'IQI RECEIVE',CLASS=B,
// NOTIFY=&SYSUID,
// MSGCLASS=O,MSGLEVEL=(1,1)
//RECEIVE EXEC PGM=GIMSMP,REGION=0M /* <=== NOTE 2 */
// *
//SMPCSI DD DSN=IQI.V5R9M0.GLOBAL.CSI,DISP=SHR /* <=== NOTE 3 */
// *
// *SMPTLIB DD UNIT=SYSALLDA,DISP=OLD, /* <=== NOTE 4 */
// * VOL=SER=volser
// *
// *SMPPTFIN DD DSN=SMPMCS,UNIT=200, /* <=== NOTE 5 */
// * VOL=SER=IQI590,DISP=OLD
// *
//SMPLOG DD DSN=IQI.V5R9M0.SMPLOG,DISP=SHR /* <=== NOTE 6 */
//SMPLOGA DD DSN=IQI.V5R9M0.SMPLOGA,DISP=SHR /* <=== NOTE 6 */
//SMPPTS DD DSN=IQI.V5R9M0.SMPPTS,DISP=SHR /* <=== NOTE 6 */
//SYSUT1 DD UNIT=SYSALLDA,SPACE=(CYL,(3,3)),DISP=(,DELETE)
//SYSUT2 DD UNIT=SYSALLDA,SPACE=(CYL,(3,3)),DISP=(,DELETE)
//SYSUT3 DD UNIT=SYSALLDA,SPACE=(CYL,(3,3)),DISP=(,DELETE)
//SYSUT4 DD UNIT=SYSALLDA,SPACE=(TRK,(3,3)),DISP=(,DELETE)
// ***
//SMPPTFIN DD DISP=SHR,DSN=IQI.V5R9M0.IBM.HIQI590.SMPMCS
// DD DISP=SHR,DSN=IQI.V5R9M0.IBM.HIQI590.PRODDATA
//SMPHOLD DD DISP=SHR,DSN=IQI.V5R9M0.IBM.HIQI590.SMPHOLD
//SMPCNTL DD *
SET BOUNDARY (GLOBAL)
.
RECEIVE SELECT(HIQI590)
RFPREFIX(IQI.V5R9M0)
SYSMODS
HOLDDATA
LIST
.
//

```

Figure 10-20 JCL

Expected return codes and messages: The job is considered successful if a return code of 0 is received.

### 10.3.8 Performing SMP/E APPLY

Edit and submit sample job IQIJAPP to perform an SMP/E APPLY CHECK for ISPF Productivity Tool.

Consult the instructions in the sample job for more information.

To receive the full benefit of the SMP/E Causer SYSMOD Summary Report, do *not* bypass the following on the APPLY CHECK: PRE, ID, REQ, and IFREQ because the SMP/E root cause analysis identifies the cause only of ERRORS and not of WARNINGS (SYSMODs that are bypassed are treated as warnings, not errors, by SMP/E).

We used the JCL to APPLY the ISPF Productivity Tool, as shown in Figure 10-21 on page 219.



```
//INTT025@ JOB (1),'IQI APPLY',CLASS=B,
// NOTIFY=&SYSUID,
// MSGCLASS=H,MSGLEVEL=(1,1)
//APPLY      EXEC PGM=GIMSMP,REGION=0M          /* <=== Note 2 */
// *
//SMPCSI     DD DSN=IQI.V5R9M0.GLOBAL.CSI,DISP=SHR /* <=== NOTE 3 */
//SMPLOG     DD DSN=IQI.V5R9M0.SMPLOG,DISP=SHR /* <=== NOTE 3 */
//SMPLOGA    DD DSN=IQI.V5R9M0.SMPLOGA,DISP=SHR /* <=== NOTE 3 */
//SMPMTS     DD DSN=IQI.V5R9M0.SMPMTS,DISP=SHR
//SMPSTS     DD DSN=IQI.V5R9M0.SMPSTS,DISP=SHR /* <=== NOTE 3 */
//SMPPTS     DD DSN=IQI.V5R9M0.SMPPTS,DISP=SHR /* <=== NOTE 3 */
//SMPSCDS    DD DSN=IQI.V5R9M0.SMPSCDS,DISP=SHR /* <=== NOTE 3 */
// *
//SMPCNTL    DD *
//      SET BDY(TZONE)                          /* <=== NOTE 4 */
//      OPTIONS(DEFOPT) .
//
//      APPLY SELECT(HIQI590)
//      FORMID(HIQI590)
//      GROUPEXTEND
//      /* CHECK */                               /* <=== Note 5 */
//
//
```

Figure 10-21 JCL

Expected return codes and messages from APPLY CHECK: The job is considered successful if a return code of 0 is received.

Expected return codes and messages from APPLY: The job is considered successful if a return code of 0 is received.

### 10.3.9 Performing SMP/E ACCEPT

Edit and submit sample job IQIJACC to perform a SMP/E ACCEPT CHECK for the ISPF Productivity Tool.

Consult the instructions in the sample job for more information.

To receive the full benefit of the SMP/E Causer SYSMOD Summary Report, do *not* bypass the following on the ACCEPT CHECK: PRE, ID, REQ, and IFREQ, because the SMP/E root cause analysis identifies the cause only of ERRORS and not of WARNINGS (SYSMODs that are bypassed are treated as warnings, not errors, by SMP/E).

Set the ACCJCLIN indicator in the distribution zone *before* you use SMP/E to load new distribution libraries, which saves the entries that are produced from JCLIN in the distribution zone whenever a SYSMOD that contains inline JCLIN is ACCEPTed. For more information about the ACCJCLIN indicator, see the description of inline JCLIN in the SMP/E manuals.

After you take any of the actions that are indicated by the ACCEPT CHECK, remove the CHECK operand, and run the job again to perform the ACCEPT.

We used the JCL to ACCEPT the ISPF Productivity Tool, as shown in Figure 10-22 on page 220.

```
//INTT025A JOB (1),'IQI ACCEPT',CLASS=B,
// NOTIFY=&SYSUID,
// MSGCLASS=O,MSGLEVEL=(1,1)
//ACCEPT EXEC PGM=GIMSMP,REGION=0M /* <=== Note 2 */
// *
//SMPCSI DD DSN=IQI.V5R9M0.GLOBAL.CSI,DISP=SHR /* <=== Note 3 */
// *
//SMPCNTL DD *
// SET BDY(DZONE) /* <=== Note 4 */
// OPTIONS(DEFOPT) .
//
// ACCEPT SELECT(HIQI590)
// FORFMID(HIQI590)
// GROUPEXTEND
// /* CHECK */ /* <=== Note 5 */
//
```

Figure 10-22 JCL

### Expected return codes and messages from ACCEPT CHECK

If PTFs that contain replacement modules are ACCEPTed, SMP/E ACCEPT processing will link-edit/bind the modules into the distribution libraries. During this processing, the Linkage Editor or Binder might issue messages that document unresolved external references, which result in a return code of 4 from the ACCEPT step. We can ignore these messages. The distribution libraries are not executable and the unresolved external references do not affect the executable system libraries.

### Expected return codes and messages from ACCEPT

The job is considered successful if a return code of 0 is received.

## 10.4 Customizing the LOGON Proc to invoke ISPF-PT

The ISPF Productivity Tool delivers code work above any already existing ISPF environment. The only requirement for creating an ISPF Productivity Tool base environment is to modify an existing TSO LOGON procedure. However, it is better to create a new TSO LOGON procedure for ISPF-PT rather than modify the existing procedure.

We created the new TSO LOGON procedure as follows:

Copy the existing TSO LOGON procedure for ISPF. The new ISPF-PT LOGON procedure is ISPFPT.

Figure 10-23 on page 221 and Figure 10-24 on page 222 illustrates the changes for you to make to a copy of your LOGON procedure, or you can incorporate the changes into a new LOGON procedure that you create.

We created the ISPFPT TSO LOGON procedure as shown in Figure 10-23 on page 221 and Figure 10-24 on page 222.

```

// *-----
// * SERVERPAC LOGON PROCEDURE
// *
// * THIS PROCEDURE ENABLES USERS TO LOG ON TO TSO/E.
// * THE CLIST ISPPDF, WHICH RESIDES IN CPAC.CMDPROC,
// * IS EXECUTED AT FIRST TO INVOKE THE ISPF.
// *-----
// ISPFIT  PROC
// ISPFIT  EXEC PGM=IKJEFT01,DYNAMNBR=500,PARM=ISPIPT
// STEPLIB DD DISP=SHR,DSN=IQI.V5R9M0.SIQILPA
//          DD DISP=SHR,DSN=MQM.SCSQANLE
//          DD DISP=SHR,DSN=MQM.SCSQAUTH
//          DD DSN=QMF.SDSQEXIT,DISP=SHR          * QMF MODULES *
//          DD DSN=QMF.SDSQLOAD,DISP=SHR          * QMF MODULES *
//          DD DSN=SYS1.DSN810.SDSNEXIT,DISP=SHR   * DB2 MODULES *
//          DD DSN=DSN810.SDSNLOAD,DISP=SHR        * DB2 MODULES *
//          DD DSN=GDDM.SADMMOD,DISP=SHR          * GDDM MODULES *
// SYSPROC DD DISP=SHR,DSN=CPAC.CMDPROC
//          DD DISP=SHR,DSN=IQI.V5R9M0.SIQICLIB
// SYSHELP DD DISP=SHR,DSN=SYS1.HELP
//          DD DISP=SHR,DSN=ISF.SISFHELP
//          DD DISP=SHR,DSN=SYS1.SBDTHelp
//          DD DISP=SHR,DSN=SYS1.HELPENP
//          DD DISP=SHR,DSN=ISP.SISPHELP
// SYSLBC  DD DISP=SHR,DSN=SYS1.BROADCAST
// SYSPRINT DD TERM=TS,SYSOUT=*
// SYSTEM  DD TERM=TS,SYSOUT=*
// ISPLIB  DD DSN=QMF.SDSQPLBE,DISP=SHR
// ISPLIB  DD DSN=QMF.SDSQMLBE,DISP=SHR
// ISPLIB  DD DSN=QMF.SDSQSLBE,DISP=SHR
// *****
// * QMF/GDDM DATA SETS *
// *****
// ADMGMAP DD DSN=QMF.SDSQMAPE,DISP=SHR
// ADMCFORM DD DSN=QMF.SDSQCHRT,DISP=SHR
// *DSQUCFRM DD DSN=AAAAAAA,DISP=SHR
// ADMCDATA DD DSN=GDDM.SADMCD,DISP=SHR
// ADMGDF DD DSN=GDDM.SADMGDF,DISP=SHR
// ADMSYMBL DD DSN=GDDM.SADMSYM,DISP=SHR

```

Figure 10-23 LOGON procedure - Part 1 of 2

```

//*****
// * DATASETS USED BY QMF *
//*****
//DSQPNLE DD DSN=QMF.DSQPNLE,DISP=SHR
//DSQPRINT DD SYSOUT=A,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//DSQDEBUG DD SYSOUT=A,DCB=(RECFM=FBA,LRECL=121,BLKSIZE=1210)
//DSQEDIT DD UNIT=SYSDA,DCB=(RECFM=FBA,LRECL=79,BLKSIZE=4029),
// DISP=NEW,SPACE=(CYL,(1,1))
//DSQUDUMP DD SYSOUT=A,DCB=(RECFM=FBA,LRECL=125,BLKSIZE=1632)
//SYSUDUMP DD SYSOUT=A
//DSQSPILL DD DSN=QMF.SPILL,DISP=(NEW,DELETE),
// UNIT=SYSDA,SPACE=(CYL,(1,1),RLSE),
// DCB=(RECFM=F,LRECL=4096,BLKSIZE=4096)
//SYSIN DD TERM=TS
// *
//*****
// * IPT/IQI DATA SETS *
//*****
//IQITLIB DD DISP=SHR,DSN=IQI.V5R9M0.SIQITLIB

```

Figure 10-24 LOGON procedure - Part 2 of 2

## 10.4.1 The LPA load library SIQILPA

The ISPF-PT target library, SIQILPA, is specified at the first of DD (STEPLIB).

## 10.4.2 CLIST library SIQICLIB

Add the CLIST library to the DD (SYSPROC) concatenation. We recommend that you place the ISPF Productivity Tool CLIST library as first in the concatenation. We created the new CLIST PROC ISPIPT and placed it in CPAC.COMDPROC. PROC name. ISPIPT is the parameter to IKJEFT01.

## 10.4.3 Table library SIQITLIB

Add a new statement for the table library with DD (IQITLIB). Do not confuse this DD name with ISPF's ISPTLIB.

Remember that the ISPF-PT target libraries must always be *ahead of* ISPF's SISPLPA and SISPLD.

The new procedure ISPIPT invokes the CLIST ISPIPT. We created the CLIST ISPIPT as shown in Figure 10-25 on page 223 through Figure 10-30 on page 228.

```

PROC 0 PANEL()
/*****
/*
/* 5751-CS9
/* (C) COPYRIGHT IBM CORP. 1996, 2006
/*
/* LIB: CPAC.CMDPROC(ISPPDF)
/* DOC: THIS CLIST ALLOCATES THE FILES REQUIRED BY ISPF AND PDF
/*      AND THEN INVOKES PDF.
/*
/* NOTE: SYSPROC IS FREED AND REALLOCATED TO INCLUDE THE PDF CLIST
/*      DATASET. THIS MAY RESULT IN A DIFFERENT CONCATENATION
/*      THAN EXISTED BEFORE THIS CLIST WAS INVOKED.
/*
*****/
CONTROL NOFLUSH NOMSG MAIN
PROFILE MODE WTPMSG MSGID
FREE FILE(ISPLLIB,ISPLLIB,ISPLLIB,ISPTLIB,ISPSLIB, +
          ISPPROF,ISPTABL,SMPTABL,IPCSPARM,ISPPALT,ISPMALT,ISPILIB)
/*****
WRITE
WRITE LOGON PROC IS ISPIPT
WRITENR ALLOCATING ISPF AND PRODUCT DATASETS
/*****
FREE FI(SYSPROC)
ALLOC FI(SYSPROC) SHR DA( +
                        'CPAC.CMDPROC' +
                        'IQI.V5R9M0.SIQICLIB' +
                        'DSN810.SDSNCLST' +
                        'SYS1.SBLSCLI0' +
                        'SYS1.SERBCLS' +
                        'SYS1.HRFCLST' +
                        'GIM.SGIMCLS0' +
                        'CBC.SCCNUTL' +
                        'SYS1.SICECLIB' +
                        'SYS1.SBDTCLI0' +
                        'SYS1.SCBDCLST' +
                        'CSF.SCSFCLI0' +
                        'EOY.SEOYCLIB' +
                        'EOX.SEPHCLB1' +
                        'ICQ.ICQCCLIB' +
                        'ISP.SISPCLIB' +
                        'GDDM.SADMSAM' +
                        'FFST.V120ESA.SEPWCENU' +
                        'SYS1.DGTCLIB')

```

Figure 10-25 CLIST ISPIPT - Part 1 of 6

```

FREE FI(SYSEXEC)
ALLOC FI(SYSEXEC) SHR DA( +
    'MQM.SCSQEXEC' +
    'QMF.SDSQEXCE' +
    'EUV.SEUVEXEC' +
    'ISF.SISFEXEC' +
    'AOP.SAOPEXEC' +
    'IOE.SIOEEXEC' +
    'EOY.SEOYCLIB' +
    'EOX.SEPHCLB1' +
    'SYS1.SBPXEXEC' +
    'EUVF.SEUVFEXC' +
    'GLD.SGLDEXEC' +
    'ISP.SISPEXEC' +
    'FFST.V120ESA.SEPWSRC1' +
    'SYS1.SEDGEXE1')

WRITENR .
SET &DSNAME = &SYSUID..ISPF.ISPPROF
ALLOC FI(ISPPROF) SHR DA('&DSNAME.')
IF &LASTCC ^= 0 THEN +
    DO
        FREE FI(ISPCRTE)
        CONTROL MSG
        ATTRIB ISPCRTE DSORG(PO) RECFM(F B) LRECL(80) BLKSIZE(6160)
        ALLOC DA('&DSNAME.') SP(5,1) CYLINDERS DIR(20) USING(ISPCRTE) +
            FI(ISPPROF)
        IF &LASTCC = 0 THEN +
            WRITE *** ISPF PROFILE DATA SET '&DSNAME.' HAS BEEN CREATED
        ELSE +
            DO
                WRITE *** UNABLE TO ALLOCATE ISPF PROFILE DATA SET '&DSNAME.'
                FREE FI(ISPCRTE)
                EXIT CODE(12)
            END
        FREE FI(ISPCRTE)
    END
CONTROL MSG
ERROR EXIT
IF &PANEL = &STR() THEN +
    SET &PNL = PANEL(ISR@PRIM)
ELSE +
    SET &PNL = PANEL(&PANEL)
ALLOC FI(ISPTABL) SHR DA( +
    '&DSNAME' +
    )
ALLOC FI(SMPTABL) SHR DA('&DSNAME.')
ALLOC FI(ISPLLIB) SHR DA( +
    'IQI.V5R9M0.SIQILOAD' +
    'MQM.SCSQANLE' +
    'SYS1.DFQLLIB' +
    'SYS1.DGTLLIB')

```

Figure 10-26 CLIST ISPIPT - Part 2 of 6

```

WRITENR .
ALLOC FI(ISPPLIB) SHR DA( +
      'CPAC.ISPPLIB' +
      'IQI.V5R9M0.SIQIPLIB' +
      'DSN810.SDSNPFPE' +
      'ISP.SISPPENU' +
      'SYS1.SERBPENU' +
      'SYS1.HRFPANL' +
      'MQM.SCSQPNLE' +
      'GIM.SGIMPENU' +
      'EUV.SEUVPNL' +
      'TCPIP.SEZAPENU' +
      'SYS1.SBLSPNL0' +
      'ISF.SISFPLIB' +
      'SYS1.SHASPNL0' +
      'AOP.SAOPPENU' +
      'SYS1.SICEPENU' +
      'EOY.SEOYPENU' +
      'SYS1.SBDTPNL0' +
      'IOE.SIOEPNLE' +
      'SYS1.SCBDPENU' +
      'CSF.SCSFPNL0' +
      'EOX.SEPHPNL1' +
      'SYS1.SBPXPENU' +
      'ICQ.ICQPLIB' +
      'GDDM.SADMPNL' +
      'FFST.V120ESA.SEPWPENU' +
      'SYS1.DFQPLIB' +
      'SYS1.DGTPLIB' +
      'SYS1.SEDGPENU')

```

Figure 10-27 CLIST ISPIPT - Part 3 of 6

```

WRITENR .
ALLOC FI(ISPPLIB) SHR DA( +
      'CPAC.ISPPLIB' +
      'IQI.V5R9M0.SIQIPLIB' +
      'DSN810.SDSNPFPE' +
      'ISP.SISPPENU' +
      'SYS1.SERBPENU' +
      'SYS1.HRFPANL' +
      'MQM.SCSQPNLE' +
      'GIM.SGIMPENU' +
      'EUV.SEUVPNL' +
      'TCPIP.SEZAPENU' +
      'SYS1.SBLSPNL0' +
      'ISF.SISFPLIB' +
      'SYS1.SHASPNL0' +
      'AOP.SAOPPENU' +
      'SYS1.SICEPENU' +
      'EOY.SEOYPENU' +
      'SYS1.SBDTPNL0' +
      'IOE.SIOEPNLE' +
      'SYS1.SCBDPENU' +
      'CSF.SCSFPNL0' +
      'EOX.SEPHPNL1' +
      'SYS1.SBPXPENU' +
      'ICQ.ICQPLIB' +
      'GDDM.SADMPNL' +
      'FFST.V120ESA.SEPWPENU' +
      'SYS1.DFQPLIB' +
      'SYS1.DGTPLIB' +
      'SYS1.SEDGPENU')

```

Figure 10-28 CLIST ISPIPT - Part 4 of 6



```

WRITENR .
ALLOC FI (ISPMLIB) SHR DA( +
    'SYS1.SERBMENU' +
    'IQI.V5R9M0.SIQIMLIB' +
    'DSN810.SDSNSPFM' +
    'SYS1.HRFMSG' +
    'MQM.SCSQMSG' +
    'GIM.SGIMMENU' +
    'EUV.SEUVMSG' +
    'TCPIP.SEZAMENU' +
    'ISF.SISFMLIB' +
    'AOP.SAOPMENU' +
    'SYS1.SICEMENU' +
    'EOY.SEOYMENU' +
    'SYS1.SBDTMSG' +
    'IOE.SIOEMSG' +
    'SYS1.SCBDMENU' +
    'CSF.SCSFMSG0' +
    'EOY.SEOYBENU' +
    'EOX.SEPHMSG1' +
    'SYS1.SBLSMSG0' +
    'SYS1.SBPXMENU' +
    'ICQ.ICQMLIB' +
    'ISP.SISPMENU' +
    'GDDM.SADMMSG' +
    'SYS1.DGTMLIB' +
    'SYS1.DFQMLIB' +
    'SYS1.SEDGMENU')
ALLOC FI (IPCSPARM) SHR DA( +
    'SYS1.SHASPARM' +
    'SYS1.IBM.PARMLIB')
WRITENR .
ALLOC FI (ISPTLIB) SHR DA( +
    '&DSNAME' +
    'SYS1.SERBT' +
    'SYS1.SERBTENU' +
    'MQM.SCSQTBL' +
    'GIM.SGIMTENU' +
    'ISF.SISFTLIB' +
    'SYS1.SICETLIB' +
    'EOY.SEOYTENU' +
    'SYS1.SCBDTENU' +
    'CSF.SCSFTLIB' +
    'EOX.SEPHTBL1' +
    'SYS1.SBLSTBL0' +
    'SYS1.SBPXTENU' +
    'ICQ.ICQTABLS' +
    'ISP.SISPTENU' +
    'SYS1.DGTTLIB')

```

Figure 10-29 CLIST ISPIPT - Part 5 of 6

```

WRITENR .
ALLOC FI (ISPSLIB) SHR DA( +
                                'SYS1.HRFSKEL'
                                'IQI.V5R9M0.SIQISLIB'
                                'GIM.SGIMSENU'
                                'ISF.SISFSLIB'
                                'SYS1.SICESLIB'
                                'CSF.SCSFSKL0'
                                'SYS1.SBLSKEL0'
                                'ICQ.ICQSLIB'
                                'ISP.SISPSLIB'
                                'ISP.SISPSENU'
                                'SYS1.DGTSLIB')
WRITENR .
ALLOC FI (ISPILIB) SHR DA( +
                                'ISP.SISPSAMP')
ALLOC FI (DITPLIB) SHR DA('DIT.V1R3M0.SDITPLIB')
WRITE
BROADCAST
ERROR RETURN
PDF &PNL

```

Figure 10-30 CLIST ISPIPT - Part 6 of 6

#### 10.4.4 ISPF-PT load library SIQILOAD

Place the ISPF-PT load library, SIQILOAD, in DD(ISPLLIB) or DD(STEPLIB).

#### 10.4.5 CLIST library SIQICLIB

Add the CLIST library to the DD (SYSPROC) concatenation. We recommend that you place the ISPF Productivity Tool CLIST library as first in the concatenation.

#### 10.4.6 Message library SIQIMLIB

Add the message library to the DD(ISPMLIB) concatenation.

#### 10.4.7 ISPF-PT panel library SIQIPLIB

Add the panel library to the DD(ISPPLIB) concatenation. We recommend that you place the ISPF Productivity Tool panel library as first in the concatenation.

#### 10.4.8 Skeleton library IQISLIB

Add the skeleton library to the DD(ISPSLIB) concatenation.

## 10.5 Creating RACF profile to use the customized TSO LOGON PROC for ISPF-PT

To create the RACF® profile to use the customized TSO LOGON PROC for ISPF-PT:

1. Choose the **RACF - Resource Access Control Facility** menu from ISPF.
2. Choose option **2 - GENERAL RESOURCE PROFILES** from the RACF menu, as shown in Figure 10-31.

```

                                     RACF - SERVICES OPTION MENU

SELECT ONE OF THE FOLLOWING:

  1  DATA SET PROFILES
  2  GENERAL RESOURCE PROFILES
  3  GROUP PROFILES AND USER-TO-GROUP CONNECTIONS
  4  USER PROFILES AND YOUR OWN PASSWORD
  5  SYSTEM OPTIONS
  6  REMOTE SHARING FACILITY
  7  DIGITAL CERTIFICATES AND KEY RINGS
  99 EXIT

                                     Licensed Materials - Property of IBM
                                     5647-A01 (C) Copyright IBM Corp. 1983, 2000

OPTION === 2
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 10-31 RACF Services Option Menu

3. Choose option **1: ADD - Add a profile RACF SERVICES** menu, as shown in Figure 10-32.

```

                                     RACF - GENERAL RESOURCE PROFILE SERVICES

OPTION === 1
SELECT ONE OF THE FOLLOWING:

  1  ADD           Add a profile
  2  CHANGE        Change a profile
  3  DELETE        Delete a profile
  4  ACCESS        Maintain access list
  5  AUDIT         Monitor access attempts (Auditors only)

D or 8  DISPLAY    Display profile contents
S or 9  SEARCH     Search the RACF data base for profiles

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 10-32 RACF Services Options

In Figure 10-33, we specified CLASS as TSOPROC and PROFILE as ISPFIPT.

```

                                RACF - GENERAL RESOURCE SERVICES - ADD
OPTION ===>

ENTER THE FOLLOWING PROFILE INFORMATION:

CLASS    ===> TSOPROC
PROFILE  ===> ISPFIPT

                                <==end of data

USE A MODEL      ===>      YES or NO

NOTE: Embedded Blanks are NOT ALLOWED in class or profile names.
      The profile name may be case sensitive. View the help and
      select PROFILE NAME for more detail.

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 10-33 RACF - Add Profile

In Figure 10-34, we specified the relevant parameters. We recommend that you specify UACC as READ, as shown in Figure 10-34.

4. Press Enter. The profile ISPFIPT is ready to use.

```

                                RACF - ADD GENERAL RESOURCE PROFILE
COMMAND ===>

CLASS:    TSOPROC
PROFILE   _ ISPFIPT

ENTER OR CHANGE THE FOLLOWING INFORMATION:

OWNER      ===> AMIT      Userid or group name
LEVEL      ===> 0        0-99
FAILED ACCESSES ===> FAIL  FAIL or WARN
UACC       ===> READ     NONE, READ, UPDATE,
                        CONTROL, ALTER or EXECUTE
AUDIT SUCCESSES ===> NOAUDIT READ, UPDATE, CONTROL,
                        ALTER, or NOAUDIT
AUDIT FAILURES  ===> READ  READ, UPDATE, CONTROL,
                        ALTER, or NOAUDIT
NOTIFY        ===>      Userid

TO ADD OPTIONAL INFORMATION, ENTER YES ===>

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 10-34 Add Profile

## 10.6 Using the new TSO LOGON procedure to invoke ISPF-PT

Use the following steps to invoke ISPF-PT using the new TSO LOGON procedure:

1. Use the new ISPFIPT procedure to logon. When you logon for the first time using the new procedure ISPFIPT, it shows the ISPF Productivity Tool window, which provides you the

option to read more about ISPF-PT or continue to the ISPF session, as shown in Figure 10-35.

```

----- TSO/E LOGON -----

Enter LOGON parameters below:          RACF LOGON parameters:

Userid   ==> E09921
Password ==>
Procedure ==> ISPFIT
Acct Nubr ==> ACCT#
Size     ==>
Perform  ==>
Command  ==>

Enter an 'S' before each option desired below:
      -Nomail      -Nonnotice      S -Reconnect      -OIDcard

PF1/PF13 ==> Help   PF3/PF15 ==> Logoff   PA1 ==> Attention   PA2 ==> Reshow
You may request specific help information by entering a '?' in any entry field
  
```

Figure 10-35 TSO/E

2. Press Enter to read more about the ISPF-PT, as shown in Figure 10-36.

```

+----- IBM IPT VERSION 5.9 -----+
| ISPF Productivity Tool is installed in this system. You may:          |
| * Press the ENTER key to read more about -IPT-.                      |
| * Press the END key to continue with your ISPF session.              |
|                                                                           |
| Later, you will be able to display the -IPT- tutorial by entering     |
| the IPHELP command in the command field of any ISPF panel.            |
| Likewise, the IPT? main command, will remind you of all -IPT-        |
| command shortcuts available from any ISPF panel.                      |
+-----+

+-----+
| IBM ISPF Productivity Tool                                           |
| 5698-A81 (C) Copyright IBM Corp. 1989,2007. All rights reserved.    |
+-----+
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 10-36 IPT - Information

Figure 10-37 on page 232 gives us the option to choose the topic of our choice or to learn about ISPF-PT.

3. Press F3 to exit from the ISPF-PT general information.

```

----- ISPF Productivity Tool -----TUTORIAL
|          GENERAL INFORMATION          |
|-----|
The following topics are presented in sequence, or may be selected by number:

1  Introduction
2  General Commands
3  Enhanced EDIT, BROWSE, and VIEW
4  The Point-and-Shoot Interface
5  Data Set History Facility
6  Access to Data Sets by DD Name
7  Enhanced Data Set List Utility
8  Enhanced Printing Facilities

9  OLIST - Objects List
10 MSL - Member Selection List
11 Enhanced TSO Command Support
12 Enhanced EDIT Commands
13 Enhanced BROWSE Commands
14 The VIEW Facility-EDIT-Comp.Browse
15 Setting IBMIPT Options
16 IPTHELP-Displaying This Tutorial

SELECTION ===> _
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 10-37 ISPF-PT General Information

- When you exit the ISPF-PT General Information window, it shows the Persistent Table Management window, Figure 10-38, which indicates that IPITBLIB is unavailable. Press Enter to continue.

```

-IPT----- Persistent Table Management ----- IPITBLIB unavailable
COMMAND ===> _

IBMPT recommends a dedicated library (PDSE) for storing its
persistent tables. DD(IPITBLIB) is currently not allocated or may
be associated with an unusable dataset.
* You may either specify an existing library or dynamically allocate a
new library to be used for this purpose.
* After successfully allocating a valid library, its name is stored in
your profile and will be automatically used from now on, as long as
DD(IPITBLIB) is not explicitly included in your LOGON procedure.
* The library name first suggested by IBMPT is either your already
allocated permanent OLIST library, or if DD(PLSTLIBW) is missing,
a name and volume derived from your PROFILE library name.
Note: you may choose to operate without saving any IBMIPT persistent
tables by declining to allocate DD(IPITBLIB).

Main line commands: D Disable S Select (default)
Press ENTER to process or F3 to cancel.

LIBRARY NAME --> 'R00872.ISPF.IPITBLIB'

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 10-38 IPT Persistent Table Management assignment

- Allocate the data sets so that the procedure ISPFITP can use them. Specify the relevant fields, and press Enter to continue, as shown in Figure 10-39.

```

Menu RefList Utilities Help

Allocate New Data Set

Data Set Name . . . : E09921.ISPF.IPITBLIB      More: +

Management class . . . (Blank for default management class)
Storage class . . . (Blank for default storage class)
Volume serial . . . (Blank for system default volume) **
Device type . . . (Generic unit or device address) **
Data class . . . (Blank for default data class)
Space units . . . BLOCK (BLKS, TRKS, CYLS, KB, MB, BYTES
                        or RECORDS)
Average record unit . . (M, K, or U)
Primary quantity . . 320 (In above units)
Secondary quantity . . 160 (In above units)
Directory blocks . . 50 (Zero for sequential data set) *
Record format . . . FB
Record length . . . 80
Block size . . . 6160
Data set name type . . LIBRARY (LIBRARY, HFS, PDS, LARGE, BASIC, *
Command ==>
F1=Help F2=Split F3=Exit F7=Backward F8=Forward F9=Swap
F10=Actions F12=Cancel

```

Figure 10-39 Allocate Persistent Data Set

After allocation is successful, ISPF displays IPITBLIB is available for use, as shown in Figure 10-40.

```

Menu Utilities Compilers Options Status Help

ISPF Primary Option Menu  IPITBLIB now available

0 Settings      Terminal and user parameters      User ID . . : E09921
1 View          Display source data or listings  Time. . . : 20:43
2 Edit          Create or change source data    Terminal. . : 3278
3 Utilities     Perform utility functions       Screen. . . : 1
4 Foreground    Interactive language processing  Language. . : ENGLISH
5 Batch         Submit job for language processing Appl ID . . : ISP
6 Command       Enter TSO or Workstation commands TSO logon : ISPFITP
7 Dialog Test   Perform dialog testing          TSO prefix: E09921
9 IBM Products  IBM program development products        System ID : EARTH
10 SCLM         SW Configuration Library Manager MVS acct. : ACCT#
11 Workplace    ISPF Object/Action Workplace   Release . . : ISPF 5.8
D DB2          Perform DB2 Interactive Functions
Q QMF          Query Management Facility
S SDFII        Screen Definition Facility
O OmegaMon     Omegamon XE for DB2 PM v3.1
SD SDSF        System Display and Search Facility
P Master       Primary Master Menu
Option ==>
F1=Help F2=Split F3=Exit F7=Backward F8=Forward F9=Swap
F10=Actions F12=Cancel

```

Figure 10-40 IPITBLIB now available

We can continue to work after IPITBLIB is available.



On every instance after the first, when we logon using ISPFPT PROC, the ISPF main menu displays IBM IPT VERSION 5.9, as shown in Figure 10-41.

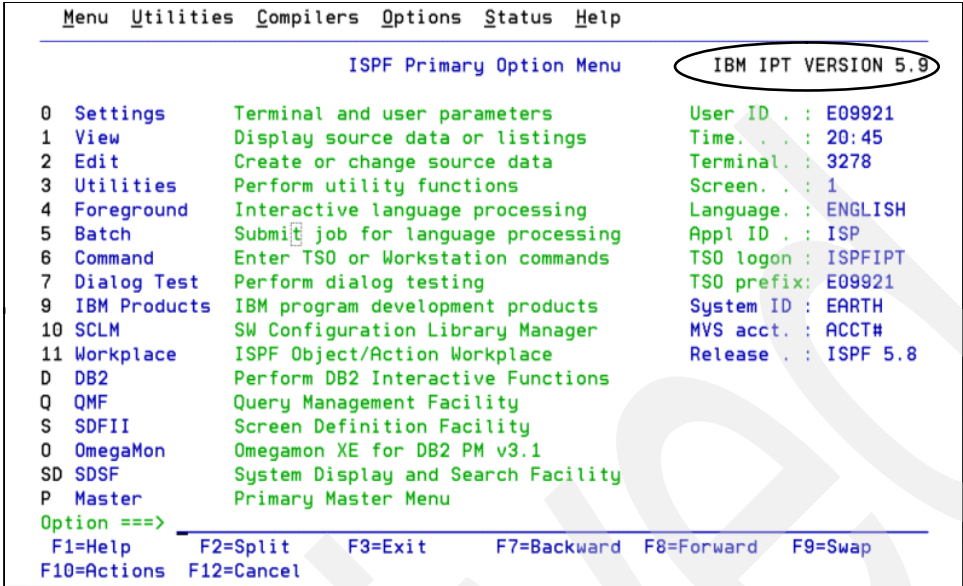


Figure 10-41 IBM ISPF-PT Version 5.9 displayed

## 10.7 ISPF Productivity Tool maintenance

SMP/E controls the maintenance of the ISPF Productivity Tool. You must periodically review and apply all of the available maintenance, as posted on the IBM ISPF Productivity Tool Web site at:

<http://www-306.ibm.com/software/awdtools/ispfproductivitytool/support>

To determine the maintenance level of the ISPF Productivity Tool at your installation, enter the following command on any ISPF Productivity Tool panel:

IPT VER or shortcut IVER

The initial window displays the range of installed APARs, as shown in Figure 10-42 on page 235. The scrollable list displays the date and maintenance level of each ISPF Productivity Tool module.



```

-IPT- ----- IBMIPT MODULE LIST - 10/19/07 (07.292) 20:49 Row 1 to 12 of 150
Commands: CONFIG, DOWN, END, FIND, REPORT, SORT, UP
USERID:E09921 LOGON PROC:ISPFPT ISPF VERSION:5.8
CPU SERIAL NUMBER:3B7BE MODEL:2094 GROUP:00 OS:MVS SP7.0.8 (HBB7730 )
IBMIPT version 5.9.0 with applied APARs OA15704 through OA22265.

COMMENTS          MODULE          LEVEL DATE      TIME      FIX-MARKER
-----
IQIDIRA  LVL:5009 08/23/07 - 12.04 FIX:OA22265
IQIORTN  LVL:5009 08/23/07 - 12.54 FIX:OA22265
IQILDMEM LVL:5009 08/23/07 - 11.54 FIX:OA22265
IQIPLST  LVL:5009 09/09/07 - 10.18 FIX:OA22265
IQISPF24 LVL:5009 08/22/07 - 23.01 FIX:OA22265
IQISPF31 LVL:5009 08/22/07 - 23.02 FIX:OA22265
IQIOBT   LVL:5009 08/21/07 - 13.09 FIX:OA22140
QIXDSL  LVL:5009 08/10/07 - 14.58 FIX:OA22140
QIXDSNC LVL:5009 08/10/07 - 21.36 FIX:OA22140
QICOPY   LVL:5009 07/06/07 - 18.35 FIX:OA21720
QIECMP   LVL:5009 07/06/07 - 18.45 FIX:OA21720
QIMAIN   LVL:5009 07/23/07 - 20.38 FIX:OA21720

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

```

Figure 10-42 IPT VER output

## 10.8 Ordering ISPF-PT V5.9 for India customers and installing ISPF-PT out-of-the box

In this section, we describe the following:

- ▶ Process to order ISPF-PT V 5.9 for India customers
- ▶ Procedure and steps to install ISPF-PT on z/OS environment

## 10.9 Ordering ISPF-PT V5.9 for India customers

In this section, we provide information about the procedure to order ISPF-PT V5.9, Shopzseries, which is the most commonly used mode for buying IBM z/series software products, is not available for India customers. The supply of the ISPF Productivity Tool is as a Custom-Built Product Delivery Offering (CBPDO, 5751-CS3).

India customers can order ISPF-PT V5.9 by calling the IBM help desk for India/SA, which is also known as Shop by phone. The details about the IBM help desk are available on:

<http://www.ibm.com>

The help desk technician gathers information about the customer and hardware environment to install ISPF-PT.

India customers can also place an order using e-mail. Send an e-mail to:

[inswtech@in.ibm.com](mailto:inswtech@in.ibm.com)

In the e-mail, state the interest to procure ISPF-PT V5.9. This internet e-mail ID stands for India/SA Techline Support. An India/SA Techline Specialist responds to the customer using e-mail or the telephone, as per the scenario.

## 10.9.1 India/SA Techline - Process

India/SA Techline accepts the e-mail request from the customer and assigns priority as follows:

- ▶ VERY HIGH and HIGH priority – one working day turnaround time (Requires customer name and estimated size of opportunity - \$USD)
- ▶ MEDIUM priority - two working days turnaround time
- ▶ LOW priority - three working days or more turnaround time (Default priority)

India/SA Techline Specialist require the following mandatory information to proceed with an order for ISPF-PT 5.9:

- ▶ Customer Information

Techline Specialists need the customer number, customer name, an existing customer of IBM, or new, and any other relevant details on the customer are helpful to place the order.

- ▶ Hardware specification of z-series system

Techline Specialists need the customer's hardware specification of the z-series system on which the customer plans to install ISPF-PT.

- ▶ Software information about ISPF-PT

India/SA Techline categorizes ISPF-PT under Application Development and AD Tool Solutions. Type and Model configuration for IPT V5.9 is 5698-A81.

- ▶ Additional information

The customer can provide any specific instructions for processing the order, which includes suggestions on the mode of delivery of ISPF-PT.

Figure 10-43 shows a sample request for ISPF-PT V5.9 along with other IBM software.

```
Customer information
Customer number 990535
Customer name ABC Solutions

Hardware
The machine is a 2094-S08 model 704 rated at 298 MSU.

Software List
z/OS                5694-A01      1.8
ISPF Productivity Tool      5698-A81      5.9
SDF II                  5665-366      1.7
CICS Transaction Gateway 5655-M69      6.1
DB2 UDB (version 9)      5635-DB2
WebSphere App Server      5655-N01      6.0.1

Additional information
1. These are all z/OS software.
2. Configure all the software in serverpac format.
3. Configure all the software in 3592 format.
4. Please, do not configure in any other media.
5. Configure the latest version.
6. Configure ISPF Productivity Tool as CBDPO tape product.
```

Figure 10-43 Sample request

The India/SA Techline Specialist verifies and validates the information that the customer provides. Techline Specialist prepare a configuration file that is referred to as a config file. The config file is sent to CSO to proceed with the ISPF-PTorder. CSO submits the configured orders to a system called CBS (SAP) for processing and follows it to final delivery to the customer.

Archived

# Implementing an OLIST User Object Interface

The objective of this exercise is to implement an OLIST *user-defined object* (UDO). *User-objects* are installation defined. The ISPF Productivity Tool (ISPF-PT) invokes the functionality of your choice to handle the user objects. User-defined objects are specified with a leading greater-than sign, for example, >MYOBJECT.

A skeleton UDO is provided and more usefully a specific sample, where the defined object is a job name and various commands perform different functions. The *ISPF Productivity Tool V5 R9 M0 Installation and Customization Guide* makes suggestions of other uses for the UDO.

In this Appendix, we analyze and rationalize the skeleton and sample. The supplied sample is deliberately simple so that you can easily understand its functionality. We clarify the functionality and later expand on it.

The *ISPF Productivity Tool V5 R9 M0 Installation and Customization Guide* manual covers the customization of ISPF-PT by example, showing a set of screen captures of the several windows within the supplied Wizard dialog.

## A.1 Proposed method for exploring user-defined objects

The following process is the method we use to explore the user-defined objects and explain how you can tailor them:

1. Create the development environment and the new LOGON procedure.
2. Convert the two sample CLISTs into REXX.
3. Run the wizard to activate the REXX 'output' exec.
4. Extend the latter REXX exec into a general purpose user-defined object.
5. Summarize how this is done for those impatient to *just-do-it!*

### A.1.1 Creating the development environment and new LOGON procedure

You can edit the supplied ISPF-PT libraries directly; however, there is a more professional approach. So, initially we need to create a development copy of the SIQICLIB and SIQIPLIB libraries.

Create a new LOGON procedure, especially for the development and testing of new or modified components in these libraries. Ours is called ADTTEST, as shown in Figure A-1.

```
Menu Reflist Refmode Utilities Settings Test Help Exit
-----
-IPT--L1          BROWSE - ENTRY PANEL
COMMAND ==>
HOTBAR?

ISPF LIBRARY:
  Project ==> SE16661
  Group   ==> USER      ==>      ==>      ==>
  Type    ==> JCL
  Member  ==>           (Blank or pattern for selection list)
Other data set, @H, or @L for 'SYS1.LOGON(ADTTEST)':
  DSN/Cat. level ==> 'SYS1.LOGON(ADTTEST)'
  Volume serial ==>      (Optional VOLSER or pattern for selection list)
  Password      ==>      (If password protected)
  Default process ==> B    (B=Browse, V=View, E=Edit)
  Execute TAILOR ==> N    (Y=Yes ,N=no, D=define commands)
EDIT/VIEW parameters:
  Initial Macro ==>      Confirm Cancel/Move/Replace ==> N (Y, N)
  Profile Name  ==>      Action Bar in Edit/View      ==> Y (Y, N)
  Format Name    ==>      Highlight coloring in Edit/View ==> Y (Y, N)
                                     Exclusive access of viewed file ==> Y (Y, N)
Preserve VB record length => N Mixed Mode (NLS DBCS char. set) ==> N (Y, N)
```

Figure A-1 Select the LOGON procedure

Figure A-2 on page 241 shows the LOGON procedure we use for creating test objects.

```

Menu Utilities Compilers Help

-IPT- BROWSE SYS1.LOGON(ADTTEST) - 01.01 Line 00000000 Col 001 080
Command ==> Scroll ==> CSR
***** Top of Data *****
//ADTTEST PROC 00010001
//***** 00020000
//* ADTTEST LOGON PROCEDURE (USED FOR TEST NEW IPT) 00030001
//* ALLOCATIONS ARE DONE IN "CENTER.CLIST(ADTTEST)" 00040001
//***** 00050000
//SYSUSER EXEC PGM=IKJEFT01,DYNAMNBR=250,PARM='%ADTTEST' 00060001
//STEPLIB DD DISP=SHR,DSN=IMS.V9R1.SDFSRESL 00061000
//* DD DISP=SHR,DSN=LPA1ST.IPT.V5R9.SIQILPA 00061100
// DD DISP=SHR,DSN=COBOL.V3R4.SIGYCOMP 00062000
//SYSPROC DD DISP=SHR,DSN=CENTER.CLIST 00070000
//SYSHELP DD DISP=SHR,DSN=SYS1.HELP 00080000
//SYSLBC DD DISP=SHR,DSN=SYS1.BROADCAST 00090000
//SYSPRINT DD TERM=TS,SYSOUT=X 00100000
//SYSTEM DD TERM=TS,SYSOUT=X 00110000
//SYSIN DD TERM=TS 00120000
//* 00130000
***** Bottom of Data *****

```

Figure A-2 JCL LOGON procedure used with this example

The JCL LOGON procedure is relatively simple to provide flexibility. The checks and allocations are done within the initial CLIST, which is also called ADTTEST. Example A-1 shows the logon we are using to creating test objects.

Example: A-1 Shows the actual logon CLIST

```

PROC 0
/*****
**/
/* ADTTEST LOGON CLIST */
/* ADTTEST LOGON PROC IS IN SYS1.LOGON */
/* USED WITH BASE SYSTEM */
/*****
**/
CONTROL MAIN NOFLUSH NOMSG NOLIST NOCONLIST
PROFILE MODE WTPMSG MSGID
FREE FILE(ISPLLIB,ISPLLIB,ISPLLIB,ISPTLIB,ISPSLIB, +
          SYSEXEC,ISPPROF,ISPTABL)
/*****
**/
/* ALLOCATE ISPPROF DATASET */
/*****
**/
SET &DSNAME = &SYSUID..ISPF.ISPPROF

ALLOC FI(ISPPROF) SHR DA('&DSNAME.')
IF &LASTCC = 0 THEN +
DO
FREE FI(ISPCRTE)
CONTROL MSG
ATTRIB ISPCRTE DSORG(PO) RECFM(F B) LRECL(80) BLKSIZE(6160)
ALLOC DA('&DSNAME.') SP(2,1) TRACKS DIR(5) USING(ISPCRTE) +
FI(ISPPROF)
IF &LASTCC = 0 THEN +
DO

```

```

        WRITE *** ISPF PROFILE DATA SET '&DSNAME.' HAS BEEN CREATED
        END
    ELSE +
        DO
            WRITE *** UNABLE TO ALLOCATE ISPF PROFILE DATA SET '&DSNAME.'
            FREE FI(ISPCRTE)
            EXIT CODE(12)
        END
    FREE FI(ISPCRTE)
    END
/*****
**/
/* ALLOCATE HFS      DATASET                                */
/*****
**/
/*CONTROL NOMSG                                           */
/*SET &HFSNAME = &SYSUID..HFS                             */
/*                                                         */
/*ALLOC FI(HFSFILE) SHR  DA('&HFSNAME.')

```



```

'CBC.SCBCUTL' /* C++ */ +
'SYS1.DGTCLIB' /* DFSMS™ */ +
'DFSORT™.SICECLIB' /* DFSORT */ +
'FFST™.SEPWCENU' /* FFST */ +
'SYS1.SCBDCLST' /* HCD */ +
'ISP.SISPCLIB' /* ISPF */ +
'RMF™.SERBCLS' /* RMF */ +
'SYS1.HRFCLST' /* RACF */ +
'GIM.SGIMCLS0' /* SMP/E */ +
'ICQ.ICQCCLIB' /* TSOE */ +
'CCCA.V2R1.SABJCLST' /* CCCA */ +
'DEBUG.V7R1.SEQAEXEC' /* DEBUG TOOL */ +
'DB2CFG.DB2TOOLS.CLIST' /* DB2TOOLS */ +
'DB2.V8R1.SDSNCLST') /* DB2 */

/*****
**/
/* ALLOCATE SYSEXEC DATASETS */
/*****
**/
ALLOC FI(SYSEXEC) SHR DA( +
'SYS1.SBPXEXEC' /* OMVS */ +
'BOOKMAN.SEOYCLIB' /* BOOKMAN */ +
'SYS1.SEDGEXE1' /* DFSMS */ +
'FFST.SEPWSRC1' /* FFST */ +
'ISP.SISPEXEC' /* ISPF */ +
'ISF.SISFEXEC' /* SDSF */ +
'DCE.SEUVEXEC' /* DCE */ +
'DFS™.SIOEEXEC' /* DFS */ +
'INFOPRT.SAOPEXEC' /* INFOPRINT */ +
'QMF™.V7R1M0.SDSQEXCE' /* QMF */ +
'FILEMGR.V7R1.SFMNEXEC' /* DB2 FM */ +
'LDAP.SGLDEXEC') /* LDAP */

WRITENR .
/*****
**/
/* ALLOCATE ISPLLIB DATASETS */
/*****
**/
ALLOC FI(ISPLLIB) SHR DA( +
'IPT.V5R9.SIQILOAD' /* IPT */ +
'CENTER.LOADLIB' /* LOCAL */ +
'SYS1.DFQLLIB' /* DFSMS */ +
'SYS1.DGTLLIB' /* DFSMS */ +
'DEBUG.V7R1.SEQAMOD' /* DEBUG TOOL */ +
'CCCA.V2R1.SABJMOD1' /* CCCA */ +
'QMF.V7R1M0.SDSQLOAD' /* QMF */ +
'DB2.V8R1.SDSNLOAD' /* DB2 */ +
'SYS1.SCBDHENU') /* HCD */

WRITENR .
/*****
**/
/* ALLOCATE ISPLMLIB DATASETS */
/*****
**/
ALLOC FI(ISPLMLIB) SHR DA( +

```

```

' IPT.V5R9.SIQIMLIB'      /* IPT          */ +
' CENTER.ISPMLIB'         /* LOCAL        */ +
' SYS1.SBLMSG0'           /* IPCS         */ +
' SYS1.SBPXMENU'          /* OMVS         */ +
' BOOKMAN.SEOYBENU'       /* BOOKMAN      */ +
' BOOKMAN.SEOYMENU'       /* BOOKMAN      */ +
' CBC.SCBCIMG'            /* C++          */ +
' SYS1.DGTMLIB'           /* DFSMS        */ +
' SYS1.DFQMLIB'           /* DFSMS        */ +
' SYS1.SEDGMENU'          /* DFSMS        */ +
' DFSORT.SICEMENU'        /* DFSORT       */ +
' GDDM®.SADMMMSG'         /* GDDM         */ +
' SYS1.SCBDMENU'          /* HCD          */ +
' ICSF.SCSFMSG0'          /* ICSF         */ +
' ISP.SISPMENU'           /* ISPF         */ +
' RMF.SERBMENU'           /* RMF          */ +
' SYS1.SISTMSG0'          /* VTAM®        */ +
' ISF.SISFMLIB'           /* SDSF         */ +
' SYS1.HRFMSG'            /* RACF         */ +
' GIM.SGIMMENU'           /* SMP/E        */ +
' ICQ.ICQMLIB'            /* TSO/E        */ +
' DCE.SEUVMSG'            /* DCE          */ +
' DFS.SIOEMSGE'           /* DFS          */ +
' INFOPRT.SAOPMENU'       /* INFOPRINT    */ +
' CCCA.V2R1.SABJMLIB'     /* CCCA         */ +
' DEBUG.V7R1.SEQAMENU'    /* DEBUG TOOL   */ +
' DB2.V8R1.SDSNSPFM'      /* DB2          */ +
' TCP/IP.SEZAMENU')       /* TCP/IP       */ +

WRITENR .
/*****
**/
/* ALLOCATE ISPPLIB DATASETS */
/*****
**/
ALLOC FI(ISPPLIB) SHR DA( +
' IPT.TEST.PLIB'          /* IPT TEST     */ +
' IPT.V5R9.SIQIPLIB'      /* IPT          */ +
' CENTER.ISPPLIB'         /* LOCAL        */ +
' SYS1.SBLSPNL0'          /* IPCS         */ +
' SYS1.SBPXPENU'          /* OMVS         */ +
' BOOKMAN.SEOYAENU'       /* BOOKMAN      */ +
' BOOKMAN.SEOYPENU'       /* BOOKMAN      */ +
' CBC.SCBCPNL'            /* C++          */ +
' SYS1.DGTPLIB'           /* DFSMS        */ +
' SYS1.DFQPLIB'           /* DFSMS        */ +
' SYS1.SEDGPENU'          /* DFRMM        */ +
' DFSORT.SICEPENU'        /* DFSORT       */ +
' FFST.SEPWPENU'          /* FFST         */ +
' GDDM.SADMPNL'           /* GDDM         */ +
' SYS1.SCBDPENU'          /* HCD          */ +
' ICSF.SCSFPNL0'          /* ICSF         */ +
' ISP.SISPPENU'           /* ISPF         */ +
' SYS1.SHASPNL0'          /* JES2         */ +
' RMF.SERBPENU'           /* RMF          */ +
' ISF.SISFPLIB'           /* SDSF         */ +

```

```

        'SYS1.SISTPNL0'          /* VTAM          */ +
        'SYS1.SISTPNL1'          /* VTAM          */ +
        'SYS1.HRFPANL'           /* RACF          */ +
        'GIM.SGIMPENU'           /* SMP/E         */ +
        'ICQ.ICQPLIB'            /* TSO/E         */ +
        'DCE.SEUVPNL'            /* DCE           */ +
        'DFS.SIOEPNLE'           /* DFS           */ +
        'INFOPRT.SAOPPENU'        /* INFOPRINT     */ +
        'CCCA.V2R1.SABJPLIB'      /* CCCA          */ +
        'DEBUG.V7R1.SEQAPENU'     /* DEBUG TOOL    */ +
        'DB2.V8R1.SDSNSFPF'       /* DB2           */ +
        'TCPIP.SEZAPENU')         /* TCPIP         */ +

```

WRITENR .

```

/*****
**/

```

```

/* ALLOCATE ISPSLIB DATASETS                                     */
/*****
**/

```

```

ALLOC FI(ISPSLIB) SHR DA( +
        'IPT.V5R9.SIQISLIB'      /* IPT           */ +
        'CENTER.ISPSLIB'         /* LOCAL         */ +
        'SYS1.SBLSKEL0'          /* IPCS          */ +
        'BOOKMAN.SEOYSENU'       /* BOOKMAN       */ +
        'CBC.SCBCSKL'            /* C++           */ +
        'SYS1.DGTSLIB'           /* DFSMS         */ +
        'SYS1.SEDGSKL1'          /* DFSMS         */ +
        'DFSORT.SICESLIB'        /* DFSORT        */ +
        'ICSF.SCSFSKL0'          /* ICSF          */ +
        'ISP.SISPSENU'           /* ISPF          */ +
        'ISP.SISPSLIB'           /* ISPF          */ +
        'ISF.SISFSLIB'           /* SDSF          */ +
        'SYS1.HRFSKEL'           /* RACF          */ +
        'GIM.SGIMSENU'           /* SMP/E         */ +
        'CCCA.V2R1.SABJSLIB'      /* CCCA          */ +
        'DEBUG.V7R1.SEQASENU'     /* DEBUG TOOL    */ +
        'DB2.V8R1.SDSNSPFS'       /* DB2           */ +
        'ICQ.ICQSLIB')           /* TSO/E         */ +

```

WRITENR .

```

/*****
**/

```

```

/* ALLOCATE ISPTLIB DATASETS                                     */
/*****
**/

```

```

ALLOC FI(ISPTLIB) SHR DA('&DSNAME' +
        'IPT.V5R9.SIQITLIB'      /* IPT           */ +
        'CENTER.ISPTLIB'         /* LOCAL         */ +
        'SYS1.SBLSTBL0'          /* IPCS          */ +
        'SYS1.SBPXTENU'          /* OMVS          */ +
        'BOOKMAN.SEOYTENU'       /* BOOKMAN       */ +
        'SYS1.DGTTLIB'           /* DFSMS         */ +
        'SYS1.SEDGTBL1'          /* DFSMS         */ +
        'DFSORT.SICETLIB'        /* DFSORT        */ +
        'SYS1.SCBDTENU'          /* HCD           */ +
        'ISP.SISPTENU'           /* ISPF          */ +
        'RMF.SERBT'              /* RMF           */ +

```

```

        'RMF.SERBTENU'           /* RMF          */ +
        'ISF.SISFTLIB'          /* SDSF         */ +
        'GIM.SGIMTENU'          /* SMP/E        */ +
        'ICQ.ICQTABLS'          /* TSO/E        */ +
        'CCCA.V2R1.SABJTLIB'     /* CCCA         */ +
        'DEBUG.V7R1.SEQATLIB'    /* DEBUG TOOL   */ +
        'DB2.V8R1.SDSNSPFS'      /* DB2          */ +
        'TCPIP.SEZAPENU')        /* TCPIP        */

WRITENR .
/*****
**/
/* ALLOCATE SYSHELP DATASETS */
/*****
**/
FREE FI(SYSHELP)
ALLOC FI(SYSHELP) SHR DA( +
        'SYS1.HELP'             /* MVS          */ +
        'SYS1.SEDGHLP1'         /* DFSMS        */ +
        'ISP.SISPHELP'          /* ISPF         */ +
        'SYS1.HELPEP'           /* TSO/E        */ +
        'TCPIP.SEZAHLP')        /* TCPIP        */
/*****
**/
/* ALLOCATE GDDM DATASETS */
/*****
**/
ALLOC FI(ADMP) SHR REU DA('GDDM.SADMP')
ALLOC FI(ADMPROJ) SHR REU DA('GDDM.SADMP')
ALLOC FI(ADMIMG) SHR REU DA('GDDM.SADMP')
ALLOC FI(ADMIMP) SHR REU DA('GDDM.SADMP')
ALLOC FI(ADMGGMAP) SHR REU DA('GDDM.SADMP')
ALLOC FI(ADMGDF) SHR REU DA('GDDM.SADMP')
ALLOC FI(ADMSYMBL) SHR REU DA('GDDM.SADMSY')
/*****
**/
/* ALLOCATE MISC DATASETS */
/*****
**/
ALLOC FI(SYSTCPD) SHR REU DA('CENTER.PARMLIB(TCPDATA)') /* TCPIP */
ALLOC FI(SMPTABL) SHR REU DA('&DSNAME.') /* SMP/E */
ALLOC FI(CIDTABL) SHR REU DA('GIM.CIDTABL') /* SMP/E */
ALLOC FI(IPITLIB) SHR DA('IPT.V5R9.SIQITLIB') /* IPT */
ALLOC FI(ISPILIB) SHR REU DA('ISP.SISPSAMP') /* ISPF */
ALLOC FI(ISPTABL) SHR DA('&DSNAME.') /* ISPF */
ALLOC FI(SDSFDUMP) SHR DA(*) /* SDSF */
ALLOC FI(ICQAATAB) SHR REU DA('ICQ.ICQAATAB') /* TSO/E */
ALLOC FI(ICQABTAB) SHR REU DA('ICQ.ICQABTAB') /* TSO/E */
ALLOC FI(ICQANTAB) SHR REU DA('ICQ.ICQANTAB') /* TSO/E */
/*****
**/
/* ISSUE INFO MESSAGES */
/*****
**/
WRITE
BROADCAST

```

```

/*****
**/
/* RECEIVE MAIL */
/*****
**/
CONTROL MSG PROMPT
RECEIVE
CONTROL NOMSG
/*****
**/
/* START ISPF */
/*****
**/
PDF PANEL(ISR@390)
/*****
**/
EXIT

```

---

Logging on with the new LOGON procedure shows your test libraries ahead of the installation libraries.

To check that your test libraries are ahead of the installation libraries, we could use a number of TSO commands, but it is better to embrace the full functionality of the ISPF-PT environment. The OLIST @DD (or its shortcut OLDD) method is better and a good habit to adopt. In Figure A-3, we use the OLDD command to display both the current SYSPROC and ISPLLIB allocations.

Figure A-3 shows the SYSPROC list.

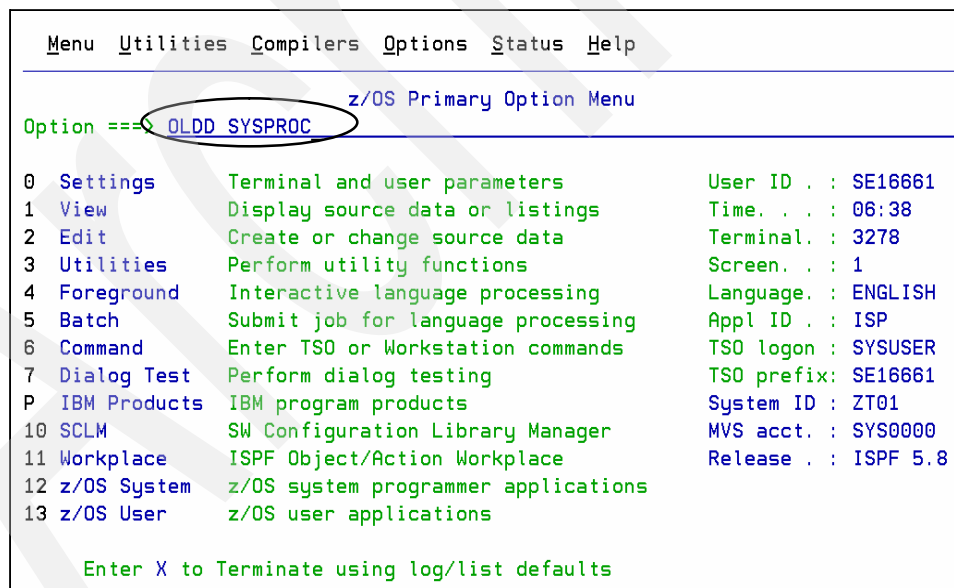


Figure A-3 Using the OLDD command

Figure A-4 on page 248 shows the results of the OLDD command.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (B) ----- ALLOCATION LIST FOR "SYSPROC" -- Row 1 to 19 of 19
Command ===> SCROLL ===> PAGE
Hotbar: FLIP CLRVOL FILLVOL REFRESH UTIL CUT SET UPDATE
*TEMPORARY LIST*

TSO PARMS ===>
Command Member Numbr Data Set Names / Objects Volume
-----
-SYSPROC
1 'IPT.TEST.CLIB' Z00D16
2 'IPT.V5R9.SIQCCLIB' Z00D17
3 'CENTER.CLIST' Z00CAT
4 'SYS1.SBLSCLIO' Z00RES
5 'BOOKMAN.SEOYCLIB' Z00OS3
6 'CBC.SCBCUTL' Z00OS4
7 'SYS1.DGTCLIB' Z00RES
8 'DFSORT.SICECLIB' Z00OS4
9 'FFST.SEPWCENU' Z00OS2
10 'SYS1.SCBDCCLST' Z00RES
11 'ISP.SISPCLIB' Z00RES
12 'RMF.SERBCLS' Z00OS4
13 'SYS1.HRFCLST' Z00RES
14 'GIM.SGIMCLS0' Z00RES
15 'ICQ.ICQCCLIB' Z00OS2
16 'CCCA.V2R1.SABJCLST' Z00D09
17 'DEBUG.V7R1.SEQAEXEC' Z00D05
18 'DB2CFG.DB2TOOLS.CLIST' DB2G01
19 'DB2.V8R1.SDSNCLST' Z00D04
----- END OF LIST -----

```

Figure A-4 Results of the OLDD SYSPROC command

For this development, it is better and easier, to start by setting up your own permanent OLIST. Ours OLIST was named IPTRED, as shown in Figure A-5.

```

Menu Utilities Compilers Options Status Help
-----
z/OS Primary Option Menu
Option ==> ol iptred
-----
0 Settings Terminal and user parameters User ID . : SE16661
1 View Display source data or listings Time. . . : 10:15
2 Edit Create or change source data Terminal. : 3278
3 Utilities Perform utility functions Screen. . : 1
4 Foreground Interactive language processing Language. : ENGLISH
5 Batch Submit job for language processing Appl ID . : ISP
6 Command Enter TSO or Workstation commands TSO logon : SYSUSER
7 Dialog Test Perform dialog testing TSO prefix: SE16661
P IBM Products IBM program products System ID : ZT01
10 SCLM SW Configuration Library Manager MVS acct. : SYS0000
11 Workplace ISPF Object/Action Workplace Release . : ISPF 5.8
12 z/OS System z/OS system programmer applications
13 z/OS User z/OS user applications

Enter X to Terminate using log/list defaults

```

Figure A-5 Using a permanent OLIST to logically collect your project activities

Figure A-6 shows the numbering reference.

File	Edit	Find	Display	Populate	Settings	Menu	Util	Test	Help	Exit
-----										
-IPT- OLIST (B) ----- USER DEFINED OBJECTS ----- "A" will display assist										
Command	===>							SCROLL	===>	PAGE
Hotbar:	FLIP	-	CLR VOL	FILL VOL	REFRESH	UTIL	CUT	SET		UPDATE
Open list	===>		IPTRED	(or BLANK for reference list)						
TSO PARMS	===>									
Command	Member	Numbr	Data Set Names / Objects							Volume
-----										
		1	!----- TAILORED LIBRARIES							
	IQI\$OUT	2	'IPT.TEST.CLIB'							
		3	'IPT.TEST.PLIB'							
		4	!----- BASE COMPONENTS							
-V		5	'IPT.V5R9.SIQ*'							
		6	!----- CURRENT ALLOCATIONS							
		7	:LISTA							
-B	ADTTEST	8	'SYS1.LOGON'							
-B	ADTTEST	9	'CENTER.CLIST'							
		10	!----- USER-DEFINED OBJECT							
		11	>SE16661J							
----- END OF LIST -----										

Figure A-6 Numbering reference

The following list contains the Number reference. Note that the commenting allows for easy identification and grouping of the objects:

- ▶ A comment to describe the category of the group following
- ▶ The test CLIST/REXX library, CLIST member IQI\$OUT
- ▶ The test Panel library
- ▶ A comment to describe the category of the group following
- ▶ A dynamic OLIST containing all of the base system libraries for ISPF-PT 5.9
- ▶ A comment to describe the category of the group following
- ▶ A dynamic OLIST that contains all ddnames and data sets in the current allocation
- ▶ The LOGON procedure JCL library, member ADTTEST
- ▶ The logon CLIST library, member ADTTEST
- ▶ A comment to describe the category of the group following
- ▶ The UDO that is being developed and tested

Only a few more objects can be added and changes made so that the OLIST evolves as the project proceeds. Refer to the *ISPF Productivity Tool for z/OS Users Guide* to see the full wealth of options.

Figure A-6 gives a temporary list that shows all of the current allocations. We can exclude all of the entries that are not currently of interest. Note that the \*EXCLUDE\* flag shows that some lines are excluded, and it shows the gaps between the line numbers in Figure A-7 on page 250.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (B) ----- ALLOCATION LIST ----- Row 28 from 174
Command   ==>
Hotbar: FLIP  CLRVOL  FILLVOL  REFRESH  UTIL  CUT  SET  UPDATE
*TEMPORARY LIST*

TSO PARMS ==>
Command  Member  Numbr Data Set Names / Objects  *EXCLUDE* Volume
-----
-SYSPROC          28 'IPT.TEST.CLIB'          Z00D16
                  29 'IPT.V5R9.SIQCCLIB'      Z00D17
                  30 'CENTER.CLIST'          Z00CAT
-ISPLLIB          105 'IPT.TEST.PLIB'          Z00D08
                  106 'IPT.V5R9.SIQCPLIB'      Z00D07
                  107 'CENTER.ISPLLIB'        Z00CAT
----- END OF LIST -----

```

Figure A-7 Using a dynamic list and excluding what is not of interest

At this installation of ISPF-PT that we used for this example, the two libraries that we show in Figure A-7, SYSPROC and ISPLLIB, are shown as numbers 28 and 105. The base installation libraries are shown below them as numbers 29 and 106. All other libraries were conveniently excluded from view in this OLIST view of a :LISTA command by using the /X and /XX commands.

## A.1.2 Converting the two sample CLISTs into REXX

There are two supplied sample user objects in the ISPF Productivity Tool distributed CLIST library, as shown in Table A-1.

Table A-1 Sample user objects

User object	Description
IQI\$USER	This is the sample CLIST interface that handles user defined objects.
IQI\$OUT	This is the IPT CLIST interface that demonstrates how the user defined object interface can be used to provide job processing via the TSO Status, Cancel, and Output commands.

CLIST language is very powerful, but it does not have the elegance or readability of REXX.

Converting the CLISTs into REXX versions makes the CLISTs easier to understand and customize.

Example A-2 on page 251 and Example A-4 on page 255 show the original CLISTs IQI\$USER and IQI\$OUT. The examples are followed in Example A-3 on page 252 and Example A-5 on page 259 by their respective REXX versions, which are not exactly equivalent in many cases because REXX has more elegant constructs. The IQI\$USER has been more literally translated because its structure really only accommodates the most simple cases. You would need to do work to make a more comprehensive UDO.

You might identify certain improvements in functionality that were made. You might also see several opportunities for further enhancement of functionality and might, very likely, be able to improve on the examples we supplied here.



Example: A-2 Supplied IQI\$USER CLIST

```

CONTROL MAIN MSG NOPROMPT /*LIST CONLIST SYMLIST /*TRACE OPTIONS */
/*-----*/
/* 5698-A81 (C) COPYRIGHT IBM CORP 1992,2006 */
/* ISPF PRODUCTIVITY TOOL: 5.9.0 */
/*-----*/
/*-----*/
/* THIS IS THE SPIFFY CLIST INTERFACE THAT HANDLES USER DEFINED */
/* OBJECTS. */
/*-----*/
/* (C) COPYRIGHT 1992 ISOGON CORPORATION */
/*-----*/
/* THE FOLLOWING VARIABLES CAN BE USED TO CONTROL THE OPERATION OF */
/* THE INTERFACE: */
/* - &LETNEST - DETERMINES IF NESTED CALLS TO THIS CLIST ARE ALLOWED.*/
/* SET TO "Y" IF ALLOWED, "N" IF NOT ALLOWED. */
/* - &APPLID - DEFINES THE REQUIRED ISPF APPLICATION ID FOR THIS */
/* CLIST. (SAME AS NEWAPPL(XXXX) PARAMETER IN THE */
/* ISPEXEC SELECT STATEMENT). */
/* LEAVE AS BLANKS IF NONE REQUIRED. */
/* - &OBJNAME - A CHARACTER STRING DEFINING THE NAME OF THE USER */
/* OBJECT. THIS STRING APPEARS IN MESSAGES. */
/* */
/* - &DEBUG - IF SET TO "Y", WILL DISPLAY THE LIST OF PASSED */
/* PARAMETERS AND THEIR VALUES. */
/* */
/*-----*/
ERROR RETURN /*HANDLE CLIST ERRORS */
SET APPLID= /*CONTROL VARIABLE. SEE COMMENTS ABOVE */
SET OBJNAME=&STR(USER OBJECTS) /*CONTROL VARIABLE SEE COMMENTS ABOVE */
SET LETNEST=Y /*CONTROL VARIABLE. SEE COMMENTS ABOVE */
SET DEBUG=N /*CONTROL VARIABLE. SEE COMMENTS ABOVE */
ISPEXEC CONTROL ERRORS RETURN /*HANDLE ISPF ERRORS */
SET SHORTMSG= /*CLEAR SHORT MESSAGE */
/*----- STEP 1: HANDLE NESTING REQUIREMENTS -----*/
IF (&LETNEST=N) THEN DO /*IF NESTING IS CONTROLLED */
    SET MAXCC=0
    ISPEXEC TBCREATE &SYSICMD KEYS(USEROBJ) REPLACE NOWRITE
    IF (&MAXCC=4) THEN DO /*NESTING ALREADY EXISTS? */
        SET SHORTMSG= /* */
        SET LONGMSG=&STR(Nested processing of &OBJNAME not allowed)
        ISPEXEC SETMSG MSG(IQIX100) /*IBMIPT 590*/
        EXIT CODE(0) /*EXIT WITH QUEUED MESSAGE */
    END
END
/*----- STEP 2:ENSURE CLIST IS CALLED WITH CORRECT APPLID -----*/
IF (&APPLID=) THEN DO /*IF A SPECIFIC APPLID IS REQUESTED */
    ISPEXEC VGET (ZAPPLID) SHARED /*GET CURRENT ISPF APPLID */
    IF (&ZAPPLID=) THEN DO /*IF DIFFERENT THAN REQUESTED APPLID */
        SET MAXCC=0 /* THEN REINVOKE WITH - */
        ISPEXEC SELECT CMD(%&SYSICMD) NEWAPPL(&APPLID) /*CORRECT APPLID */
        EXIT CODE(&MAXCC) /*EXIT WITH NESTED CLIST RETURN CODE */
    END
END
/*----- LIBDEF NOTES -----*/
/* IF YOUR APPLICATION NEEDS LIBDEF STATEMENTS, YOU MAY INSERT LIBDEF */
/* STATEMENTS HERE (BEFORE STEP3). */
/* LIBDEF STATEMENTS INSERTED HERE SHOULD BE DEACTIVATED LATER */
/* IN THIS CLIST. */
/* LIBDEF EXAMPLE: ISPEXEC LIBDEF ISPLIB DATASET ID('PANSPPF.PANELS') */

```

```

/*----- STEP 3: OBTAIN -IPT- PASSED PARAMETERS -----*/
ISPEXEC TBTOP IPISHARE /*GET -IPT- PASSED PARAMETERS */
ISPEXEC TBSKIP IPISHARE /*RESIDING IN THE IPISHARE TABLE */
/* THE FOLLOWING PARAMETERS ARE AVAILABLE: */
/* &IPIDSN1 = NAME OF OBJECT (UNQUOTED, WITHOUT MEMBER PORTION(IF ANY)*/
/* AND WITHOUT THE LEADING ">" SIGN. */
/* &IPIVOL1 = VOLUME VALUE FROM CALLING APPLICATION */
/* &DEFAULT = REQUESTED COMMAND (E(DIT), B(ROWSE), ETC.) */
/* &MEMBER = MEMBER NAME (OPTIONAL) */
/* &WORD1 = MEMBER PATTERN (OPTIONAL) */
/* &WORD2 = SIGNIFICANT PORTION OF THE PATTERN (OPTIONAL) */
/* &WORD3 = USER SPECIFIED PASSWORD(ON EDIT/BROWSE/VIEW ENTRY PANEL)*/
/*-----*/
IF (&DEBUG=Y) THEN DO /*IF DEBUGGING MODE IS ON */
    WRITE &STR(DEFAULT=&DEFAULT,IPIDSN1=&IPIDSN1,IPIVOL1=&IPIVOL1)
    WRITE &STR(MEMBER=&MEMBER,WORD1=&WORD1,WORD2=&WORD2,WORD3=&WORD3)
END
/*----- STEP 4: PROCESS THE USER REQUEST -----*/
SET MAXCC=0 /*CLEAR RETURN CODE */
ISPEXEC SELECT PGM(IEFBR14) /*REPLACE WITH YOUR OWN PROGRAM OR */
/* CLIST, OR INCLUDE YOU CODE IN HERE */
SET RC=&MAXCC /*SAVE RETURN CODE */
/*----- STEP 5: CLEAR THE NESTING LEVEL -----*/
IF (&LETNEST=N) THEN ISPEXEC TBEND &SYSICMD
/*----- STEP 6: REMOVE LIBDEFS (IF ANY ESTABLISHED)-----*/
/*----- LIBDEF NOTES -----*/
/* IF YOU HAVE ESTABLISHED A LIBDEF EARLIER IN THIS CLIST, */
/* NOW IS THE TIME TO REMOVE THE LIBDEFS. */
/* EXAMPLE: ISPEXEC LIBDEF ISPLIB */
/*-----*/
/*----- STEP 7: TERMINATE PROCESSING -----*/
IF (&RC=0) THEN EXIT CODE(0) /*TERMINATE SUCCESSFULLY */
/* ELSE - A FAILURE -ISSUE MESSAGE & QUIT */
SET LONGMSG=&STR(&IPIDSN1 NOT PROCESSED (RC=&RC)) /* BUILT ERROR MSG */
ISPEXEC SETMSG MSG(IQIX100) /*ISSUE THE MESSAGE*/ /*IBMIPT 590*/
EXIT CODE(8) /*INDICATE A FAILURE */
/*-----*/
/* 5698-A81 (C) COPYRIGHT IBM CORP 1992,2006 */
/* ISPF PRODUCTIVITY TOOL: 5.9.0 */
/*-----*/

```

---

*Example: A-3 Converted IQIRUSER REXX Exec*

---

```

/**REXX(IQIRUSER)*****
/* 5698-A81 (C) COPYRIGHT IBM CORP 1992,2006 */
/* ISPF PRODUCTIVITY TOOL: 5.9.0 */
/*-----*/
/* This is the IPT REXX interface that handles user defined objects */
/*-----*/
/* the following variables can be used to control the operation of */
/* the interface: */
/* - letnest - Determines if nested calls to this clist are allowed.*/
/* set to "Y" if allowed, "N" if not allowed. */
letnest = "Y"
/* - applid - Defines the required ispf application id for this */
/* clist. (same as newappl(xxxx) parameter in the */
/* ispexec select statement). */
/* leave as blanks if none required. */

```

```

applid = ""
/* - objname - A character string defining the name of the user      */
/*              object. this string appears in messages.            */
objname = "USER OBJECTS"
/* - debug    - If set to "Y", will display the list of passed      */
/*              parameters and their values.                        */
debug    = "N"
/*                                                              */
/*-----*/
CALL MSG "ON"
CALL PROMPT "OFF"
CALL ON ERROR
ADDRESS ISPEXEC "CONTROL ERRORS RETURN"
TRACE (tracev)
sysicmd = SYSVAR("SYSICMD")
libdefs = 0
endnest = 0

/*-----*/
/* Step 1: Handle nesting requirements                               */
/* if nesting is controlled then check if nesting already exists    */
/* and if so exit with appropriate message                          */
/*-----*/
IF( &letnest = "N" )THEN DO
    "TBCREATE" sysicmd "KEYS(USEROBJ) REPLACE NOWRITE"
    IF( RC=4 )THEN
        CALL CTLEXIT 0,"","Nested processing of" objname "not allowed"
    END
ELSE NOP

/*-----*/
/* Step 2: Ensure clist is called with correct applid              */
/* If a pecific applid is requested then get current ispf applid and */
/* if different reinvoke with correct applid and exit with nested   */
/* command return code                                              */
/*-----*/
IF( applid<>"")THEN DO
    "VGET (ZAPPLID) SHARED"
    IF( zapplid<>applid )THEN DO
        "SELECT CMD("sysicmd") NEWAPPL(&APPLID)"
        EXIT (RC)
    END
ELSE NOP

/*----- LIBDEF Notes -----*/
/* If your application needs LIBDEF or ALTLIB statements then you   */
/* may insert these statements here (before Step3).                 */
/* LIBDEF and ALTLIB statements inserted here should be deactivated */
/* later in this exec.                                              */
/*-----*/

/*-----*/
/* Step 3: OBTAIN -IPT- passed parameters from IPISHARE table      */
/* The following parameters are available:                           */
/* ipidsn1 = Name of object (unquoted, without member portion(if any) */
/*           and without the leading ">" sign.                       */
/* ipivol1 = volume value from calling application                  */
/* default = Requested command (e(dit), b(rowse), etc.)            */
/* member   = Member name (optional)                                */
/* word1    = Member pattern (optional)                             */

```

```

/* word2  = Significant portion of the pattern (optional) */
/* word3  = User specified password(on edit/browse/view entry panel)*/
/*-----*/
"TBTOP IPISHARE"
"TBSKIP IPISHARE"
IF( debug = "Y" )THEN DO
    SAY "DEFAULT="default",IPIDSN1="ipidsn1",IPIVOL1="ipicol1
    SAY "MEMBER="member",WORD1="word1",WORD2="word2",WORD3="word3
    END
ELSE NOP

/*-----*/
/* Step 4: process the user request */
/* Replace with your own program/clist/exec or include you code here */
/* save return code */
/*-----*/
custrc = 0
/* "SELECT PGM(IEFBR14)" */
/* custrc = RC */
SAY "USER OBJECT"

/*-----*/
/* Step 5: Clear the nesting level */
/*-----*/
IF( letnest="N" )THEN "TBEND" sysicmd

/*-----*/
/* Step 6: remove LIBDEFs (if any established) */
/* If you have established a libdef earlier in this exec now is the */
/* time to remove them. */
/*-----*/
/* ISPEXEC LIBDEF ISPPLIB */
/* Step 7: Terminate processing */
/* If the return code from the user exit is 0 the exit with that */
/* otherwise end with return code 8 and an error message. */
/*-----*/
IF (custrc<>0 )THEN
    CALL CTLEXIT 8,"",ipidsn1" not processed (RC="custrc")"
EXIT 0

/*=====*/
/* controlled exit with message */
/*=====*/
CTLEXIT:
ARG exitrc,shortmsg,longmsg
"SETMSG MSG(IQIX100)"
EXIT (exitrc)

/*=====*/
/* error action */
/*=====*/
ERROR: RETURN

/*=====*/
/* 5698-A81 (C) COPYRIGHT IBM CORP 1992,2006 */
/* ISPF PRODUCTIVITY TOOL: 5.9.0 */
/*=====*/

```

*Example: A-4 Supplied IQI\$OUT CLIST*

```

CONTROL MAIN MSG NOPROMPT /*LIST CONLIST SYMLIST /*TRACE OPTIONS */
/*-----*/
/* 5698-A81 (C) COPYRIGHT IBM CORP 1992,2006 */
/* ISPF PRODUCTIVITY TOOL: 5.9.0 */
/*-----*/
/*          (C) COPYRIGHT 1992 ISOGON CORPORATION */
/*-----*/
/* THIS IS THE SPIFFY CLIST INTERFACE THAT DEMONSTRATES HOW THE */
/* USER DEFINED OBJECT INTERFACE CAN BE USED TO PROVIDE JOB PROCESSING*/
/* VIA THE TSO STATUS, CANCEL, AND OUTPUT COMMANDS. */
/* */
/* INSPECT THIS CLIST ALONG WITH IQI$USER. THIS CLIST IS AN ADAPTATION*/
/* OF IQI$USER TO PROVIDE SIMPLE JOB MANAGEMENT FUNCTIONS AS FOLLOWS: */
/* */
/* THE OBJECT NAME IS A JOBNAME. IT CAN BE SPECIFIED AS >JOBNAME */
/* OR >JOBNAME(JOBNUMBER) ON THE EDIT/BROWSE/VIEW AND OLIST SCREENS. */
/* */
/* THE SUPPORTED COMMANDS ARE: */
/* "B" - DISPLAY (BROWSE) JOB OUTPUT (PROVIDED IT HAS AN HELD SYSOUT) */
/* "E" - EDIT A COPY OF THE JOB'S OUTPUT (PROVIDE CUT/PASTE ABILITY */
/*       FOR JOB'S LISTINGS.) */
/* "V" - DISPLAY JOB STATUS */
/* "D" - DELETE JOB (CANCEL WITH PURGE) - AVAILABLE UNDER PLIST. */
/* */
/* THIS CLIST REQUIRES THAT THE OBJECT NAME (JOBNAME) IS SPECIFIC. */
/* THIS MEANS THAT EITHER BOTH JOBNAME(JOBNUMBER) ARE SPECIFIED, OR */
/* WHEN JOBNAME IS SPECIFIED (WITHOUT JOBNUMBER), THAT THE JOBNAME */
/* IS UNIQUE. */
/* */
/* SAMPLE INVOCATIONS (FROM PLIST): */
/* */
/* COMMAND MEMBER NUM DATA SET NAME          TYPE */
/* ----- */
/*          1 'KEVIN.TSO.JCL'          PDS */
/* V          2 >RJK002I(JOB123)        USER  <-- REQUEST JOB STATUS*/
/* B          3 >RJK002K                USER  <-- REQUEST TO DISPLAY*/
/*          4 !                        ----    THE JOB'S OUTPUT*/
/* D          5 >RJK002J                USER  <-- REQUEST TO PURGE */
/*          6 !                        ----    THE JOB'S OUTPUT*/
/*          7 'KEVIN.TSO.CLIST'        PDS */
/* */
/* SINCE BROWSE AND VIEW ARE GENERAL COMMANDS, ACCESS TO THE JOBS */
/* IS AUTOMATICALLY AVAILABLE FROM EVERY SCREEN. FOR EXAMPLE, THE */
/* FOLLOWING COMMAND WILL DISPLAY THE OUTPUT OF JOB "RJK002I": */
/* BROWSE >RJK002I */
/* AND THE FOLLOWING WILL DISPLAY A JOB'S STATUS: */
/* VIEW  >RJK002K(JOB0158) */
/* */
/*-----*/
/* NOTE: THIS SPECIFIC MODIFICATION OF IQI$USER ALLOWS NESTING CALLS */
/*       AND DOES NOT REQUIRE ANY APPLICATION ID. */
/* */
/* - &DELFILE - IF SET TO "Y", EACH OUTPUT JOB WORKFILE WILL BE */
/*             DELETED UPON TERMINATION OF THE BROWSE REQUEST. */
/*             OTHERWISE, IT WILL BE KEPT, FOR A SESSION "CLEANUP" */

```

```

/*          CLIST WHICH WILL PRESUMABLY INCLUDE A SINGLE          */
/*          COMMAND LIKE "DELETE '&SYSUID..OUTLIST.*'"          */
/* - &DEBUG      - IF SET TO "Y", WILL DISPLAY THE LIST OF PASSED  */
/*          PARAMETERS AND THEIR VALUES.                      */
/* - &TRACE      - IF SET TO "Y", WILL DISPLAY THE EXECUTED CLIST CODE */
/*          */
/*-----*/
ERROR RETURN          /*HANDLE CLIST ERRORS          */
SET APPLID=           /*NO SPECIFIC APPLICATION ID REQUIRED  */
SET OBJNAME=&STR(JOBS) /*OBJECT NAME                      */
SET LETNEST=Y         /*ALLOW NESTED CALLS                Q */
SET DELFILE=Y         /*CONTROL VARIABLE. SEE COMMENTS ABOVE */
SET DEBUG=N           /*CONTROL VARIABLE. SEE COMMENTS ABOVE */
SET TRACE=N           /*CONTROL VARIABLE. SEE COMMENTS ABOVE */
IF (&TRACE=Y) THEN CONTROL CONLIST SYMLIST LIST
ISPEXEC CONTROL ERRORS RETURN /*HANDLE ISPF ERRORS          */
SET SHORTMSG=         /*CLEAR SHORT MESSAGE              */
SET LONGMSG=          /*CLEAR LONG MESSAGE              */
/*----- STEP 1: HANDLE NESTING REQUIREMENTS -----*/
IF (&LETNEST=N) THEN DO /*IF NESTING IS CONTROLLED          */
    SET MAXCC=0
    ISPEXEC TBCREATE &SYSICMD KEYS(USEROBJ) REPLACE NOWRITE
    IF (&MAXCC=4) THEN DO /*NESTING ALREADY EXISTS?          */
        SET LONGMSG=&STR(Nested processing of &OBJNAME not allowed)
        ISPEXEC SETMSG MSG(IQIX100) /*IBMIPT 590*/
        EXIT CODE(0) /*EXIT WITH QUEUED MESSAGE          */
    END
END
/*----- STEP 2:ENSURE CLIST IS CALLED WITH CORRECT APPLID -----*/
IF (&APPLID=) THEN DO /*IF A SPECIFIC APPLID IS REQUESTED */
    ISPEXEC VGET (ZAPPLID) SHARED /*GET CURRENT ISPF APPLID          */
    IF (&ZAPPLID=) THEN DO /*IF DIFFERENT THAN REQUESTED APPLID */
        SET MAXCC=0 /* THEN REINVOKE WITH -          */
        ISPEXEC SELECT CMD(%&SYSICMD) NEWAPPL(&APPLID) /*CORRECT APPLID */
        EXIT CODE(&MAXCC) /*EXIT WITH NESTED CLIST RETURN CODE */
    END
END
/*----- LIBDEF NOTES -----*/
/* IF YOUR APPLICATION NEEDS LIBDEF STATEMENTS, YOU MAY INSERT LIBDEF */
/* STATEMENTS HERE (BEFORE STEP3). */
/* LIBDEF STATEMENTS INSERTED HERE SHOULD BE DEACTIVATED LATER */
/* IN THIS CLIST. */
/* LIBDEF EXAMPLE: ISPEXEC LIBDEF ISPLIB DATASET ID('PANSPPF.PANELS') */
/*----- STEP 3: OBTAIN -IPT- PASSED PARAMETERS -----*/
ISPEXEC TBTOP IPISHARE /*GET -IPT- PASSED PARAMETERS          */
ISPEXEC TBSKIP IPISHARE /*RESIDING IN THE IPISHARE TABLE          */
/* THE FOLLOWING PARAMETERS ARE AVAILABLE: */
/* &IPIDSN1 = NAME OF THE OBJECT: JOB NAME OR JOBNAME(JOBNUMBER) */
/* &IPIVOL1 = IGNORED */
/* &DEFAULT = REQUESTED COMMAND. B=REQUEST STATUS INFO, V=VIEW OUTPUT */
/*          D=CANCEL (AVAILABLE FROM PLIST) */
/* &MEMBER = MEMBER NAME (IGNORED) */
/* &WORD1 = MEMBER PATTERN (IGNORED) */
/* &WORD2 = SIGNIFICANT PORTION OF THE PATTERN (IGNORED) */
/* &WORD3 = USER SPECIFIED PASSWORD(IGNORED) */
/*-----*/
IF (&DEBUG=Y) THEN DO /*IF DEBUGGING MODE IS ON          */
    WRITE &STR(DEFAULT=&DEFAULT,IPIDSN1=&IPIDSN1,IPIVOL1=&IPIVOL1)
    WRITE &STR(MEMBER=&MEMBER,WORD1=&WORD1,WORD2=&WORD2,WORD3=&WORD3)
END

```

```

/*----- STEP 4: VALIDATE THE PASSED OBJECT NAME-----*/
IF (&TRACE=Y) THEN CONTROL CONLIST SYMLIST LIST
SET SYSDVAL=&NRSTR(1,'(&IPIDSN1 )
READDVAL I LPAREN JOBNAME /* FROM PASSED OBJECT
IF (&TRACE=Y) THEN STATUS &STR(&JOBNAME)
CONTROL NOPROMPT NOCONLIST NOSYMLIST NOLIST
SET SYSOUTTRAP=10
STATUS &STR(&JOBNAME) /*GET STATUS OF CURRENT JOB */
SET SYSOUTTRAP=0
/*IF (&TRACE=Y) THEN CONTROL CONLIST SYMLIST LIST
SET I=&SYSOUTLINE /*NUMBER OF GENERATED OUTPUT INFO LINES*/
IF (&I>1) THEN DO /*TOO MANY OUTPUT LINES? */
ISPEXEC CONTROL DISPLAY LINE START(4)
SET J=1
DO WHILE(&J<=&I OR &J<4) /*DISPLAY UP TO 4 MESSAGE LINES */
SET LINE=&STR(&&SYSOUTLINE&J)
SET LINE=&STR(&LINE)
WRITE &LINE
SET J=&J+1
END
SET LONGMSG=&STR(A SPECIFIC JOBNAME(JOBNUMBER) IS REQUIRED)
GOTO EXIT /*ISSUE ERROR MESSAGE AND EXIT */
END
SET SYSDVAL=&STR(&&SYSOUTLINE&I) /*OBTAINED RETURNED INFO LINE */
SET SYSDVAL=&STR(&SYSDVAL)
READDVAL A
IF (&SUBSTR(1:5,&STR(&A .....))=IKJ56) +
THEN READDVAL MSGID VERB JOBNAME STATUS1 STATUS2 STATUS3 STATUS4
ELSE READDVAL VERB JOBNAME STATUS1 STATUS2 STATUS3 STATUS4
IF (&STR(&STATUS1&STATUS2)=&STR(NOTFOUND)) THEN DO /* IF JOB NOT FOUND*/
SET LONGMSG=&STR(&SYSDVAL) /*ISSUE THE ERROR MESSAGE */
GOTO EXIT /*AND EXIT */
END
SET L=&LENGTH(&JOBNAME)
CONTROL NOPROMPT NOCONLIST NOSYMLIST NOLIST
DO WHILE(&I<&L)
SET C=&SUBSTR(&I:&I,&STR(&JOBNAME))
IF (&C=&LPAREN) THEN DO
SET S=&I+1
SET E=&L-1
SET JOBNUM=&SUBSTR(&S:&E,&STR(&JOBNAME ...))
SET JOBNUM=&STR(&JOBNUM)
SET I=&L
END
SET I=&I+1
END
IF (&TRACE=Y) THEN CONTROL CONLIST SYMLIST LIST
/*----- PROCESS THE V COMMAND (DISPLAY JOB STATUS)-----*/
IF (&DEFAULT=V) THEN DO /*HANDLE STATUS COMMAND */
SET LONGMSG=&STR(&SYSDVAL) /*RETURN STATUS INFORMATION */
GOTO EXIT /*TERMINATE PROCESSING OF THE B COMMAND*/
END
/*----- PROCESS THE D COMMAND (CANCEL JOB )-----*/
IF (&DEFAULT=D) THEN DO /*HANDLE STATUS COMMAND */
CANCEL &JOBNAME PURGE
GOTO EXIT /*TERMINATE PROCESSING OF THE B COMMAND*/
END
/*----- PROCESS THE B COMMAND (LOOK AT THE OUTPUT)-----*/
/*----- PROCESS THE E COMMAND (EDIT THE OUTPUT)-----*/
IF (&DEFAULT=B OR &DEFAULT=E) THEN DO

```

```

IF (&STR(&STATUS1&STATUS2)~=&STR(ONOUTPUT)) THEN DO
  SET LONGMSG=&STR(&SYSDVAL) /*BROWSE REQUIRES JOB BE ON OUTPUT QUE */
  GOTO EXIT /*AND EXIT */
END
SET DSN=&STR(&SYSUID..OUTLIST.&JOBNUM)
IF (&SYSDSN(&DSN)~&OK) THEN DO
  SET MAXCC=0
  ALLOC DA('&DSN') LRECL(133) BLK(13300) DSORG(PS) RECFM(F B A) +
    TRACKS SPACE(200 100) RELEASE NEW CAT
  IF (&MAXCC~&0) THEN DO
    SET LONGMSG=&STR(CANNOT ALLOCATE WORK FILE FOR &JOBNAME, TRY LATER)
    GOTO EXIT
  END
END
CONTROL NOPROMPT NOCONLIST NOSYMLIST NOLIST
SET SYSOUTTRAP=10
OUTPUT &JOBNAME PRINT('&DSN') BEGIN KEEP HOLD
SET SYSOUTTRAP=0
IF (&TRACE=Y) THEN CONTROL CONLIST SYMLIST LIST
SET MAXCC=0
SET I=&SYSOUTLINE /*NUMBER OF GENERATED OUTPUT INFO LINES*/
SET SYSDVAL=&STR(&&SYSOUTLINE&I) /*OBTAIN RETURNED INFO LINE */
SET SYSDVAL=&STR(&SYSDVAL)
READDVAL A
IF (&SUBSTR(1:5,&STR(&A .....))=IKJ56) +
  THEN READDVAL MSGID W1 W2 W3 W4 W5 W6 W7 W8 W9 W10
  ELSE READDVAL W1 W2 W3 W4 W5 W6 W7 W8 W9 W10
IF (&STR(&W1&W2&W3&W4&W5)=&STR(NOHELDOUTPUTFORJOB)) THEN DO
  SET LONGMSG=&STR(&SYSDVAL)
  GOTO EXIT
END
SET MAXCC=0
IF (&DEFAULT=B) +
  THEN ISPEXEC BROWSE DATASET('&DSN')
  ELSE ISPEXEC EDIT DATASET('&DSN') +
    MACRO(!IQIEMAC) PROFILE(IPIVIEW) /*IBMIPT 590*/
SET RC=&MAXCC
CONTROL NOMSG
IF (&DELFILE=Y) THEN DELETE '&DSN'
CONTROL MSG
IF (&RC>4) THEN +
  SET LONGMSG=&STR(OUTPUT NOT AVAILABLE) /*BROWSE FAILED-EMPTY FILE */
  GOTO EXIT /*TERMINATE PROCESSING OF THE B COMMAND*/
END
SET LONGMSG=&STR(INVALID COMMAND "&DEFAULT". VALID COMMANDS: B,E,V,D)
GOTO EXIT /*AND EXIT */
/*----- STEP 5: CLEAR THE NESTING LEVEL -----*/
EXIT: IF (&LETNEST=N) THEN ISPEXEC TBEND &SYSCMD
/*----- STEP 6: REMOVE LIBDEFS (IF ANY ESTABLISHED)-----*/
/*----- LIBDEF NOTES -----*/
/* IF YOU HAVE ESTABLISHED A LIBDEF EARLIER IN THIS CLIST, */
/* NOW IS THE TIME TO REMOVE THE LIBDEFS. */
/* EXAMPLE: ISPEXEC LIBDEF ISPLIB */
/*----- STEP 7: TERMINATE PROCESSING -----*/
IF (&STR(&LONGMSG)~&) THEN +
  ISPEXEC SETMSG MSG(IQIX100) /*ISSUE MESSAGE*/ /*IBMIPT 590*/
EXIT CODE(0)
/*****
/* 5698-A81 (C) COPYRIGHT IBM CORP 1992,2006 */

```



```

/* ISPF PRODUCTIVITY TOOL: 5.9.0 */
/*****

```

*Example: A-5 Converted IQIROUT REXX Exec*

```

/**REXX(IQIROUT)*****
/* 5698-A81 (C) COPYRIGHT IBM CORP 1992,2006 */
/* ISPF PRODUCTIVITY TOOL: 5.9.0 */
/*-----*/
ADDRESS TSO
CALL MSG "OFF"
CALL PROMPT "OFF"
/*-----*/
/* This is an ipt rexx interface that demonstrates how the user */
/* defined object interface can be used to provide job processing */
/* via the TSO STATUS, CANCEL, and OUTPUT commands. inspect this */
/* exec along with iqifuser. this clist is an adaptation of IQIFUSER */
/* to provide simple job management functions as follows: */
/* */
/* the object name is a jobname. it can be specified as >jobname or */
/* >jobname(jobnumber) on BROWSE EDIT & VIEW and OLIST screens. */
/* screens. */
/* */
/* The supported commands are: */
/* "B" - Browse the job output (provided it has any held sysout) */
/* "E" - Edit a copy of the job's output (provide cut/paste ability */
/* for JOB's listings.) */
/* "V" - Display job status */
/* "D" - Delete job (cancel with purge) - available under plist. */
/* */
/* This exec requires that the object name (jobname) is specific. */
/* this means that either both jobname(jobnumber) are specified, or */
/* when jobname is specified (without jobnumber), that the jobname */
/* is unique. */
/* */
/* Sample invocations (from plist): */
/* TSO PARMS ==> */
/* Command Member Numbr Data Set Names / Objects */
/*-----*/
/* */
/* 1 'KEVIN.TSO.JCL' PDS */
/* V 2 >RJK002I(JOB123) USER (1) */
/* B 3 >RJK002K USER (2) */
/* 4 ! ---- */
/* D 5 >RJK002J USER (3) */
/* 6 ! ---- */
/* 7 'KEVIN.TSO.CLIST' PDS */
/* */
/* (1) request job status */
/* (2) request to display the job's output */
/* (3) request to purge the job's output */
/* */
/* Since BROWSE and VIEW are general commands, access to the jobs */
/* is automatically available from every screen. for example, the */
/* following command will display the output of job "RJK002I": */
/* BROWSE >RJK002I */
/* and the following will display a job's status: */
/* VIEW >RJK002K(JOB0158) */
/* */
/*-----*/

```

```

/* NOTE: This specific modification of IQIfUSER allows nesting calls */
/*      and does not require any application id.                      */
/* - objname - Object name                                           */
objname = "JOBS"
/* - letnest - Allows nesting calls                                  */
letnest = "Y"
/* - applid - Does not require any application id.                  */
applid = ""
/* - delfile - If set to "Y", each output job workfile will be      */
/*             deleted upon termination of the browse request.       */
/*             otherwise, it will be kept, for a session "cleanup"  */
/*             clist which will presumably include a single         */
/*             command like "delete 'userid..OUTLIST.*'"            */
delfile = "Y"
/* - debug - If set to "Y", will display the list of passed        */
/*           parameters and their values.                            */
debug = "N"
/* - tracev - If set to "Y", will display the executed clist code */
tracev = "N"
/*-----*/
CALL ON ERROR
ADDRESS ISPEXEC "CONTROL ERRORS RETURN"
TRACE (tracev)
sysicmd = SYSVAR("SYSICMD")
libdefs = 0
endnest = 0
PARSE VALUE "Browse Edit browsed edited" WITH be.B be.E bed.B bed.E

/*-----*/
/* Step 1: handle nesting requirements                                */
/* check that user defined object is not already nested              */
/*-----*/
IF( letnest = "N" )THEN DO
  ADDRESS ISPEXEC "TBCREATE" sysicmd "KEYS(USEROBJ) REPLACE NOWRITE"
  IF( RC = 4 )THEN
    CALL CTLEXIT 0, "Nested processing of" objname " not allowed."
  END
ELSE NOP

/*-----*/
/* Step 2: ensure clist is called with correct applid (ignored) */
/*-----*/

/*-----*/
/* LIBDEF/ALTLIB notes: Add LIBDEFS+ALTLIBS? (& set libdefs=1) */
/*-----*/

/*-----*/
/* Step 3: obtain -IPT- passed parameters                            */
/* Get the IPT parameters residing in the IPISHARE table              */
/* The following parameters will be available:                        */
/* default = requested command. (available from plist) (used)      */
/*      B   = request status info,                                   */
/*      V/E = view output                                           */
/*      D   = cancel                                                */
/* member = member name (ignored) */
/* ipidsn1 = name of the object: job name[(jobnumber)] (used) */
/* ipivol1 = volume (ignored) */
/* word1 = member pattern (ignored) */

```

```

/* word2   = significant portion of the pattern           (ignored) */
/* word3   = user specified password                     (ignored) */
/*-----*/
ADDRESS ISPEXEC "TBTOP IPISHARE"
ADDRESS ISPEXEC "TBSKIP IPISHARE"
IF( debug = "Y" )THEN DO
    SAY "IQIfOUT (REXX version)"
    SAY "  DEFAULT =" default
    SAY "  IPIDSN1 =" ipidsn1
    SAY "  IPIVOL1 =" ipivol1
    SAY "  MEMBER  =" member
    SAY "  WORD1   =" word1
    SAY "  WORD2   =" word2
    SAY "  WORD3   =" word3
    END
ELSE NOP

/*-----*/
/* Step 4: validate the passed object name and process    */
/*-----*/
CALL SUBOUT ipidsn1

/*-----*/
/* Step 5: signal OK to clear the nesting level if existing */
/*-----*/
endnest = 1

/*-----*/
/* Step 6: remove libdefs (if any established) would be moved into */
/* CTLEXIT if LIBDEFS were used but they are not.                */
/*-----*/

/*-----*/
/* Step 7: terminate processing                                */
/*-----*/
CALL CTLEXIT 0,"COMPLETED",ipidsn1 "procesing completed."
EXIT 99

/***** SubRoutines *****/

/*=====*/
/* terminate processing with code, short & long ISPF messages */
/*=====*/
CTLEXIT:
PARSE ARG cc,shortmsg,longmsg
IF( debug <> "N" )THEN DO
    SAY "RC="cc
    SAY "SHORTMSG="shortmsg
    SAY "LONGMSG="longmsg
    END
ELSE NOP

/*-----*/
/* clear any existing nesting level                            */
/*-----*/
IF( donenest = 1 ,
    & letnest = "N" ) THEN ADDRESS ISPEXEC "TBEND" sysicmd

```

```

/*-----*/
/* clear any existing LIBDEF/ALTLIBs */
/*-----*/

ADDRESS ISPEXEC "SETMSG MSG(IQIX101)"
EXIT (cc)
ERROR: RETURN

/*=====*/
/* control OUTput processing */
/*=====*/
SUBOUT:
ARG jobname

/*-----*/
/* commands available are only: V,D,B,E */
/*-----*/
IF( WORDPOS(default,"V D B E")=0 )THEN
    CALL CTLEXIT 8,"INVALID SELECTION",,
        'This OLIST entry can be selected with the "V", "D", "B",',
        'or "E" line commands.'
ELSE NOP

/*-----*/
/* check that only one job meets the requirements of job[(jes)] spec.*/
/*-----*/
IF( debug = "Y" )THEN "STATUS" jobname
ms = MSG("ON")
CALL OUTTRAP "S."
"STATUS" jobname
CALL OUTTRAP "OFF"
max4 = MIN(4,s.0)
CALL MSG ms
DO i = 1 TO max4
    IF( LEFT(s.i,3) = "IKJ" )THEN s.i = SUBWORD(s.i,2)
END
IF( s.0 > 1 )THEN DO
    ADDRESS ISPEXEC "CONTROL DISPLAY LINE START(00)"
    DO i = 1 TO max4
        SAY s.i
    END
    CALL CTLEXIT 8,"FULLY QUALIFY JOBNAME",,
        "A specific jobname(jobnumber) is required."
    END
ELSE NOP

/*-----*/
/* otherwise only one job meets the requirements of job[(jes)] spec. */
/*-----*/
IF( SUBWORD(s.1,3,2) = "NOT FOUND" )THEN
    CALL CTLEXIT 8,"JOB NOT FOUND",s.1
    jobjes = SPACE(TRANSLATE(WORD(s.1,2),". ","()"),0)

/*-----*/
/* commands available are only: V,D,B,E */
/*-----*/
SELECT
WHEN( default="V" )THEN
    /*-----*/
    /* show the status in message */
    /*-----*/

```

```

/*-----*/
CALL CTLEXIT 0,SUBWORD(s.1,3),s.1
WHEN( default="D" )THEN DO
/*-----*/
/* delete the job (if available) else issue message */
/*-----*/
cmd = "CANCEL" jobname "PURGE"
ms = MSG("ON")
CALL OUTTRAP "O."
cmd
CALL OUTTRAP "OFF"
CALL MSG ms
IF( o.0 = 0 )THEN
CALL CTLEXIT 0,"JOB DELETED",,
"Command:" cmd", ended successfully."
ELSE DO
oi = LEFT(cmd "ENDED RC="RC".",76)
DO i = 1 TO o.0
IF( LEFT(o.i,3) = "IKJ" )THEN o.i = SUBWORD(o.i,2)
o.i = oi LEFT(o.i,76)
END
CALL CTLEXIT 8,"COMMAND FAILED",oi
END
END
WHEN( default="B" ,
| default="E" )THEN DO
/*-----*/
/* browse or edit the listing (if available) else issue message */
/*-----*/
IF( SUBWORD(s.1,3,2) <> "ON OUTPUT" )THEN
CALL CTLEXIT 8,"NOT ON OUTPUT QUEUE",,
LEFT(s.1,76)"BROWSE requires the job to be on the OUTPUT queue."
ELSE NOP

/*-----*/
/* set dsname, delete existing and reallocate */
/*-----*/
dsn = USERID()).OUTLIST."jobjes
ms = MSG("OFF")
"DELETE '"dsn'"
CALL MSG ms
PARSE VALUE "?" WITH sysreason re
cmd = "ALLOC DD(SYSUT2) DS('"dsn"') NEW REUSE",
"LRECL(133) DSORG(PS) RECFM(F B A) SP(3,3)CYL"
cmd
IF( RC<>0 )THEN DO
IF( sysreason<>"?" )THEN re = "REASON="sysreason
CALL CTLEXIT 8,"ALLOCATION ERROR",,
LEFT(cmd,(LENGTH(cmd)+76)%76*76),
LEFT("ENDED RC="RC re,76),
"Cannot allocate work file for" jobname", try later?"
END
ELSE NOP

/*-----*/
/* put the output to the dataset if any found else issue message */
/*-----*/
ms = MSG("ON")
CALL OUTTRAP "J."
"OUTPUT" jobname "PRINT('"dsn"') BEGIN KEEP HOLD"

```

```

CALL OUTTRAP "OFF"
CALL MSG ms
IF( LEFT(j.1,3) = "IKJ" )THEN j.1 = SUBWORD(j.1,2)
IF( RIGHT(j.1,22) = "NO HELD OUTPUT FOR JOB" )THEN
    CALL CTLEXIT 8,"NO HELD OUTPUT",j.1

/*-----*/
/* browse or edit the listing (save RC) */
/*-----*/
IF( default = "B" )THEN
    ADDRESS ISPEXEC "BROWSE DATASET('"dsn"')"
ELSE
    ADDRESS ISPEXEC "EDIT DATASET('"dsn"')",
                    "MACRO(!IQIEMAC) PROFILE(IPIVIEW)"
berc = RC

/*-----*/
/* delete the dataset */
/*-----*/
ms = MSG("OFF")
IF( delfile = "Y" )THEN DO
    PARSE VALUE "?" WITH sysreason re
    "DELETE '"dsn'"
    IF( sysreason<>"?" )THEN re = "REASON="sysreason
    IF( RC=0 )THEN
        delmsg = "'dsn'" was deleted"
    ELSE
        delmsg = "but '"dsn'" delete failed RC="RC re
    END
ELSE NOP
CALL MSG ms
IF( berc>4 )THEN
    CALL CTLEXIT 8,"OUTPUT UNAVAILABLE",,
    "No output is available," be.default "failed, empty file."
ELSE
    CALL CTLEXIT 0,"OUTPUT" TRANSLATE(bed.default),,
    "Output was" bed.default "successfully"

END
END/*SELECT*/

RETURN

/*****
/* 5698-A81 (C) COPYRIGHT IBM CORP 1992,2006 */
/* ISPF PRODUCTIVITY TOOL: 5.9.0 */
*****/

```

---

### A.1.3 Running the wizard to activate the REXX output exec

The supplied wizard CLIST is called IQIWIZRD and is best run from a native TSO READY state. Figure A-8 on page 265, Figure A-9 on page 265, and Figure A-10 on page 265 show the navigation through the panels, which is usually just a case of changing the options you want and pressing Enter. The selection choices are varied by the selections that you make on the Option selection panel (PANELID=IQIIEWZ09), but follow the selections that we provide here for the purposes of customizing ISPF-PT so that you can use UDOs.

1. Type the command PANELID to see where you are in the dialogue. The only panels that you need to change are:
  - IQIIWZ03
  - IQIIWZ1C

```
SE16661.SPFLQG1.LIST has been kept.

READY
iqiwizrd
. . . %IQIWIZRD INVOKED, PLEASE WAIT *** _
```

Figure A-8 Native TSO at a READY prompt start IQIWIZRD

2. Press Enter.

```
. . .
*** _
```

Figure A-9 Native TSO

3. Press Enter.

```
PLEASE ENTER THE NAME OF THE -IPT- TABLE LIBRARY OR JUST PRESS ENTER TO EXIT.
EXAMPLE: IQI.SIQITLIB

==> ipt.test.tlib

PLEASE ENTER THE NAME OF THE -IPT- PANEL LIBRARY OR JUST PRESS ENTER TO EXIT.
EXAMPLE: IQI.SIQIPLIB

==> ipt.test.plib

*** _
```

Figure A-10 Native TSO

4. At this point, the ISPF dialogue starts. Select option 1, and press Enter to begin, as shown in Figure A-11.

**Note:** Figure A-11 on page 265 through Figure A-29 on page 274 involve observing the figure content and simply pressing Enter. Consequently, you are not alerted to the next displayed figure because of the serial sequence display.

```
IQIIWZ00 ----- IBMIPT - Customization Wizard -----
COMMAND ==> 1

Please select which tasks you want to perform:

1 Customization      - Set up IBMIPT options and defaults.
X EXIT               - Exit - do not customize, do not set up password.

      Make selection and press the Enter key or press the END key to exit.
```

Figure A-11 ISPF-PT customization wizard

5. Type PANELID, and then press Enter, which adds the Panel-IDs to the top-left corner of your window and assists in navigation. Select 1 for Customization, and press Enter. Figure A-12 is displayed.

```

IQI1WZ01 ----- IBMIPT - Customization Wizard -----
COMMAND ==> _

+-----+
| Welcome to the IBMIPT customization wizard. You will be presented with a |
| series of screens where you will fill in different customization options. |
|                                     |
| The selected options will determine which screens will be displayed, so |
| don't be surprised if you get different results when you run this utility |
| again with different selected options. |
|                                     |
| Some of the options you select will force a change on all IBMIPT users, |
| while others will only change their defaults. Those options that change |
| default values will only affect new IBMIPT users. Existing users have |
| their defaults already set in their ISPF profile. |
|                                     |
| Note that IBMIPT users can change their defaults via the SET command (by |
| entering "IPT SET" or shortcut "ISET" on any ISPF panel). |
+-----+

Press ENTER to proceed or the END key for the initial screen

```

Figure A-12 ISPF-PT customization wizard

6. Press Enter. Figure A-13 is displayed.

```

IQI1WZ09 ----- IBMIPT - Customization Wizard -----
COMMAND ==> _

Please select the options you would like to inspect or specify:
(some options are pre-selected by IBMIPT)

Select  Option                                     Notes
-----
S      DASD storage management                     Activate built-in support
S      Browse/View preferences                     Specify VIEW data integrity
S      Obtain info about IBMIPT objects             Recommended for new installation
_      Install IBMIPT session exits                 Specify CLIST/REXX/module names
_      Change names of general commands            Rename IBMIPT command names
_      Specify PDS compress options                 Compress PDS integrity options
_      Session diagnostics options                 Control PANELID and other options
S      User interface options                       Action bars, hotbars, etc.

Press ENTER to proceed or the END key for the initial screen

```

Figure A-13 ISPF-PT customization wizard

7. Select the preferences shown in Figure A-14 on page 267, and press Enter.



```

IQIIWZ05 ----- IBMIPT - Customization Wizard -----
COMMAND ==>

Verify Object List (OLIST) defaults (for first time IBMIPT users):

Default OLIST command          ==> B (B=Browse, E=Edit, V=View)
(When a line is selected by number, with the S line command, or with cursor)

Check edit recovery when list opens ==> N (Y=Yes, N=No)

Default cursor position        ==> M (M=Main command, L=Line command)

Show VOLSER of cataloged data-sets ==> Y (Y=Yes, N=No)

Catalog search pattern rules   ==> 1 (0, 1, 2)
(0) Unrestricted dataset name patterns
(1) First character may not be a "wild card"
(2) High level qualifier may not have any "wild card" characters

Press ENTER to proceed or the END key for the initial screen

```

Figure A-14 ISPF-PT customization wizard

8. Press Enter again. Figure A-15 is displayed.

```

IQIIWZ06 ----- IBMIPT - Customization Wizard -----
COMMAND ==>

IBMIPT searches for the OLIST (object list) library in the following order:
1. In the list of allocated files, for a DD name "PLSTLIBW" . If it is
   pre-allocated (before OLIST is invoked), IBMIPT will use it.
2. In the catalog. If a library by the name "qual1.PLIST.LIBRARY" is
   catalogued, IBMIPT will use it. "qual1" stands for the first level
   qualifier you specify on this screen.
3. In the list of allocated files, for a DD name "IPITBLIB" . If it is
   pre-allocated (before OLIST is invoked), IBMIPT will use it.
If IBMIPT cannot find the library pre-allocated or catalogued (as
explained above), it will try to allocate a new PDSE library, using the
name "qual1.PLIST.LIBRARY" with the UNIT name specified below.
Enter/Verify OLIST library defaults (as explained above):
PREFIX ("qual1")      ==> U (U=USERID, P=TSO PREFIX)
Allocation UNIT name ==> SYSALLDA (Unit name, blank for SYSALLDA,
                                or * for default TSO unit name)

IBMIPT also uses DD(IPITBLIB) for specifying a personal library as a
repository of persistent tables, such as CUT/PASTE clipboards and
the TSO-SHELL command lists. This library may be dynamically allocated
at IBMIPT startup whenever DD(IPITBLIB) is omitted from the logon procedure.
Elect persistent table library use: Y (yes), N (no) ==> Y

```

Figure A-15 ISPF-PT customization wizard

9. Press Enter. Figure A-16 on page 268 is displayed.

```

IQIIWZ02 ----- IBMIPT - Customization Wizard -----
COMMAND ===> _

  IBMIPT uses object-oriented technology to provide users with access to
  different classes of objects they commonly use. The most common object
  class is a data set. Within the DATA SET object class, IBMIPT supports
  several subclasses (for example sequential data sets, PDS data sets).

  Each object class is internally associated with a "METHOD" - an access
  mechanism that indicates how IBMIPT should handle each class. This is
  similar to the "OLE" concept in Microsoft's Windows on the PC.

  Some methods are provided internally by IBMIPT (for example the method
  that handles PDS libraries and display the MSL -the Member Selection
  List). Other methods are IBMIPT interfaces to third party products - for
  example the DB2-table class method requires a third party product that
  can browse or edit DB2 tables.

  The following screens you will select which object classes should be
  supported by IBMIPT, and which methods each supports.

  Press ENTER to proceed or the END key to return to the initial screen.

```

Figure A-16 ISPF-PT customization wizard

10. Press Enter.

11. Select **User defined objects**, which is located on the second row from the bottom, and press Enter, as shown in Figure A-17.

```

IQIIWZ03 ----- IBMIPT - Customization Wizard -----
COMMAND ===>

  Select the optional object classes to be supported. Internal
  object classes are already preselected. If you are unsure, select
  all options to get more information about each one. You will be
  able to de-select undesired option later.

  Select  Object Class                               Notes
  -----
  S       Sequential data sets                       Built-in IBMIPT support
  S       PDS and PDSE libraries                     Built-in IBMIPT support
  S       Open Edition files                         Requires OpenEdition MVS
  S       PC files                                   Requires ISPF workstation
  _       Panvalet libraries                         Interfaces to third party product
  _       Librarian files                           Interfaces to third party product
  _       PDSMAN libraries                          Activates built-in support
  _       VSAM data sets                             Interfaces to third party product
  _       DB2 tables                                Interfaces to third party product
  S_     User defined objects                        Define your own interface
  _       SCLM support                               Activates built-in support

  Press ENTER to proceed or the END key to return to the initial screen.

```

Figure A-17 ISPF-PT customization wizard

12. Press Enter. Figure A-18 on page 269 is displayed.

```

IQI IWZ1F ----- IBM IPT - Customization Wizard -----
COMMAND ==>
Specify DASD management products support:
Provide DFSMS support          ==> * (Y=Yes, N=No, *=Automatic detection)

Name of MIGRATION/ARCHIVAL system ==> HSM      (HSM, DMS, ASM2, ABR or BLANK)
Name of secondary ARCHIVAL system ==> _        (ABR or BLANK)

Name of MIGRATION/ARCHIVAL volume ==> MIGRAT (e.g. MIGRAT, ARCI VE)

Name of "DELETE migrated" command ==> HDELETE (e.g. HDELETE)
(Command name, CLIST, or REXX EXEC to be used instead of the DELETE command
when a user tries to delete a migrated/archived data set in OLIST).

Notes: - If you leave this field BLANK, IBM IPT will recall/restore the
        data set and then display a confirmation screen (that displays the
        data set contents if the DSORG is PO or PS).
        - If you do specify a command, CLIST or REXX EXEC, IBM IPT will
          invoke that command after receiving a confirmation on a panel
          that only lists the name of the data set (no contents displayed).
        -
Press ENTER to proceed or the END key to return to the initial screen.

```

Figure A-18 ISPF-PT customization wizard

13. Press Enter. Figure A-16 on page 268 is displayed.

```

IQI IWZ1V ----- IBM IPT - Customization Wizard -----
COMMAND ==>
IBM IPT VIEW FUNCTION CUSTOMIZATION:

Beginning with ISPF Version 4, IBM IPT provides two types of VIEW:
SHARED-VIEW and EXCLUSIVE-VIEW (Under versions prior to ISPF
Version 4, IBM IPT will always use EXCLUSIVE-VIEW).

EXCLUSIVE-VIEW is similar to EDIT - only one user can access the
file (member) at a time. Additionally, a confirmation screen will
be displayed before changes are saved back to the file.

SHARED-VIEW does not issue ENQ on the file, thus allowing others to
edit the file at the same time. Under SHARED-VIEW the save command
is disabled, but the REPLACE/CREATE commands can still be used.

Specify the type of VIEW for your installation:
VIEW TYPE ==> E (E=Exclusive, S=Shared)

Should users be allowed to override the above setting?
USER CAN OVERRIDE THE VIEW TYPE ==> Y (N=No, Y=Yes-Override is allowed)

Press ENTER to proceed or the END key to return to the initial screen.

```

Figure A-19 ISPF-PT customization wizard

14. Press Enter. Figure A-20 on page 270 is displayed.

```

IQIIWZ10 ----- IBMIPT - Customization Wizard -----
COMMAND ==>
DEFAULT PROCESS FOR MAIN PANEL OPTION 1 (BROWSE, VIEW)

Beginning with ISPF Version 4, the program used to invoke BROWSE on the main
menu can have two modes - BROWSE or VIEW. The program ("ISRBRO") is
invoked by a statement in the ")PROC" section of the panel: "PGM(ISRBRO)"

When PGM(ISRBRO) is invoked, IBMIPT displays an entry screen that
includes a "DEFAULT ACTION ==>" field which provides the ability
to set the action to B rowse, V iew, or E dit.

The "DEFAULT ACTION" settings on this screen determines how IBMIPT
handle browse requests. This setting is easily overridden by users by
the SET command which can be issued on the same panel.

The advantages of VIEW are the ability to change the file, to use CUT
and PASTE, and to exclude lines. The disadvantage is that VIEW
pre-loads the entire file and uses more resources.

Specify the default action when PGM(ISRBRO) is invoked:
DEFAULT ACTION ==> B (B=Browse, V=View)

Press ENTER to proceed or the END key to return to the initial screen.

```

Figure A-20 ISPF-PT customization wizard

15. Press Enter. Figure A-21 is displayed.

```

IQIIWZ22 ----- IBMIPT - Customization Wizard -----
COMMAND ==>

Activate HOTBARS (on IBMIPT panels) ==> Y (Cursor sensitive command fields)
Globally activate action bars      ==> Y (CUA style pulldown menus)
If action bars are globally active:
  Use action bars in edit/BROWSE   ==> Y (Note: action bar takes 2 panel lines)
  Use action bars in MSL           ==> Y (IBMIPT's Member Selection List)
  Use action bars in OLIST         ==> Y (IBMIPT's Object list)
  Use action bars in DSLIST        ==> Y (ISPF option 3.4)
Notes 1. Users can change these defaults. The recommended method of changing
       these options is by entering the IPT SET command on the main menu.
       2. Pre-processing ISPF panels will prevent IBMIPT from changing these
          options on ISPF panels. IBMIPT panels should never be pre-processed.
Press ENTER to proceed or the END key to return to the initial screen.

```

Figure A-21 ISPF-PT customization wizard

16. Press Enter. Figure A-22 on page 271 is displayed.

```

IQIIWZ2A ----- IBMIPT - Customization Wizard -----
COMMAND ==>

Specify the default for BROWSE/EDIT/VIEW main menu display of
reserved name @H :

    Display OLIST of History-List ==> Y (Y=Yes, N=No)

Specify the default User Interface options for CUT and PASTE
command handling :

    Cut and Paste Options ==> 1    (1, 2)

    (1) IBMIPT CUT/PASTE      CUT/PASTE commands are IBMIPT's for all
                               applications (ISPF, SDSF, OMVS, etc.)
    (2) ISPF CUT/PASTE        CUT/PASTE commands are ISPF's for all
                               applications

```

Figure A-22 ISPF-PT customization wizard

17. Press Enter. Figure A-23 is displayed.

```

IQIIWZ0A ----- IBMIPT - Customization Wizard -----
COMMAND ==>

MSL (Member Selection List) is the IBMIPT built-in method of displaying
member lists in ISPF. MSL provides stacking of multiple commands,
filtering of the member list, global actions (global find, change,
edit), member-preview windows, and other powerful functions.

Specify below the following MSL defaults:

    Automatic preview          ==> Y (N=No, Y=Yes: Display top of member in a
                                   window following the Locate,
                                   Find, or FindText commands)

Notes: -These defaults will affect users who have not yet used IBMIPT.
        -Users can change their defaults via the SET command (in MSL).
Press ENTER to proceed or the END key to return to the initial screen.

```

Figure A-23 ISPF-PT customization wizard

18. Press Enter. Figure A-24 on page 272 is displayed.

```

IQIIWZ07 ----- IBMIPT - Customization Wizard -----
COMMAND ==> _

      IBMIPT has a built-in printing engine that directs prinouts to a
      SYSOUT file without the intermediate ISPF LIST data set.

      Users can direct their IBMIPT print requests to the ISPF
      list data set, or use IBMIPT's faster method.

      When the IBMIPT method is selected, IBMIPT dynamically allocates
      a SYSOUT file using the specifications that reside in each
      user's profile. On the following screen you will specify the
      defaults for users who have not yet run under IBMIPT. Users who
      have already run under IBMIPT have their defaults saved in their
      ISPF profile.

      To change the defaults, users can use the SET PRINT command (in
      MSL or OLIST).

Press ENTER to proceed or the END key to return to the initial screen.

```

Figure A-24 ISPF-PT customization wizard

19. Press Enter. Figure A-25 is displayed.

```

IQIIWZT4 ----- IBMIPT - Customization Wizard -----
COMMAND ==>

In case IBMIPT print requests directed to the IBMIPT print engine rather than
the ISPF LIST dataset, specify the SYSOUT allocation defaults (for users who
have not yet used IBMIPT).
IBMPT users can change these options via the SET command.

Enter/Verify SYSOUT attributes:
  CLASS           ==> A           (Sysout class)
  DESTINATION ID  ==>             (or remote station id)
  Lines per page  ==> 60          (page size)
  Keep in HOLD queue ==> N       (Y=Yes, N=No)
  FORM number     ==>
  FCB name        ==>
  WRITER name     ==>           (Output WRITER)

Press ENTER to proceed or the END key to return to the initial screen.

```

Figure A-25 ISPF-PT customization wizard

20. Press Enter. Figure A-26 on page 273 is displayed.

```

IQIIWZ1G ----- IBMIPT - Customization Wizard -----
COMMAND ==>

IBMIPT provides an alternative TSO command shell that provides
command history, permanent commands entries, execution of ISPF dialog
manager statements, and substitution of data sets pointed by the
cursor (create your own point-and-shoot commands).
For example, if a permanent entry number 5 is defined as:
    %TRANSFER SYSB.JONES DSN(/)
then by entering "TSO 5" on any ISPF screen while placing the cursor
on a data set name appearing on that screen would execute the
%TRANSFER command on the pointed data set.

The SET command is used to activate or deactivate the IBMIPT TSO command shell.

For users who have not yet used IBMIPT, indicate the desired defaults:
Use IBMIPT shell      ==> Y  (Y=Yes - Use the IBMIPT TSO shell, N=No)
Output line number   ==> 7  (Number of screen line for TSO command output)
Initial screen       ==> H  (H=History list, P=Permanent list)
Automatic filling     ==> Y  (Y=Yes - Automatically fill permanent list, N=No)
History recording     ==> L  (F=Full - Every command put in history list)
                        (L=Limited - Build a table of commands that are
                        not put in the history list. Use this option
                        to hide security-sensitive commands).

```

Figure A-26 ISPF-PT customization wizard

21. Press Enter. Figure A-27 is displayed.

```

IQIIWZ1H ----- IBMIPT - Customization Wizard -----
COMMAND ==>
Specify DSLIST (option 3.4) parameters:

IBMIPT adds the ability to invoke OLIST (Object/dataset list) directly
from the DSLIST panel. These object lists are faster and more powerful
than the ISPF Data set list (DSLISL).

The following option specifies the DEFAULT action taken on the DSLIST panel
when the user presses the ENTER key with a blank "OPTION ==>" field:

Default action ==> TP (DS=DSLISL, PL=Permanent List, TP=Temporary List)

Notes: - Regardless of the selected default, users can select any other
        option by explicitly entering one of the other DSLIST
        panel options (like DS, V, PV, etc).
        - The default action may also be set by each user via the
          SET command on the DSLIST panel.

Press ENTER to proceed or the END key to return to the initial screen.

```

Figure A-27 ISPF-PT customization wizard

22. Press Enter. Figure A-28 on page 274 is displayed.

```

IQI IWZ1C ----- IBM IPT - Customization Wizard -----
COMMAND ==>

You have indicated that you want to define user objects.

IBM IPT will assume that a data set name (in BROWSE, EDIT, VIEW, and OLIST)
that starts with the greater-than sign (>) is a user object, and pass
control to a CLIST (or REXX EXEC) to handle the object. A skeleton CLIST,
indicating the available parameters, resides in the IBM IPT CLIST library
under the name IQI$USER.

Specify below the name of the user object handler:

CLIST/REXX EXEC name    ==> IQIRUCTL _ (Leave BLANK for no user-object support)

NOTE: A sample object handler called IQI$OUT is in the IBM IPT CLIST library.
Press ENTER to proceed or the END key to return to the initial screen.

```

Figure A-28 ISPF-PT customization wizard

23. Enter the name of the user object handler as IQIRUCTL (or tailor the name and sample to suit your own needs), and press Enter. Figure A-29 is displayed.

```

IQI IWZ95 ----- IBM IPT - Customization Wizard -----
COMMAND ==> _

The customization wizard has completed the first phase of interviewing you for
the installation options.

You can now proceed to finalize the customization process (press the ENTER key)
or you may cancel (press the END key) and redisplay the initial screen.

Press ENTER to proceed or the END key to return to the initial screen.

```

Figure A-29 ISPF-PT customization wizard

24. Press Enter. Figure A-30 on page 275 is displayed.



```

IQITSVL ----- TSO COMMAND SHELL - RESTRICT HISTORY RECORD Row 1 to 2 of 2
COMMAND      ==> _                                SCROLL ==> CSR
Main commands: CANCEL, DOWN, END, UP
Line commands: I=insert, D=delete

+-----+
| TSO commands (or CLIST or REXX EXECs) specified here with the |
| "No History" set to Y (YES) will not be recorded in the command |
| history list of the TSO Command Shell (option 6).              |
+-----+

Line      Command No
COMMAND   or CLIST History
-----
-         PASSWORD Y
-         PROTECT  Y
-----

```

Figure A-30 TSO Command Shell

25. Press End. Depending on your ISPF LOG/LIST settings, you may get Figure A-31 might be displayed.

```

ISPPFT01          Specify Disposition of Log Data Set
Command ==> _____

Log Data Set (SE16661.SPFLOG1.LIST) Disposition:
Process Option . . . . 3  1. Print data set and delete
                        2. Delete data set without printing
                        3. Keep data set - Same
                           (allocate same data set in next session)
                        4. Keep data set - New
                           (allocate new data set in next session)

Batch SYSOUT class . . . A
Local printer ID or
writer-name . . . . .
Local SYSOUT class . . .

List Data Set Options not available

Press ENTER key to complete ISPF termination.
Enter END command to return to the primary option menu.

Job statement information: (Required for system printer)
==> //SE16661N JOB (DUMMY), 'LOG/LIST', CLASS=A, MSGCLASS=Z
==>
==>
==>

INSTALLATION DEFAULTS WERE SAVED

```

Figure A-31 Log data set

26. Press Enter to Exit to ISPF. Here you can type in ISPF and continue processing or LOGOFF, as shown in Figure A-32.

```

SE16661.SPFLOG1.LIST has been kept.

READY
LOGOFF_

```

Figure A-32 Logoff

## A.1.4 Extending the exec into a general purpose user defined object

There is no great advantage in using the UDO only to process OUTPUT or some other single function. What is really useful is to be able to have a number of different UDOs that are all controlled by one master UDO, a generalized UDO. That way we can plug-n-play with individual UDOs as we want. In this section, we suggest the concept because this is going outside of the scope of this book.

## A.1.5 Summarizing how the process for those impatient to just-do-it!

1. Customize the LOGON procedure you are using to include your new test libraries: CLIB, PLIB, TLIB. You will need the assistance of a systems programmer, if you are not one yourself.
2. Copy the code in “additional materials” for the REXX execs: IQIRUSER and IQIROUT into your test CLIB.
3. Run the ISPF-PT set up wizard (in native TSO): IQIWIZRD, requesting “user defined objects” and setting the name of the UDO controller IQIROUT as the UDO name. After you are inside the ISPF dialogue, enter the command PANELID to help you navigate. The only two panels that you need to change are IQIIWZ03 and IQIIWZ1C.
4. Create your own UDO execs based on the IQIROUT or basic IQIRUSER samples.

## A.2 Experimenting with other UDOs

There are two additional UDO subcommands to cover, as shown in Figure A-33. The first IQIRDUMY is just a “dummy” for testing. It merely writes out the debugging information. The last IQIRERRO ID is present because there is no IQIRERRO UDO, and this is just to demonstrate the result of having a UDO reference in UDOCMDS, which does not actually exist.

```
File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (B) ----- USER DEFINED OBJECTS ----- Row 11 from 14
Command   ===>                                SCROLL ===> PAGE
Hotbar: FLIP   CLRVOL  FILLVOL  REFRESH  UTIL      CUT      SET      UPDATE
Open list ===> IPTRED  (or BLANK for reference list)
TSO PARMS ===>
Command  Member  Numbr Data Set Names / Objects          *EXCLUDE* Volume
-----
-V      11  >OUT SE16661J
-V      12  >DUM something
-V      13  >ERR anything
----- END OF LIST -----
```

Figure A-33 Dummy and error UDOs

### A.2.1 IQIRDUMY

Selecting line 12 from Figure A-33 produces diagnostics, as shown in Example A-6 on page 277. The diagnostics are turned off and on from the setting of the debug variable within the UDO subcommand exec.

Example: A-6 Dummy UDO display diagnostics

```
SOURCE = TSO COMMAND IQIRDUMY SYSPROC ? ? TSO ISPF ?  
DEFAULT = V  
IPIDSN1 = DUM something  
IPIVOL1 =  
MEMBER =  
WORD1 =  
WORD2 =  
WORD3 =  
***
```

## A.2.2 IQIRERRO

Selecting IQIRERRO just displays the message shown in Figure A-34.

```
File Edit Find Display Populate Settings Menu Util Test Help Exit  
-----  
-IPT- OLIST (B) ----- USER DEFINED OBJECTS ----- Row 11 from 14  
Command ===> SCROLL ==> PAGE  
Hotbar: FLIP CLRVOL FILLVOL REFRESH UTIL CUT SET UPDATE  
Open list ==> IPTRED (or BLANK for reference list)  
TSO PARMS ===>  
Command Member Numbr Data Set Names / Objects *EXCLUDE* Volume  
-----  
11 >OUT SE16661J  
12 >DUM something  
13 >ERR anything  
----- END OF LIST -----  
  
-V  
V  
  
IKJ56500I COMMAND IQIRERRO NOT FOUND  
***
```

Figure A-34 Error UDOs – does not exist, displays an error message

## A.3 More meaningful UDOs

The UDO subcommand was developed to help with the development of ISPF panels. Quite often, when developing an ISPF application, it is necessary to check the panel <sup>1</sup>. The UDO allows the normal functions, such as Browse, Edit, and View, but also allows the user to Display or Select a panel. It can work on a library, a member of a library, or a member mask with a library.

<sup>1</sup> Standard ISPF option 7 supplies sub-options for this already, but this is just to demonstrate the power of UDOs in OLISTs.



Selecting the DISPLAY option also displays the panel, as in Figure A-37.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (B) ----- USER DEFINED OBJECTS ----- Row 11 to 14 of 14
Command   ==>
Hotbar: FLIP   CLRVOL  FILLVOL  REFRESH  UTIL    CUT    SET    UPDATE
Open list ==> IPTRED  (or BLANK for reference list)
TSO PARMS ==>
Command  Member  Numbr Data Set Names / Objects                               Volume
-----
- V              11 >OUT SE16661J
- V              12 >DUM something
- V              13 >ERR anything
DIS _          IQIDIAG  14 >PAN 'IPT.TEST.PLIB'
-----
                                END OF LIST -----

```

Figure A-37 IQIRPANL UDO – developing ISPF panels for DISPLAY or SELECT services – Display panel

Figure A-38 shows the error diagnostics panel.

```

-IPT----- ERROR DIAGNOSTICS -----
COMMAND ==> _
* * * * *
*
*
*
* The following message was displayed:
*
*
*
*
* PROGRAM ISSUING MESSAGE:          PROCESS ISSUING MESSAGE:
* Last available ISPF error message:
*
*
* ----- ENVIRONMENT INFORMATION -----
* USERID=SE16661,LOGON PROC=SYSUSER,TSO PREFIX=SE16661,APPLID=ISR
* SYSID:ZT01,PDF 5.8 ISPF 5.8MVS      TSO LANGUAGE:ENGLISH
* TERMINAL:3278,PFKEYS:12,DIMENSIONS: (0032 x 0080) (0032 x 0080) GR=YES/YES
* COLORS:0007,SPLIT:NO/1,CMD:ASIS,  TERMCID:01140,TERMCP=1140,TERMCS=0695
* INVOCATION:PDF      PGM(IQIITH ) NEWAPPL(ISR) PARM(PANEL(ISR@390) )+
*
* * * * * Press ENTER to continue * * * * *

```

Figure A-38 Error diagnostics

If we try and SELECT the panel, although it will display exactly as Figure A-38, we will get an error because this panel is not a valid selection panel. It is only displayed as shown in Figure A-39 on page 280. The short message is displayed, and pressing the HELP key adds the long message.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (B) ----- USER DEFINED OBJECTS ----- Invalid panel
Command ===>
Hotbar: FLIP      CLRVOL  FILLVOL  REFRESH  UTIL    CUT      SET      UPDATE
Open list ===> IPTRED   (or BLANK for reference list)
TSO PARMS ===>
Command  Member    Numbr Data Set Names / Objects                               Volume
-----
-11 >OUT SE16661J
-12 >DUM something
-13 >ERR anything
-14 >PAN 'IPT.TEST.PLIB'
----- END OF LIST -----

IQIX101 'IQIDIAG' is not a valid 'selection' panel.

```

Figure A-39 IQIRPANL UDO – developing ISPF panels for DISPLAY or SELECT services – Select panel

The supplied exec IQIRPANL probably still has much scope for improvement. It was developed hastily in the short time scale of this book. You are free to improve on it, and use it as the base for other UDO subcommands. Example A-7 contains the source code for it, followed by the EditMacro that it invokes in VIEW to handle the PANEL DISPLAY/SELECTs.

Example: A-7 IQIRPANL UDO - developing ISPF panels for DISPLAY or SELECT services - Select panel

```

/**REXX(IQIRPANL)*****
**/
/* This UDO subcommand processor processes ISPF panels
**/
/* It takes an action verb of PANEL and can handle libraries or
**/
/* members of libraries or member masks. */
/**REXX(IQIRPANL)*****
**/
PARSE SOURCE srce
debug = "Y"
tracev = "N"
ADDRESS ISPEXEC "CONTROL ERRORS RETURN"
ADDRESS ISPEXEC "TBTOP IPISHARE"
ADDRESS ISPEXEC "TBSKIP IPISHARE"
IF( debug = "Y" )THEN DO
    SAY "SOURCE =" srce
    SAY "DEFAULT =" default
    SAY "IPIDSN1 =" ipidsn1
    SAY "IPIVOL1 =" ipivol1
    SAY "MEMBER =" member
    SAY "WORD1 =" word1
    SAY "WORD2 =" word2
    SAY "WORD3 =" word3

```

```

END
ELSE NOP
TRACE (tracev)
PARSE VAR ipidsn1 actverb lib .
IF( ABBREV("PANEL",actverb,3)=0 )THEN
    CALL CTLEXIT 0,"ACTION VERB ERROR",,
        "Action verb" actverb "does not match action PANEL."
qe = LISTDSI(lib "DIRECTORY RECALL")
IF( qe>0 )THEN
    CALL CTLEXIT 0,"ACCESS ERROR",LEFT(sysmsglvl1,77) sysmsglvl2
IF( sysdsorg<>"PO" )THEN
    CALL CTLEXIT 0,"INVALID DSORG",,
        "Only DSORG=PO libraries are supported by this UDO."
CALL PANEL lib

/*-----
-*/
/* terminate processing */
/*-----
-*/
CALL CTLEXIT 0,"COMPLETED",ipidsn1 "processing completed."
EXIT 99

/***** SubRoutines
*****/

/*=====
=*/
/* panel display processing */
/*=====
=*/
PANEL:
TRACE "C"
opts = "0 BROWSE EDIT VIEW DISPLAY SELECT"
udocomms = WORD(opts,POS(LEFT(default,1),"0BEVDS"))
IF( udocomms = 0 )THEN
    CALL CTLEXIT 0,"INVALID SELECTION",,
        "This OLIST entry can be selected with the:",
        "DISPLAY SHOW BROWSE EDIT & VIEW",
        "line commands."
ELSE NOP
qe = LISTDSI(lib)
oq = LEFT(lib,1)
ll = LENGTH(lib)
mbr = SPACE(member word1)
IF( mbr= "" ),
THEN eds = lib
ELSE
    IF( oq="" )
    THEN eds = LEFT(lib,ll-1)("mbr")
    ELSE eds = lib("mbr")
SELECT
WHEN( udocomms = "BROWSE" ,

```

```

        | udocomms = "EDIT" ,
        | udocomms = "VIEW" )THEN DO
ADDRESS ISPEXEC udocomms "DATASET("eds")"
IF( RC>8 )THEN
    CALL CTLEXIT 0,zerrsm,zerrlm
END
WHEN( udocomms = "DISPLAY",
    | udocomms = "SELECT" )THEN DO
ADDRESS ISPEXEC "VPUT (UDOCOMMS)"
ADDRESS ISPEXEC "LIBDEF ISPPLIB DATASET ID("lib") STACK"
ADDRESS ISPEXEC "VIEW DATASET("eds") MACRO(IQIRPANE)"
ADDRESS ISPEXEC "LIBDEF ISPPLIB"
ADDRESS ISPEXEC "VGET (UDOCOMMS)"
IF( udocomms<>" " )THEN DO
    PARSE VAR udocomms zerrsm;"zerrlm
    CALL CTLEXIT 0,zerrsm,zerrlm
END
ELSE NOP
END
END/*SELECT*/
RETURN

/*=====
=*/
/* terminate processing with code, short & long ISPF messages
*/
/*=====
=*/
CTLEXIT:
PARSE ARG cc,shortmsg,longmsg
IF( debug <> "N" )THEN DO
    SAY "RC="cc
    SAY "SHORTMSG="shortmsg
    SAY "LONGMSG="longmsg
END
ELSE NOP

/*-----
-*/
/* clear any existing nesting level */
/*-----
-*/
IF( donenest = 1 ,
    & letnest = "N" ) THEN ADDRESS ISPEXEC "TBEND" sysicmd

/*-----
-*/
/* clear any existing LIBDEF/ALTLIBS */
/*-----
-*/

ADDRESS ISPEXEC "SETMSG MSG(IQIX101)"
EXIT (cc)
ERROR: RETURN

```

---



Archived

Archived



## Invoking user commands from an Object List

In this section, you will learn how to invoke CLIST and REXX commands from an Object List. We demonstrate the OLIST user commands with XMIT and RECEIVE TSO commands.

## B.1 Example 1: transmitting and receiving files, host-to-host

Using ISPF or ISPF-PT option 6, you can use the XMIT command to transmit a file, as shown below:

```
XMIT HOST344.MYID DA('MYID.PDS')
```

You can also use ISPF option 3.4, overtyping the file name, as shown below:

```
XMIT HOST344.MYID DA(/)
```

You can receive the file with the TSO RECEIVE command.

You might have to respond to questions, such as overwriting an existing file.

When you use the ISPF productivity Facility, you can:

- ▶ Enter the ?XMIT command in the Object List
- ▶ Enter the DSLIST command in the Object List
- ▶ Invoke a User Command from the Object List

### B.1.1 Using the ?XMIT command with an Object List

You can prefix any command in an Object List display with a question mark (?), which invokes a pop-up window, where you can update the parameters prior to invocation. In Figure B-1, we typed the ?XMIT OLIST line command.

File	Edit	Find	Display	Populate	Settings	Menu	Util	Test	Help	Exit
-----										
-IPT- OLIST (E) ----- LEVEL DNET424.ADLAB ----- "A" will display assist										
Command ===>								SCROLL ===> CSR		
Hotbar: OPRINT REFRESH CLRVOL FILLVOL UPDATE CUT								FLIP VALIDATE		
								*TEMPORARY LIST*		
-----										
Command	Member	Numbr	Data Set Names / Objects							Volume
-----										
?xmit		1	'DNET424.ADLAB.ADATA'							DMPU20
		2	'DNET424.ADLAB.CNTL'							DMPU29
		3	'DNET424.ADLAB.COPYLIB'							DMPU15
		4	'DNET424.ADLAB.COPYLIBI'							DMPU23
		5	'DNET424.ADLAB.CUST.F4000'							DMPU22
		6	'DNET424.ADLAB.CUST.F4000.SPAC'							DMPU06
		7	'DNET424.ADLAB.CUST.ONEREC'							DMPU18
		8	'DNET424.ADLAB.CUSTFILE'							DMPU19
		9	'DNET424.ADLAB.CUSTFILE.KSDS'							
		10	'DNET424.ADLAB.CUSTFILE.KSDS.DATA'							DMPU27
		11	'DNET424.ADLAB.CUSTFILE.KSDS.INDEX'							DMPU27
		12	'DNET424.ADLAB.CUST1'							DMPU25
		13	'DNET424.ADLAB.CUST2.KSDS'							
-----										
F1=HELP	F2=SPLIT	F3=END	F4=IPT View	F5=RFIND	F6=RCHANGE					
F7=UP	F8=DOWN	F9=SWAP	F10=LEFT	F11=RIGHT	F12=RETRIEVE					

Figure B-1 ?XMIT command

**Note:** If you enter the command SHOWCMD ON, the pop-up panel in Figure B-2 on page 287 is invoked for all OLIST commands. Enter the A SHOWCMD command for assistance with the SHOWCMD command.

```

-IPT----- OLIST SHOWCMD PANEL -----
COMMAND ==>

Verify or change the TSO or CLIST command:

    ==> XMIT 'DNET424.ADLAB.ADATA'

Press ENTER to proceed or the END key to cancel the command.

-

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

```

Figure B-2 XMIT pop-up panel

The ISPF Productivity Tool appends the data set name to the command, as shown in Figure B-2.

**Note:** TSO commands, such as HRECALL, require the data set name as a single positional parameter. You can enter these commands on the OLIST window, without any prompts.

Overtyping the XMIT command with the correct parameters, as shown Figure B-3.

```

-IPT----- OLIST SHOWCMD PANEL -----
COMMAND ==>

Verify or change the TSO or CLIST command:

    ==> XMIT demomvs.dnet424 da('DNET424.ADLAB.ADATA')

Press ENTER to proceed or the END key to cancel the command.
                                IEBCOPY MESSAGES AND CONTROL STATEMENT
S                                PAGE      1
IEB1135I IEBCOPY  FMID HDZ1180  SERVICE LEVEL UA27650  DATED 20060711 DFSMS 01.
08.00 z/OS      01.08.00 HBB7730  CPU 2094
IEB1035I DNET424  SPIFFY      SPIFFY   09:13:23 TUE 25 SEP 2007 PARM=''
        COPY OUTDD=SYS00176,INDD=([SYS00172,R])
IEB1013I COPYING FROM PDSE  INDD=SYS00172 VOL=DMPU20 DSN=DNET424.ADLAB.ADATA
IEB1014I          TO PDSU OUTDD=SYS00176 VOL=DMPW02 DSN=SYS07268.T091323.RA000
.DNET424.R0109387
IGW01551I MEMBER ASAM1  HAS BEEN UNLOADED
IGW01551I MEMBER SUBXMP HAS BEEN UNLOADED
IGW01550I 2 OF 2  MEMBERS WERE UNLOADED
IEB147I  END OF JOB - 0 WAS HIGHEST SEVERITY CODE
INMX000I 0 message and 23 data records sent as 4275 records to DEMOMVS.DNET424
***

```

Figure B-3 XMIT pop-up window with additional parameters

Using the ?cmd gives you a way to see the data set parameter and to construct the valid syntax of the TSO command without overtyping the actual data set name, as you would using ISPF 3.4.

## B.1.2 Using the DSLIST command with an Object List

You can invoke the standard ISPF 3.4 window within ISPF-PT by using the DS command on the Data Set List window, as shown in Figure B-4.

```
Menu Reflist Refmode Special-lists Utilities Settings Test Help Exit
-IPT- ----- Data Set List Utility -----
Command ==> ds
DS - Display dataset list                P - Print data set list
blank - Temporary Object List           PV - Print VTOC information
PL - Permanent Object List              V - Display VTOC information
GDG - Display Generation Datasets       XV - Extended VTOC & space summary
More: +
Enter an option or select it by placing cursor on the option code
Specify parameters below:
Object List ==> (* for selection list, = for LAB)
Dsname Level ==> DNET424.ADLAB More? ==> N
Volume serial ==> (Leave BLANK for catalog scan,
volser or pattern for VTOC scan)
Data set list options:
Initial View ==> 1 : 1. Volume 2. Space 3. Attrib 4. Total
Enter "/" to select option: / Confirm Data Set Delete
                          / Additional Dataset Qualifiers
                          / Display Catalog Name
When the data set list is displayed, enter the "/" line command for a list of
F1=Help F2=Split F3=Exit F7=Backward F8=Forward F9=Swap
F10=Actions F12=Cancel
```

Figure B-4 Data Set List Utility – DS command

You can also invoke the ISPF 3.4 window with the DSLIST command, as shown in Figure B-5.

```
File Edit Find Display Populate Settings Menu Util Test Help Exit
-IPT- OLIST (E) ----- LEVEL DNET424.ADLAB ----- "A" will display assist
Command ==> SCROLL ==> CSR
Hotbar: OPRINT REFRESH CLRVOL FILLVOL UPDATE CUT FLIP VALIDATE
*TEMPORARY LIST*
TSO PARMS ==>
Command Member Numbr Data Set Names / Objects Volume
-----
DSLIST - 1 'DNET424.ADLAB.ADATA' DMPU20
2 'DNET424.ADLAB.ADATA.BIN' DMPU26
3 'DNET424.ADLAB.CNTL' DMPU29
4 'DNET424.ADLAB.CNTL.BIN' DMPU24
5 'DNET424.ADLAB.COBOL' DMPU26
6 'DNET424.ADLAB.COBOL.BIN' DMPU29
7 'DNET424.ADLAB.COPYLIB' DMPU15
8 'DNET424.ADLAB.COPYLIB.BIN' DMPU14
9 'DNET424.ADLAB.COPYLIBI' DMPU23
10 'DNET424.ADLAB.COPYLIBI.BIN' DMPU13
11 'DNET424.ADLAB.CUST.F4000' DMPU22
12 'DNET424.ADLAB.CUST.ONEREC' DMPU18
13 'DNET424.ADLAB.CUSTFILE' DMPU19
F1=HELP F2=SPLIT F3=END F4=IPT View F5=RFIND F6=RCHANGE
F7=UP F8=DOWN F9=SWAP F10=LEFT F11=RIGHT F12=RETRIEVE
```

Figure B-5 ISPF-PT DSLIST line command

Either option provides the ISPF 3.4 window, where you can overtype the data with an XMIT command.

### B.1.3 Invoking a User Command CLIST or REXX program from the Object List

You can also use a User Command to transmit files from an Object List. This method requires a z/OS CLIST or REXX routine. In this section, we demonstrate both methods.

1. First, turn on the Command Parameter by using the following command:

```
CMD ON
```

This command adds the following line to the Object List window:

```
TSO PARMS ==>
```

2. Next, you need a REXX or CLIST library. If you prefer to use a REXX program, allocate a PDS or PDSE as Variable Blocked, 255 bytes. If you prefer to use a CLIST, allocate a library as Fixed Block, 80 bytes.
3. Make the REXX or CLIST routine available to your ISPF session. You can do this one of two ways:
  - Use the TSO ALTLIB command to dynamically allocate the REXX or CLIST library.
  - Add a DD statement to your LOGON procedure.

Most people do not update access to modify logon procedures. The examples below show how to allocate a REXX or CLIST library dynamically. The following scenarios use the library name DNET424.CLIST for the CLIST data set and DNET424.EXEC for the REXX library name.

To allocate a CLIST library, use option ISPF-PT option 6:

```
ALTLIB ACT DA('DNET424.CLIST') APPL(CLIST)
```

To allocate a REXX library, use ISPF-PT option 6:

```
ALTLIB ACT DA('DNET424.EXEC') APPL(EXEC)
```

You must issue the ALTLIB command every time you sign on to TSO.

**Suggestion:** Use the **/EP** command under the ISPF-PT Command Shell, and add the ALTLIB command to one of the lines in the TSO Command Shell permanent list. Then you can type **TSO n**, to allocate the appropriate library. See the z/OS TSO/E Command Reference for details on the ALTLIB command.

**Note:** ALTLIB is a TSO that dynamically allocates CLIST or REXX libraries to your TSO session. For Panels, Messages, Tables, Skeletons, File tailoring output, User link libraries and Images, use the LIBDEF ISPF command to dynamically allocate application-level libraries. See the *ISPF Services Guide* for details on the LIBDEF command.

### B.1.4 Using a CLIST as a User Command

In this section, we discuss using a CLIST as a User Command.

**Note:** For information about CLIST, see the z/OS TSO/E CLIST publication.

Figure B-6 on page 290 displays the CLIST member XMC.

```

-IPT- EDIT DNET424.CLIST(XMC) - 01.08
Command ==>
***** ***** Top of Data *****
000001 PROC 2 DSN=TO
000002 XMIT (&TO) DATASET(&DSNAME)
***** ***** Bottom of Data *****

```

Figure B-6 CLIST member

This CLIST consists of two lines. The first line indicates that two parameters are required, named DSN= and TO. The second line performs the XMIT statement. ISPF-PT provides the first argument, DSN=, automatically. Enter the second argument in the TSO PARMS field, as shown in Figure B-7.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (E) ----- LEVEL DNET424.ADLAB ----- "A" will display assist
Command ==>
Hotbar: OPRINT REFRESH CLRVOL FILLVOL UPDATE CUT FLIP VALIDATE
*TEMPORARY LIST*
TSO PARMS ==> DEMOMVS.DNET424
Command Member Numbr Data Set Names / Objects Volume
-----
XMC -
1 'DNET424.ADLAB.ADATA' DMPU20
2 'DNET424.ADLAB.ADATA.BIN' DMPU26
3 'DNET424.ADLAB.CNTL' DMPU29
4 'DNET424.ADLAB.CNTL.BIN' DMPU24
5 'DNET424.ADLAB.COBOL' DMPU26
6 'DNET424.ADLAB.COBOL.BIN' DMPU29
7 'DNET424.ADLAB.COPYLIB' DMPU15
8 'DNET424.ADLAB.COPYLIB.BIN' DMPU14
9 'DNET424.ADLAB.COPYLIBI' DMPU23
10 'DNET424.ADLAB.COPYLIBI.BIN' DMPU13
11 'DNET424.ADLAB.CUST.F4000' DMPU22
12 'DNET424.ADLAB.CUST.ONEREC' DMPU18
13 'DNET424.ADLAB.CUSTFILE' DMPU19
F1=HELP F2=SPLIT F3=END F4=IPT View F5=RFIND F6=RCHANGE
F7=UP F8=DOWN F9=SWAP F10=LEFT F11=RIGHT F12=RETRIEVE

```

Figure B-7 Using the XMC CLIST

Results of the XMC command are displayed in Figure B-8.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (E) ----- LEVEL DNET424.ADLAB ----- Row 1 to 13 of 61
Command ==>
Hotbar: OPRINT REFRESH CLRVOL FILLVOL UPDATE CUT FLIP VALIDATE
*TEMPORARY LIST*
TSO PARMS ==> DEMOMVS.DNET424
Command Member Numbr Data Set Names / Objects Volume
-----
IEBCOPY MESSAGES AND CONTROL STATEMENT
S PAGE 1
IEB1135I IEBCOPY FMID HDZ1180 SERVICE LEVEL UA27650 DATED 20060711 DFSMS 01.
08.00 z/OS 01.08.00 HBB7730 CPU 2094
IEB1035I DNET424 SPIFFY 12:29:38 TUE 25 SEP 2007 PARM='
COPY OUTDD=SYS00191,INDD=((SYS00187,R))
IEB1013I COPYING FROM PDSE INDD=SYS00187 VOL=DMPU20 DSN=DNET424.ADLAB.ADATA
IEB1014I TO PDSU OUTDD=SYS00191 VOL=DMPW02 DSN=SYS07268.T122938.RA000
.DNET424.R0110207
IGW01551I MEMBER ASAM1 HAS BEEN UNLOADED
IGW01551I MEMBER SUBXMP HAS BEEN UNLOADED
IGW01550I 2 OF 2 MEMBERS WERE UNLOADED
IEB147I END OF JOB - 0 WAS HIGHEST SEVERITY CODE
INMX000I 0 message and 23 data records sent as 4275 records to DEMOMVS.DNET424
***

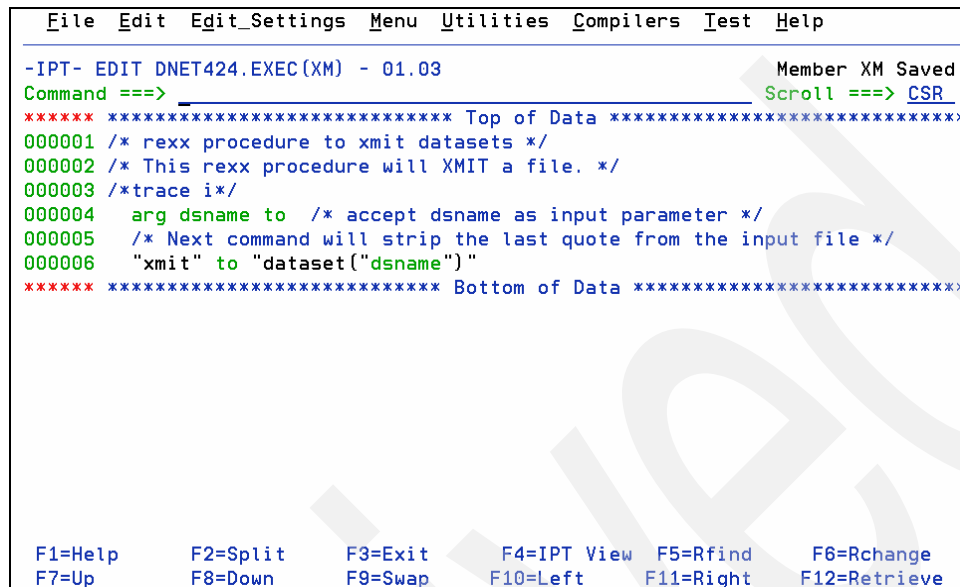
```

Figure B-8 Results of the XMC command



## B.1.5 Using a REXX routine as a User Command

REXX provides more functionality than a CLIST. Figure B-9 displays the REXX program XM.



```
File Edit Edit_Settings Menu Utilities Compilers Test Help
-IPT- EDIT DNET424.EXEC(XM) - 01.03 Member XM Saved
Command ==> Scroll ==> CSR
***** ***** Top of Data *****
000001 /* rexx procedure to xmit datasets */
000002 /* This rexx procedure will XMIT a file. */
000003 /*trace ix*/
000004 arg dsname to /* accept dsname as input parameter */
000005 /* Next command will strip the last quote from the input file */
000006 "xmit" to "dataset(dsname)"
***** ***** Bottom of Data *****

F1=Help F2=Split F3=Exit F4=IPT View F5=Rfind F6=Rchange
F7=Up F8=Down F9=Swap F10=Left F11=Right F12=Retrieve
```

Figure B-9 REXX program XM

Comments begin with `/*` and end with `*/`. The first line of a REXX program must have a comment, with the word `rexx`. The following line satisfies this requirement:

```
/* rexx procedure to xmit datasets*/
```

You can trace REXX statements, using the `TRACE` command, shown in Figure B-9. To trace this routine, remove the `/* */` around the `TRACE` command.

`DSNAME` and `TO` are the two arguments that this program accepts. Data in quotes are passed to the REXX environment, in this case TSO, for execution. Everything in the following line inside the double quotes is passed to TSO. The variables `TO` and `DSNAME` are resolved prior to execution.

```
"xmit" to "dataset(dsname)"
```

Allocate the file, using the `ALTLIB` command, which we described above. Figure B-10 on page 292 is displayed.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (E) ----- LEVEL DNET424.ADLAB ----- "A" will display assist
Command ===>
Hotbar: OPRINT REFRESH CLRVOL FILLVOL UPDATE CUT FLIP VALIDATE
*TEMPORARY LIST*
TSO PARMS ===> demomvs.dnet424
Command Member Numbr Data Set Names / Objects Volume
-----
xm 1 'DNET424.ADLAB.ADATA' DMPU20
2 'DNET424.ADLAB.ADATA.BIN' DMPU26
3 'DNET424.ADLAB.CNTL' DMPU29
4 'DNET424.ADLAB.CNTL.BIN' DMPU24
5 'DNET424.ADLAB.COBOL' DMPU26
6 'DNET424.ADLAB.COBOL.BIN' DMPU29
7 'DNET424.ADLAB.COPYLIB' DMPU15
8 'DNET424.ADLAB.COPYLIB.BIN' DMPU14
9 'DNET424.ADLAB.COPYLIBI' DMPU23
10 'DNET424.ADLAB.COPYLIBI.BIN' DMPU13
11 'DNET424.ADLAB.CUST.F4000' DMPU22
12 'DNET424.ADLAB.CUST.ONEREC' DMPU18
13 'DNET424.ADLAB.CUSTFILE' DMPU19
F1=HELP F2=SPLIT F3=END F4=IPT View F5=RFIND F6=RCHANGE
F7=UP F8=DOWN F9=SWAP F10=LEFT F11=RIGHT F12=RETRIEVE

```

Figure B-10 xm command

## B.2 Example 2: transmitting and receiving files, host-to-PC-to host

Special steps are required when you transfer files from one host to your PC or to another host, for example:

- ▶ PDS or PDSE files are automatically converted to a DATA LIBRARY when an XMIT command is used.
- ▶ Some file types, such as VSAM files, require conversion to a sequential file prior to using the XMIT command.

In our example, we want to transmit several PDS files from one host to another. However, there is no NJE or JES-to-JES communication link defined for the hosts.

The example shows you how to:

1. Create a sequential file for transit using the XMIT command in a REXX procedure.
2. Use FTP to transmit the files to your PC, and then to the target host.
3. Receive the files, converting them back to a PDS or PDSE.

We use a REXX routine to easily XMIT the files. The REXX routine creates the XMIT command, generating the OUTDA parameter automatically.

Figure B-11 on page 293 depicts a REXX routine to XMIT files, creating a sequential file of a PDS for transmission.

```

File Edit Edit_Settings Menu Utilities Compilers Test Help
-IPT- EDIT DNET424.EXEC(XMOUT) - 01.09 Columns 00001 00072
Command ==> Scroll ==> CSR
***** ***** Top of Data *****
000001 /* rexx procedure to xmit datasets */
000002 /* This rexx procedure will create a binary file for shipment */
000003 /* The file name is the same as the input name, with a '.bin' suffix */
000004 trace i /* Trace command commented */
000005 arg dsname /* accept dsname as input parameter */
000006 /* Next command will strip the last quote from the input file */
000007 bin_dsn = substr(dsname,1,(length(dsname)-1))".BIN"
000008 "xmit a.b dataset("dsname") outda("bin_dsn")"
***** ***** Bottom of Data *****

F1=Help F2=Split F3=Exit F4=IPT View F5=Rfind F6=Rchange
F7=Up F8=Down F9=Swap F10=Left F11=Right F12=Retrieve

```

Figure B-11 REXX procedure to transmit files

This routine accepts one parameter, DSNAME. The variable BIN\_DSN constructs the target data set, appending the suffix ".BIN" to the data set. The last line issues the XMIT command to create the data sequential data set. An example is shown in Figure B-12.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-IPT- OLIST (E) ----- LEVEL DNET424.ADLAB ----- "A" will display assist
Command ==> 1-5 xmout SCROLL ==> CSR
Hotbar: OPRINT REFRESH CLRVOL FILLVOL UPDATE CUT FLIP VALIDATE
*TEMPORARY LIST*
TSO PARMS ==>
Command Member Numbr Data Set Names / Objects Volume
-----
1 'DNET424.ADLAB.ADATA' DMPU20
2 'DNET424.ADLAB.CNTL' DMPU29
3 'DNET424.ADLAB.COBOL' DMPU26
4 'DNET424.ADLAB.COPYLIB' DMPU15
5 'DNET424.ADLAB.COPYLIB1' DMPU23
6 'DNET424.ADLAB.CUST.F4000' DMPU22
7 'DNET424.ADLAB.CUST.ONEREC' DMPU18
8 'DNET424.ADLAB.CUSTFILE' DMPU19
9 'DNET424.ADLAB.CUSTFILE.KSDS'
10 'DNET424.ADLAB.CUSTFILE.KSDS.DATA' DMPU27
11 'DNET424.ADLAB.CUSTFILE.KSDS.INDEX' DMPU27
12 'DNET424.ADLAB.CUST1' DMPU25
13 'DNET424.ADLAB.CUST2.KSDS'

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

```

Figure B-12 User command XMOUT

You must issue an ALTLIB command to allocate the REXX procedure.

The OLIST command 1-5 XMOUT creates a binary file for the first five data sets shown in Figure B-12.

The REXX procedure above traced all of the REXX commands using the TRACE I command, as shown in Figure B-13 on page 294.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (E) ----- LEVEL DNET424.ADLAB ----- "A" will display assist
Command ==> 1-5 xmout SCROLL ==> CSR
Hotbar: OPRINT REFRESH CLRVOL FILLVOL UPDATE CUT FLIP VALIDATE
*TEMPORARY LIST*

TSO PARMS ==>
Command Member Numbr Data Set Names / Objects Volume
-----
5 *-x arg dsname /* accept dsname as input parameter */
>>> "'DNET424.ADLAB.ADATA'"
6 *-x /* Next command will strip the last quote from the input file */
7 *-x bin_dsn = substr(dsname,1,(length(dsname)-1))
>V> "'DNET424.ADLAB.ADATA'"
>L> "1"
>V> "'DNET424.ADLAB.ADATA'"
>F> "21"
>L> "1"
>O> "20"
>F> "'DNET424.ADLAB.ADATA'"
8 *-x "xmit a.b dataset('dsname') outda('bin_dsn',BIN)"
>L> "xmit a.b dataset("
>V> "'DNET424.ADLAB.ADATA'"
***

```

Figure B-13 Trace of XMOUT command

You can see the breakdown of each REXX instruction. To remove the TRACE command, delete TRACE line, or comment the line. A TRACE example is shown in Figure B-14.

```

>O> "xmit a.b dataset('DNET424.ADLAB.ADATA'"
>L> ") outda("
>O> "xmit a.b dataset('DNET424.ADLAB.ADATA') outda("
>V> "'DNET424.ADLAB.ADATA'"
>O> "xmit a.b dataset('DNET424.ADLAB.ADATA') outda('DNET424.ADLAB.ADA
TA"
>L> ".BIN')'"
>O> "xmit a.b dataset('DNET424.ADLAB.ADATA') outda('DNET424.ADLAB.ADA
TA.BIN')'"

IEBCOPY MESSAGES AND CONTROL STATEMENT
S PAGE 1
IEB1135I IEBCOPY FMID HDZ1180 SERVICE LEVEL UA27650 DATED 20060711 DFSMS 01.
08.00 z/OS 01.08.00 HBB7730 CPU 2094
IEB1035I DNET424 SPIFFY SPIFFY 12:55:21 TUE 25 SEP 2007 PARM=''
COPY OUTDD=SYS00314,INDD=((SYS00310,R))
IEB1013I COPYING FROM PDSE INDD=SYS00310 VOL=DMPU20 DSN=DNET424.ADLAB.ADATA
IEB1014I TO PDSU OUTDD=SYS00314 VOL=DMPW02 DSN=SYS07268.T125521.RA000
.DNET424.R0110252
IGW01551I MEMBER ASAM1 HAS BEEN UNLOADED
IGW01551I MEMBER SUBXMP HAS BEEN UNLOADED
IGW01550I 2 OF 2 MEMBERS WERE UNLOADED
IEB147I END OF JOB - 0 WAS HIGHEST SEVERITY CODE
INMX000I 0 message and 23 data records sent as 4274 records to A.B
***

```

Figure B-14 XMTOU output

Figure B-15 on page 295 shows the completion of the XMOUT commands.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (E) ----- LEVEL DNET424.ADLAB ----- Row 1 to 13 of 58
Command ===> SCROLL ===> CSR
Hotbar: OPRINT REFRESH CLRVOL FILLVOL UPDATE CUT FLIP VALIDATE
*TEMPORARY LIST*

TSO PARMS ===>
Command Member Numbr Data Set Names / Objects Volume
-----
-XMOUT 1 'DNET424.ADLAB.ADATA' DMPU20
-XMOUT 2 'DNET424.ADLAB.CNTL' DMPU29
-XMOUT 3 'DNET424.ADLAB.COBOL' DMPU26
-XMOUT 4 'DNET424.ADLAB.COPYLIB' DMPU15
-XMOUT 5 'DNET424.ADLAB.COPYLIBI' DMPU23
6 'DNET424.ADLAB.CUST.F4000' DMPU22
7 'DNET424.ADLAB.CUST.F4000.SPAC' DMPU06
8 'DNET424.ADLAB.CUST.ONEREC' DMPU18
9 'DNET424.ADLAB.CUSTFILE' DMPU19
10 'DNET424.ADLAB.CUSTFILE.KSDS'
11 'DNET424.ADLAB.CUSTFILE.KSDS.DATA' DMPU27
12 'DNET424.ADLAB.CUSTFILE.KSDS.INDEX' DMPU27
13 'DNET424.ADLAB.CUST1' DMPU25

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

```

Figure B-15 XMOUT command completion

The XMOUT commands completed successfully. The XMOUT statement created the files that are circled in Figure B-16.

```

File Sort Menu Utils Settings Compilers Test Help Exit
-----
-IPT- - Data Sets Matching DNET424.ADLAB Row 1 of 60
Command ===> Scroll ===> CSR

Command - Enter "/" to select action Message Volume
-----
DNET424.ADLAB.ADATA XMIT RC=0 DMPU20
DNET424.ADLAB.ADATA.BIN DMPU26
DNET424.ADLAB.CNTL DMPU29
DNET424.ADLAB.CNTL.BIN DMPU24
DNET424.ADLAB.COBOL DMPU26
DNET424.ADLAB.COBOL.BIN DMPU29
DNET424.ADLAB.COPYLIB DMPU15
DNET424.ADLAB.COPYLIB.BIN DMPU14
DNET424.ADLAB.COPYLIBI DMPU23
DNET424.ADLAB.COPYLIBI.BIN DMPU13
DNET424.ADLAB.CUSTFILE DMPU19
DNET424.ADLAB.CUSTFILE.KSDS *VSAM*
DNET424.ADLAB.CUSTFILE.KSDS.DATA DMPU27
DNET424.ADLAB.CUSTFILE.KSDS.INDEX DMPU27
DNET424.ADLAB.CUST1 DMPU25
DNET424.ADLAB.CUST2.KSDS *VSAM*

F1=Help F2=Split F3=Exit F4=IPT View F5=Rfind F7=Up
F8=Down F9=Swap F10=Left F11=Right F12=retrieve

```

Figure B-16 New XMIT files

Unlike PDS or PDSE files, you can transmit these files to your PC and then to another host.

We elected to use FTP to transfer the files. We could use PCOM or any 3270 emulator that supports file transmissions.

XMOUT creates the OUTDA data set as a Fixed Block, character data set with a block size of 3120:

- ▶ If you FTP to transmit the data set from your PC to the host, and the host file already exist, it must have the file attributes Fixed Block, 80 characters.

- If you use a 3270 emulator to send the file from your PC to the host, you must either set up the emulator variables to catalog new files as Fixed Block, 80 characters, or transfer it to an existing file with Fixed Block 80 character attributes.

If these rules are not followed, the RECEIVE command will fail.

Figure B-17 shows the FTP commands.

```
C:\Documents and Settings\Administrator>cd \temp
C:\temp>ftp demomvs.demopkg.ibm.com
Connected to demomvs.demopkg.ibm.com.
220-FTPD1 IBM FTP CS V1R8 at DEMOMVS.DEMOPKG.IBM.COM, 15:04:31 on 2007-09-24.
220 Connection will close if idle for more than 5 minutes.
User (demomvs.demopkg.ibm.com:(none)): dnet424
331 Send password please.
Password:
230 DNET424 is logged on. Working directory is "DNET424.".
ftp> bin
200 Representation type is Image
ftp> get adlab.adata.bin
200 Port request OK.
125 Sending data set DNET424.ADLAB.ADATA.BIN FIXrecfm 80
250 Transfer completed successfully.
ftp: 341760 bytes received in 0.89Seconds 383.57Kbytes/sec.
ftp> quit
221 Quit command received. Goodbye.

C:\temp>ftp smpomva.dfw.ibm.com
Connected to smpomva.dfw.ibm.com.
220-FTPSEIVE IBM FTP CS V1R7 at SMPOMVA.DFW.IBM.COM, 15:06:15 on 2007-09-24.
220 Connection will close if idle for more than 5 minutes.
User (smpomva.dfw.ibm.com:(none)): jrice1
331 Send password please.
Password:
230 JRICE1 is logged on. Working directory is "JRICE1.".
ftp> bin
200 Representation type is Image
ftp> put adlab.adata.bin
200 Port request OK.
125 Storing data set JRICE1.ADLAB.ADATA.BIN
250 Transfer completed successfully.
ftp: 341760 bytes sent in 4.73Seconds 72.19Kbytes/sec.
ftp> quit
221 Quit command received. Goodbye.

C:\temp>
```

Figure B-17 FTP commands to GET and PUT the files

**Note:** With most z/OS hosts, you can determine the correct IP address by issuing the command HOMETEST from option 6. As shown in Figure B-17, you must change the file representation to Binary before initiating the transmission.

Use the ?RECEIVE command to receive all of the files, as shown in Figure B-18 on page 297 and Figure B-19 on page 297.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-IPT- OLIST (E) ----- LEVEL DNET424.ADLAB.*.BIN "A" will display assist
Command ==> 1-5 ?receive SCROLL ==> CSR
Hotbar: OPRINT REFRESH CLRVOL FILLVOL UPDATE CUT FLIP VALIDATE
*TEMPORARY LIST*

Command Member Numbr Data Set Names / Objects Volume
-----
1 'DNET424.ADLAB.ADATA.BIN' DMPU17
2 'DNET424.ADLAB.CNTL.BIN' DMPU14
3 'DNET424.ADLAB.COBOL.BIN' DMPU19
4 'DNET424.ADLAB.COPYLIB.BIN' DMPU17
5 'DNET424.ADLAB.COPYLIBI.BIN' DMPU17
----- END OF LIST -----

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

```

Figure B-18 RECEIVE command

```

-IPT----- OLIST SHOWCMD PANEL -----
COMMAND ==>

Verify or change the TSO or CLIST command:

==> RECEIVE 'DNET424.ADLAB.ADATA.BIN'

Press ENTER to proceed or the END key to cancel the command.

IQIP730 Enter QUIT to abort executing "?RECEIVE" before end of range.

F7=UP F8=DOWN F9=SWAP F10=LEFT F11=RIGHT F12=RETRIEVE

```

Figure B-19 RECEIVE command

Change the syntax, as shown in Figure B-20 on page 298 and Figure B-21 on page 298 to receive the data for the first file.

```

-IPT----- OLIST SHOWCMD PANEL -----
COMMAND ==>

Verify or change the TSO or CLIST command:

    ==> RECEIVE inda('DNET424.ADLAB.ADATA.BIN')_

Press ENTER to proceed or the END key to cancel the command.

IQUIP730 Enter QUIT to abort executing "?RECEIVE" before end of range.

F7=UP      F8=DOWN    F9=SWAP   F10=LEFT   F11=RIGHT  F12=RETRIEVE

```

Figure B-20 Modifying the RECEIVE command prior to execution

```

08.00 z/OS 01.08.00 HBB7730 CPU 2094
IEB1035I DNET424 SPIFFY SPIFFY 09:48:52 MON 24 SEP 2007 PARM=''
COPY OUTDD=SYS00305,INDD=((SYS00301,R))
IEB1013I COPYING FROM PDSE INDD=SYS00301 VOL=DMPU20 DSN=DNET424.ADLAB.ADATA
IEB1014I TO PDSU OUTDD=SYS00305 VOL=DMPW02 DSN=SYS07267.T094852.RA000
,DNET424.R0103268
IGW01551I MEMBER ASAM1 HAS BEEN UNLOADED
IGW01551I MEMBER SUBXMP HAS BEEN UNLOADED
IGW01550I 2 OF 2 MEMBERS WERE UNLOADED
IEB147I END OF JOB - 0 WAS HIGHEST SEVERITY CODE
INMX000I 0 message and 23 data records sent as 4272 records to A.B
INMX001I Transmission occurred on 09/24/2007 at 09:48:52.
***

```

Figure B-21 Receiving the files

Repeat the process for all of the binary files.



## Permanent OLISTs and UNIX System Services objects

This appendix continues the saga of the UNIX System Services object, which we were going to invoke from the ELUX permanent OLIST, as described in Chapter 2, “The Object List” on page 7.

We were given an HFS filename where the developers were planning to put the REXX exec to FTP the output data directly to the Asset Management department.

When we Browsed that file, we found the familiar “Hello World” stub as shown in Figure C-1.

```

BROWSE -- /u/se16661/ftpdnld.rexx ----- Line 00000000 Col 001 033
Command ==> _____ Scroll ==> CSR
***** Top of Data *****
/*REXX (/u/se16661/ftpdnld.rexx)*/
SAY "Hello World"
***** Bottom of Data *****

```

*Figure C-1 Browsing the HFS file holding the REXX exec to see the stage of development*

We see that the promised code was not yet delivered or just the stub so far. Perhaps the planned delivery date was over-ambitious.

We can test out our driver REXX exec, called TRYIX. This was written separately on a private user library (SE16661.USER.EXEC) for testing. This library is currently concatenated at the top of the SYSEXEC concatenations. We issue an OLDD SYSEXEC command, as shown in Figure C-2 on page 300.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (B) ----- Asset Management ----- "A" will display assist
Command ===> oldd sysexec_ SCROLL ===> CSR
Hotbar: FLIP CLRVOL FILLVOL REFRESH UTIL CUT SET UPDATE
Open list ===> ELUX (or BLANK for reference list)
TSO PARMS ===>
Command Member Numbr Data Set Names / Objects Class
-----
FTP%LD 1 !----- FTP Up/Download -----
2 'SE16661.ELUX.PCTL' PDSE
3 !----- Spreadsheets -----
4 'SE16661*ELUX*CSV' LIST
5 !----- Program elements -----
CONASS 6 'SE16661.ELUX.EXEC' PDSE
REXXCL 7 'SE16661.ELUX.JCL' PDSE
CONASS 8 'SE16661.ELUX.JCL' PDSE
9 !----- Job status -----
10 >OUT SE16661C USER
11 +SDSF ST SE16661C CMD
12 !----- FTP transfer in OMVS -----
13 )/u/se16661/ftpdnld.rex OE
----- END OF LIST -----

```

Figure C-2 OLDD SYSEEXEC shows all the libraries allocated to DDname SYSEEXEC – Part-1

This displays a temporary OLIST that shows all of the libraries allocated to SYSEEXEC in order, as shown in Figure C-3.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (B) ----- ALLOCATION LIST FOR "SYSE "A" will display assist
Command ===> _ SCROLL ===> CSR
Hotbar: FLIP CLRVOL FILLVOL REFRESH UTIL CUT SET UPDATE
*TEMPORARY LIST*
TSO PARMS ===>
Command Member Numbr Data Set Names / Objects Class
-----
-SYSEEXEC 1 'SE16661.USER.CEXEC'
2 'SE16661.USER.EXEC'
3 'SYS1.SBPXEXEC'
4 'BOOKMAN.SEOYCLIB'
5 'SYS1.SEDGEXE1'
6 'FFST.SEPWSRC1'
7 'ISP.SISPEXEC'
8 'ISF.SISFEXEC'
9 'DCE.SEUVEXEC'
10 'DFS.SIOEEXEC'
11 'INFOPT.SAOPEXEC'
12 'QMF.V7R1M0.SDSQEXCE'
13 'FILEMGR.V7R1.SFMNEXEC'
14 'LDAP.SGLDEXEC'
----- END OF LIST -----

```

Figure C-3 OLDD SYSEEXEC shows all the libraries allocated to DDname SYSEEXEC – Part-2

Library type CEXEC contains compiled REXX code, but the TRYIX exec is in library type EXEC because it is not (yet) compiled.

We could type a B against line 2 to Browse the contents of the library, but instead we return to our permanent 'ELUX' OLIST and add it there, specifically by member-name.

In Figure C-4, we added the TRYIX exec as an object (number 14) to the bottom of our OLIST.

Go into Edit on the program to see what it looks like.

We type E in the command field of line 14.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (B) ----- Asset Management ----- Row 1 to 14 of 14
Command ==> SCROLL ==> CSR
Hotbar: FLIP CLRVOL FILLVOL REFRESH UTIL CUT SET UPDATE
Open list ==> ELUX (or BLANK for reference list)
TSO PARMS ==>
Command Member Numbr Data Set Names / Objects Class
-----
1 !----- FTP Up/Download -----
FTP%LD 2 'SE16661.ELUX.PCTL' PDSE
3 !----- Spreadsheets -----
4 'SE16661*ELUX*CSV' LIST
5 !----- Program elements -----
CONASS 6 'SE16661.ELUX.EXEC' PDSE
REXXCL 7 'SE16661.ELUX.JCL' PDSE
CONASS 8 'SE16661.ELUX.JCL' PDSE
9 !----- Job status -----
10 >OUT SE16661C USER
11 +SDSF ST SE16661C CMD
12 !----- FTP transfer in OMVS -----
13 )/u/se16661/ftpdnld.rex OE
E_ TRYIX 14 USER.EXEC PDSE
----- END OF LIST -----

```

Figure C-4 Editing the driver exec TRYIX – Part 1

Press Enter to open the member in Edit, as shown in Figure C-5 on page 302.

```

File Edit Edit_Settings Menu Utilities Compilers Test Help

-IPT- EDIT SE16661.USER.EXEC(TRYIX) - 01.26 Columns 00001 00072
Command ==> Scroll ==> CSR
***** Top of Data *****
000001 /*REXX(TRYIX)*/ TRACE "N"
000002 ADDRESS ISPEXEC "CONTROL ERRORS RETURN"
000003 PARSE ARG fn .
000004 "ALLOC DD(SYSUT2) NEW REUSE UNIT(VIO) SP(5) TR LRECL(255) RECFM(V B)"
000005 uc = BPXWUNIX(fn,,stdout.,stderr.) /* execute in UNIX environ */
000006 DO i = 1 TO stdout.0; QUEUE stdout.i; END
000007 DO i = 1 TO stderr.0; QUEUE stderr.i; END
000008 q = QUEUED()
000009 "EXECIO" q "DISKW SYSUT2 (FINIS"
000010 ADDRESS ISPEXEC "LINIT DATAID(OE) DDNAME(SYSUT2)"
000011 PARSE VALUE fn "ENDED RC="uc WITH . zedsmg 1 zedlmsg
000012 ADDRESS ISPEXEC "SETMSG MSG(ISRZ000)"
000013 ADDRESS ISPEXEC "BROWSE DATAID(&OE)"
000014 ADDRESS ISPEXEC "LMFREE DATAID(&OE)"
000015 IF( uc>=0 & uc<=255 )
000016 THEN EXIT 0
000017 ELSE EXIT ABS(uc)
***** Bottom of Data *****

```

Figure C-5 Editing the driver exec TRYIX – Part 2

Table C-1 provides an explanation of Figure C-5.

Table C-1 TRYIX explanation

Line number	Command	Explanation
000001	/*REXX(TRYIX)*/ TRACE "N"	REXX comment to indicate that it's a REXX exec and a TRACE requesting Normal tracing.
000002	ADDRESS ISPEXEC "CONTROL ERRORS RETURN"	Continues processing if an ISPF 'error' is encountered.
000003	PARSE ARG fn	Accepts an argument of the HFS filename, and forgets anything afterwards.
000004	"ALLOC DD(SYSUT2) NEW REUSE UNIT(VIO) SP(5) TR LRECL(255) RECFM(V B)"	Allocates a temporary file in cache to hold the output or any errors that are encountered.
000005	uc = BPXWUNIX(fn,,stdout.,stderr.) /* execute in UNIX environ */	Calls the BPXWUNIX (UNIX System Services) function, which runs a UNIX shell command and optionally provides its input and traps any output or errors and could export a set of environment variables ( <i>but here we just use the standard and error output stems</i> ).
000006	DO i = 1 TO stdout.0; QUEUE stdout.i; END	Reads the stem that contains the trapped output and writes it to the queue.
000007	DO i = 1 TO stderr.0; QUEUE stderr.i; END	Reads the stem that contains any trapped errors and writes it to the queue.
000008	q = QUEUED()	Saves the number of records queued.
000009	"EXECIO" q "DISKW SYSUT2 (FINIS"	Writes the queued records to the temporary file.
000010	ADDRESS ISPEXEC "LINIT DATAID(OE) DDNAME(SYSUT2)"	Allocates a token to the temporary data set to allow ISPF to access it.

Line number	Command	Explanation
000011	PARSE VALUE fn "ENDED RC="uc WITH . zedsmg 1 zedlmsg	Loads the short and long ISPF error messages with the results of the UNIX call. The long message contains the filename whereas the short will not.
000012	ADDRESS ISPEXEC "SETMSG MSG(ISRZ000)"	Sets the ISPF message number to be displayed.
000013	ADDRESS ISPEXEC "BROWSE DATAID(&OE)"	Browses the temporary data set with the token.
000014	ADDRESS ISPEXEC "LMFREE DATAID(&OE)"	Frees the token.
000015	IF( uc>=0 & uc<=255 )	Allowable return codes are from 0-255 in UNIX System Services.
000016	THEN EXIT 0	If its OK, end with a zero return code.
000017	ELSE EXIT ABS(uc)	Otherwise (negative or 1000, which is a stop signal or something else, pass-through).

So what happens actually? Return to the OLIST to see.

We type the command name TRYIX in the command field of line 13, as shown in Figure C-6.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-----
-IPT- OLIST (B) ----- Asset Management ----- Row 1 to 14 of 14
Command ==> SCROLL ==> CSR
Hotbar: FLIP CLRVOL FILLVOL REFRESH UTIL CUT SET UPDATE
Open list ==> ELUX (or BLANK for reference list)
TSO PARMS ==>
Command Member Numbr Data Set Names / Objects Class
-----
1 !----- FTP Up/Download -----
FTP%LD 2 'SE16661.ELUX.PCTL' PDSE
3 !----- Spreadsheets -----
4 'SE16661*ELUX*CSV' LIST
5 !----- Program elements -----
CONASS 6 'SE16661.ELUX.EXEC' PDSE
REXXCL 7 'SE16661.ELUX.JCL' PDSE
CONASS 8 'SE16661.ELUX.JCL' PDSE
9 !----- Job status -----
10 >OUT SE16661C USER
11 +SDSF ST SE16661C CMD
12 !----- FTP transfer in OMVS -----
TRYIX 13 )/u/se16661/ftpdnld.rex OE
-E 14 USER.EXEC PDSE
----- END OF LIST -----

```

Figure C-6 Executing the driver exec TRYIX against the HFS file

Press Enter, and ISPF-PT executes the command passing it to the name of the UNIX System Services object (without the leading ") UNIX System Services type identifier). The UNIX exec completed with a return code of 255, as shown in Figure C-7 on page 304.

```

Menu Utilities Compilers Help
BROWSE      SYS07282.T193434.RA000.SE16661.R0106879      Line 00000000 ENDED RC=255
Command ==>
***** Top of Data *****
Hello World
***** Bottom of Data *****

```

Figure C-7 The trapped output of the executed HFS exec is impressively displayed

This is because none was explicitly set in the HFS exec. If we now press the END key, we return to the OLIST, as shown in Figure C-8.

```

File Edit Find Display Populate Settings Menu Util Test Help Exit
-IPT- OLIST (B) ----- Asset Management ----- Row 1 to 14 of 14
Command ==>
Hotbar: FLIP CLRVOL FILLVOL REFRESH UTIL CUT SET UPDATE
Open list ==> ELUX (or BLANK for reference list)
TSO PARMS ==>
Command Member Numbr Data Set Names / Objects Class
-----
1 !----- FTP Up/Download -----
FTP%%LD 2 'SE16661.ELUX.PCTL' PDSE
3 !----- Spreadsheets -----
4 'SE16661*ELUX*CSV' LIST
5 !----- Program elements -----
CONASS 6 'SE16661.ELUX.EXEC' PDSE
REXXCL 7 'SE16661.ELUX.JCL' PDSE
CONASS 8 'SE16661.ELUX.JCL' PDSE
9 !----- Job status -----
10 >OUT SE16661C USER
11 +SDSF ST SE16661C CMD
12 !----- FTP transfer in OMVS -----
-TRYIX 13 )/u/se16661/ftpdnld.rex OE
-E TRYIX 14 USER.EXEC PDSE
----- END OF LIST -----

```

Figure C-8 Returning to the OLIST after the command the status in the command area is updated

# Customizing the IBM Personal Communications

Most 3270 emulators allow you to customize the keyboard or the mouse. In this chapter, we show you how to customize the IBM Personal Communications (PCOM) product to use the mouse to navigate using the ISPF Productivity Tool. If you are using a different 3270 emulator, consult the appropriate documentation to change your mouse settings. Using the PCOM emulator, we redefine our mouse left-click key to open files, and the mouse right-click to close files. Using this method, you can quickly navigate Object Lists and Member Selection Lists with a click of the mouse.

## Current PCOM mouse settings

With PCOM, the default mouse settings are:

- ▶ Left click – Marks the box around text for cut and paste activities
- ▶ Right click – Displays keys

Defining the mouse for Point-and-Click for ISPF-PT disables the mouse functions above. In the following sections, we describe how to perform the above functions without a mouse.

## Marking the box for cut/paste activities without the mouse

Figure D-1 shows the basic keyboard arrows.

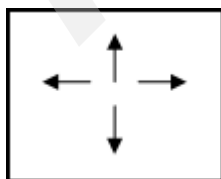


Figure D-1 Keyboard arrows

To mark a box around text for cut/paste activities without the mouse:

1. Use the keyboard arrows shown in Figure D-1 on page 305 to move the cursor to a corner of the area that you want to mark.
2. While holding down the Shift key, use the cursor-movement keys to mark the area.
3. Release the Shift key.

You can use the mouse to click Edit → Copy to continue with the copy operation.

## Displaying the pop-up keypad panel without a mouse

Figure D-2 shows the basic pop-up keypad panel.

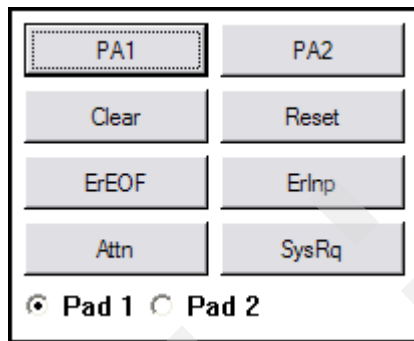


Figure D-2 Pop-up keypad panel

To display the pop-up keypad, shown in Figure D-2, without using the mouse:

1. Left-click the **Action** menu bar command.
2. Left-click the display pop-up keypad.

Because you learned to use an alternate method to mark text and to display the pop-up Keypad, you are ready to program your mouse to open and close files.

## Changing PCOM to use the mouse to open and close files

Use the following steps to customize PCOM to use the mouse to open and close files:

1. Open a PCOM Session.
2. Left-click **Edit** → **Preferences** → **Macro/Script**.
3. Click the **Customize** button.
4. Click the pull-down list. Scroll down until you find the mouse position. Click the mouse position to highlight it, and then click **Add**.
5. Scroll up until you find Enter. Click Enter to highlight it, and click **Add**. You should now see both commands [mouse position] and [enter] in the Macro Statements box.
6. Click **File** → **Save**. Enter a name of the Macro, such as mouse.mac., as shown in Figure D-3 on page 307.



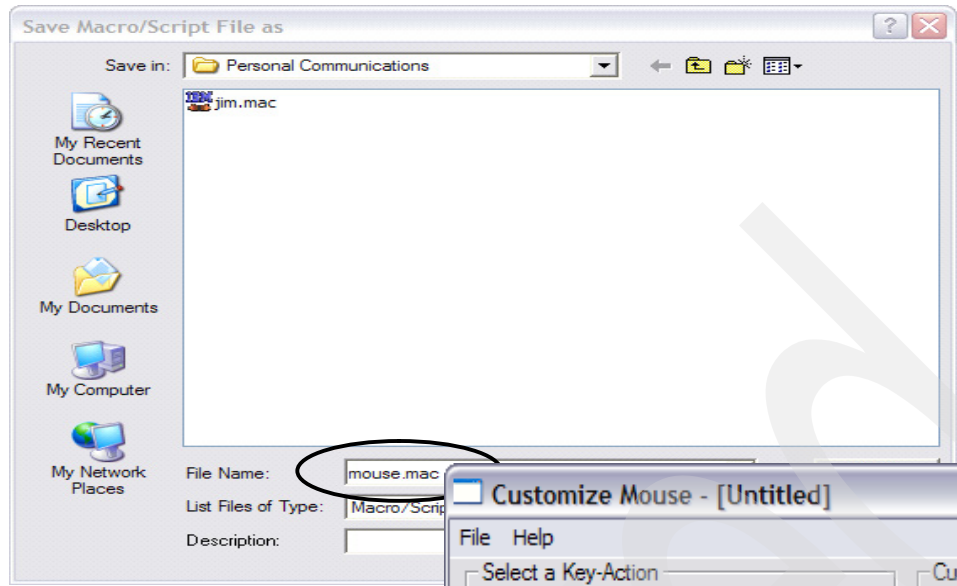


Figure D-3 Macro/Script File save

7. Close the Macro/Script window by clicking on the "X" in the upper-right corner of the box. The previous steps defined the macro file "mouse.mac" and positioned the cursor and invoked the **Enter** key. The next series of steps defines the mouse actions.
8. Left-click **Edit** → **Preferences** → **Mouse**.
9. Left-click the User-Defined radio button, and then left-click the Customize box.
10. When you see the completed Customization panel, follow these steps:
  - a. Left-click in the Macro/Script box, and locate the Macro/Script file name that you saved earlier.
  - b. When you find the correct file name, click the Left box of the **Current Action of Mouse** Button, which enables the Macro/Script that you created in earlier using the left-mouse click.
  - c. Left-click the Function Pull Down, and locate **Program Function Key 3**. Your PF3 key is set to END, which saves and closes files. Select the **Program Function Key 3**.
  - d. Left-click the Right box of the **Current Action of Mouse** Button.
  - e. Click the red "X" to close the Customize Mouse panel, and Figure D-4 is displayed.

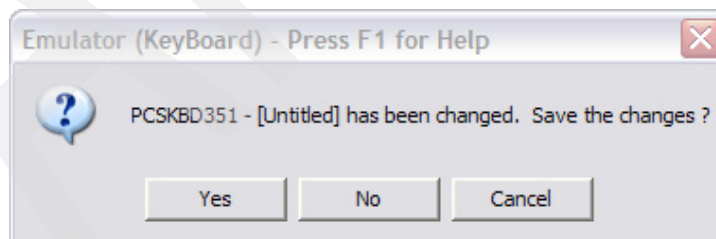


Figure D-4 Emulator (keyboard) confirmation panel

- f. Click **Yes**.
- g. Enter a valid file name. Save the MMP file.

## Touring ISPF-PT using the mouse enabled point-and-click function

Now our mainframe session is starting to behave like to our workstation. We can open and close files with a click of our mouse, as shown in Figure D-5 and Figure D-6.

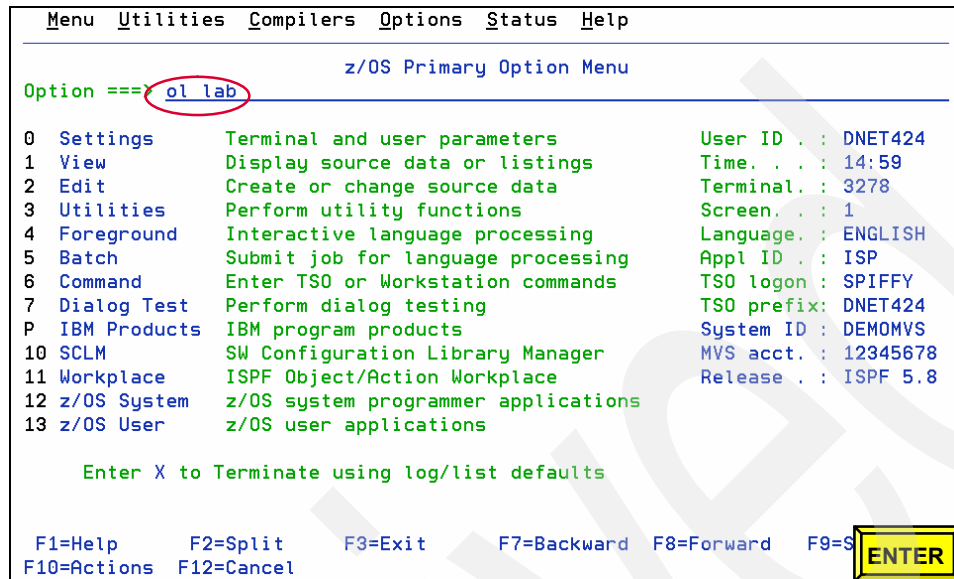


Figure D-5 Request to display the Object List LAB

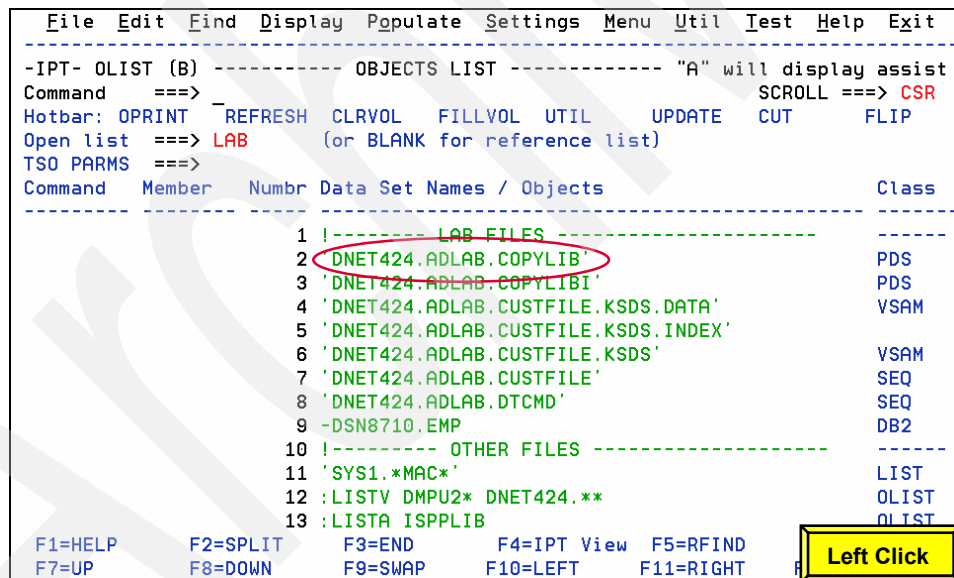


Figure D-6 Object List LAB

1. Left-click DNET424.ADLAB.COPYLIB, which is the MSL in Figure D-7 on page 309.

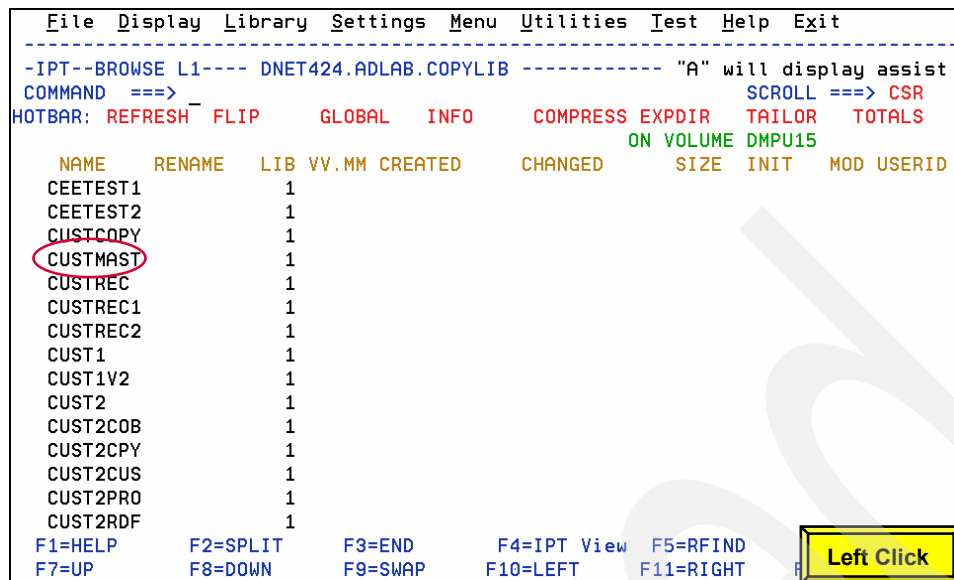


Figure D-7 Member Selection List Shown

2. Left-click CUSTMAST to browse the member. Figure D-8 is displayed.

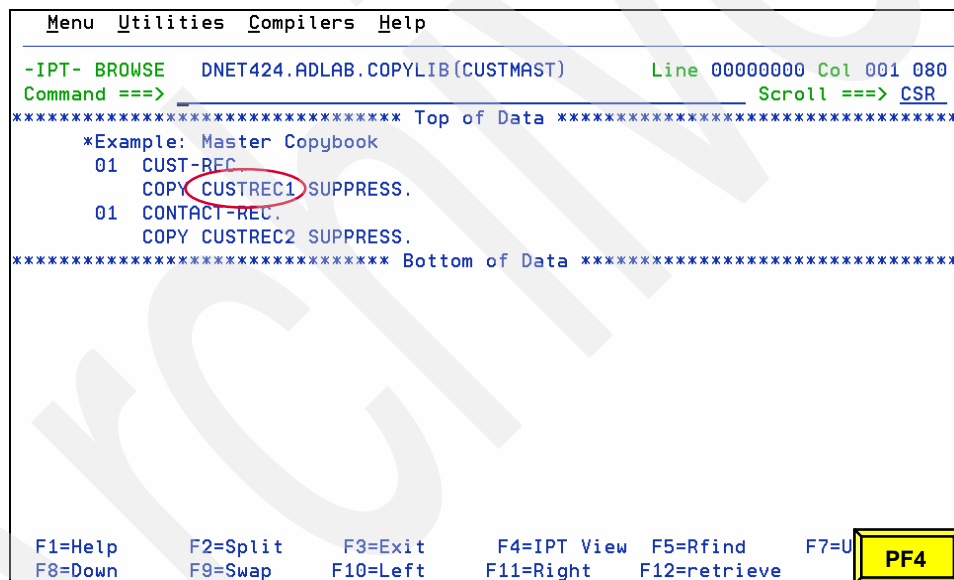


Figure D-8 Contents of CUSTMAST

We customized PF4 using the VIEW command. Put your cursor under CUSTREC1, and press PF4 to view the CUSTREC1 member. Figure D-9 on page 310 is displayed.

```

File Edit Edit_Settings Menu Utilities Compilers Test Help
-IPT- VIEW DNET424.ADLAB.COPYLIB(CUSTREC1) - 01.00 Columns 00001 00072
Command ==> Scroll ==> CSR
***** Top of Data *****
000001 *** *****
000002 * Sample COBOL Copybook for IBM PD Tools Workshops
000003 *
000004 * The sample data described by this copybook
000005 * is <USERID>.ADLAB.CUSTFILE
000006 *** *****
000007 05 CUSTOMER-KEY.
000008 10 CUST-ID PIC X(5).
000009 10 REC-TYPE PIC X.
000010 05 NAME PIC X(17).
000011 05 ACCT-BALANCE PIC S9(7)V99 COMP-3.
000012 05 ORDERS-YTD PIC S9(5) COMP.
000013 05 ADDR PIC X(20).
000014 05 CITY PIC X(14).
000015 05 STATE PIC X(02).
000016 05 COUNTRY PIC X(11).
000017 05 MONTH PIC S9(7)V99 COMP-3 OCCURS 12
F1=Help F2=Split F3=Exit F4=IPT View F5=Rfind F6=Right Click
F7=Up F8=Down F9=Swap F10=Left F11=Right

```

Figure D-9 CUSTREC1 member view

Using PF4, we can view the CUSTREC1 member. Right-click to invoke the END command, and return to the previous panel.

## Restoring the PCOM mouse settings

Use the following steps to easily restore the PCOM Mouse settings:

1. Left-click **Edit** → **Preferences** → **Mouse**.
2. Select **IBM Default**, as shown in Figure D-10.

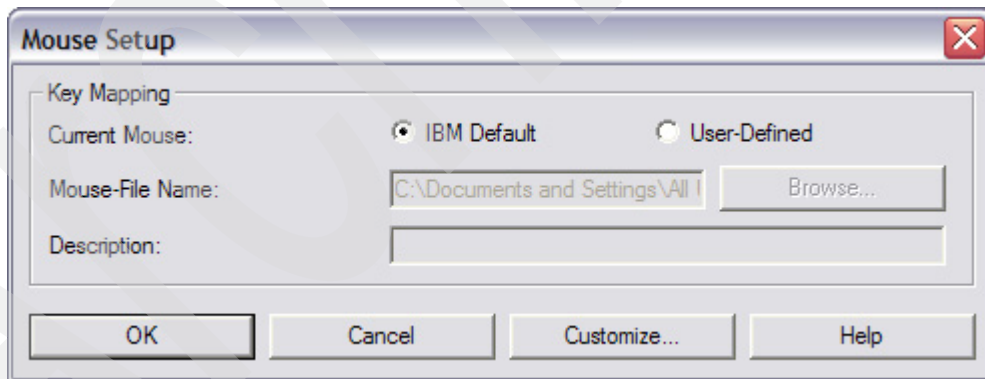


Figure D-10 Mouse Setup with IBM Default Restored

3. Click **OK** to restore the defaults.

# Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

## IBM Redbooks

For information about ordering these publications, see “How to get Redbooks” on page 312. Note that some of the documents referenced here may be available in softcopy only.

- ▶ *IBM Application Development and Problem Determination Tools V7 for System z: Application Performance Analyzer, Debug Tool Utilities and Advanced Functions, Fault Analyzer, File Export, File Manager, and Workload Simulator*, SG24-7372
- ▶ *Getting Started with SCLM: A Practical Guide to SCLM and SCLM Advanced Edition*, SG24-7392

## Other publications

These publications are also relevant as further information sources:

- ▶ *ISPF Productivity Tool - User's Guide*, SC32-1533-01
- ▶ *ISPF Productivity Tool - Installation Guide*, SC32-1532-01
- ▶ *ISPF Productivity Tool - Program Directory*, GI11-4091-02
- ▶ *ISPF User's Guide Vol 1*, SC34-4822-04
- ▶ *ISPF User's Guide Vol 2*, SC34-4823-04
- ▶ *ISPF Dialog Developer's Guide and Reference*, SC34-4821-04
- ▶ *ISPF SCLM Project Manager's and Developer's Guide*, SC34-4817-04
- ▶ *File Manager User Guide and Reference*, SC19-1037-00
- ▶ *TSO/E User's Guide*, SA22-7794-03
- ▶ *TSO/E CLISTs*, SA22-7781-03
- ▶ *TSO/E REXX Reference*, SA22-7790-06

## Online resources

These Web sites are also relevant as further information sources:

- ▶ ISPF Productivity Tool documentation and support:  
<http://www-306.ibm.com/software/awdtools/ispfproductivitytool/>
- ▶ IBM ShopzSeries for planning and ordering zSeries software:  
<https://www14.software.ibm.com/webapp/ShopzSeries/ShopzSeries.jsp>
- ▶ ISPF for z/OS documentation and support  
<http://www.ibm.com/software/awdtools/ISPF/library/>
- ▶ File Manager documentation and support:  
<http://www.ibm.com/software/awdtools/filemanager/>

## How to get Redbooks

You can search for, view, or download Redbooks, Redpapers, Technotes, draft publications and Additional materials, as well as order hardcopy Redbooks, at this Web site:

[ibm.com/redbooks](http://ibm.com/redbooks)

## Help from IBM

IBM Support and downloads

[ibm.com/support](http://ibm.com/support)

IBM Global Services

[ibm.com/services](http://ibm.com/services)

# Index

## Symbols

?XMIT command with an Object List 286

## A

activate SCLM support 142  
Allocation List from an OLDD command without a DD-name parameter 9  
APF libraries 23  
APF library objects 23  
ASSIST command 19  
automatic integrated access 5  
automatic PDS compression 5  
automatic PDS directory expansion 5

## B

Browse, View and Edit commands with File Manager 134  
build and promote the SCLM ARCHDEF members 161

## C

catalog entry from an object list 28  
CBPDO tape 194  
class display 24  
clipboards 118  
CLIST as a User command 289  
converting two sample CLISTs into Rexx 250  
COPY command 93  
COPY request 96  
customization wizard 265  
customize your ISPF Keys for Point-and-Shoot 185  
customizing IBM Personal Communications (PCOM) 305  
customizing IPT to invoke File Manager 132  
Customizing IPT with View 185  
cut example 116

## D

data set descriptions 196  
DEFAULT command 76  
defining PDSEs 86  
definition of terms 1  
Del command 27  
deleting datasets 32  
development environment 240  
Directory Stow error 88  
DS commnad 288  
DSLISIT command with an Object list 288

## E

ELUX project 34  
ELUX workFlow 49  
enhanced clipboards 5

enhanced Cut and Paste 115  
enhanced data set patterns 5  
enhanced DSLIST ISPF 5  
experimenting with other UDOs 276  
exploration of system datasets 15  
extending exec into a general purpose user defined object 276

## F

File Manager for z/OS 131  
File Manager Template Workbench 136  
File Manager walkthrough 135  
File Manager with IPT 134  
FILLVOL command 46  
FILTER command 13, 17–18, 92  
FILTERing only datasets 13  
FINDTEXT command 69  
FindText command 83  
FINDTEXT OLIST command 69  
first time logon to IPT 179  
FLIP command 14, 18, 68, 83  
FLIPed object list 14  
FLIPped object list 19  
FTP command to start to populate the Permanent OLIST 37  
FTP upload command 49, 61

## G

Global Change 81  
Global Edit Command settings parameters 77  
Global Edit commands 81  
Global Find 80, 82  
Global Find and Change command invocation methods 79  
Global Find and Global Change command 76  
Global Find PROGRAM-ID 91  
Global IPT Defaults 77  
Global Settings for a Specific Search 78  
Global, Target and Distribution Zone options 206  
group execution 28  
group execution by generic reference 31  
group execution by number reference 28  
group of datasets in one IPT command 29

## H

History Command List 175  
History Command Shell 175  
History Commands 171  
HOMETEST command 173

## I

identify members and move them to another PDS 97

- implement an OLIST User Defined Object (UDO) 239
- installation requirements for IPT 194
- installing out-of-the-box ISPF Productivity Tool 193
- intuitive searching of generic patterns 10, 48
- IPT commands with the IPT? command 22
- IPT functional overview 5
- IPT History commands 172
- IPT History Commands List 170
- IPT integration with other products 3
- IPT productivity 3
- IPT SCLM walkthrough 149
- IQIRDUMY 276
- IQIRERRO 277
- ISSET command 77
- ISPF concatenated libraries to manage change 100
- ISPF Productivity Tool 1
- ISPF Productivity Tool installation 196
- ISPF Productivity Tool maintenance 234

## L

- learning IPT 2
- Libraries for Target and Distribution Zone 203
- LINKLIST, LPALIB, APF and PARMLIB libraries 17
- LISTDIR member 120
- LISTVTOC object list 21
- logon procedure 180
- logon procedure 180

## M

- meaningful UDOs 277
- Member Selection List 73
- MEMFIND command 64–65
- MEMFIND command with Global Exclude 67
- MEMFIND search 65
- MEMFIND with a wild card 68
- method for exploring UDO 240
- MSL 1
- MSL command 75
- MSL FindText 83
- MSL functions 74
- MSL Hotbars 111
- MSL member identification functions 90
- MSL Point-and-Shoot fields 108
- MYPROJ object list 4

## N

- NOT FIND 26

## O

- Object Linking and Embedding 4
- object list 4
- object list matching a pattern 11
- Object List Options panel 36
- OL/ command 15
- OLDD command without a DDname parameter 8
- OLDD command 247
- OLDD SYSPROC command 248
- OLIST defaults using ISET 35

- OLIST defaults using ISET option P 35
- OLIST description 38
- OLIST of the VTOC of a disk volume 21
- Option G from the IPT default 77
- OUT UDO 57
- OUT UDO and an SDSF command 45

## P

- Partitioned Data Set Extended 86
- paste example 119
- paste TSO command 124
- PASTE VIEW 127
- PASTE with VIEW command 127
- patterns 11
- Permanent Command List 176
- Permanent Commands List 171
- permanent object lists 34
- Permanent OLIST 36
- permanent OLISTs and USS objects 299
- Persistent Table Management 181
- Persistent Tables defined by the system programmer 182
- Point and Shoot commands with a Member Selection List 189
- Point-and-Shoot 5
- Point-and-Shoot commands 189
- pop-up selection panel 16
- product distribution tape 195
- Project MSL command 104
- PROMPT message 92
- provides extensive search capabilities 5

## Q

- QUIT STOP 33

## R

- RACF profile 229
- recalling commands 177
- Redbooks Web site 312
  - Contact us xii
- relative file tape 195
- reset Del command 27
- REXX routine as a User command 291

## S

- SCLM 142
- SCLM Edit using an MSL 146
- SCLM integration with IPT 141
- SCLM Libraries 146
- SCLM options 144
- SCLM, Librarian, Panvalet support 5
- SDSF command 58
- Search commands 76
- SMP/E Global Zone CSI Libraries 198
- SMP/E Target and Distribution Zone Libraries 199
- SMP/E Temporary Libraries for Global Zone 201
- SORT command 20
- supplied wizard CLIST is called IQIWIZRD 264



system libraries sorted by Volume 20

## T

TAILOR command 113  
toggling from Volume to Class 15  
transmitting and receiving files - host to host 286  
transmitting and receiving files - host to PC to host 292  
TRYIX 303  
TSO Command Shell 5, 169  
TSO Command Shell demonstration 172  
TSO Command Shell options 169  
TSO Command Shell with the History Command List 171  
TSO Command Shell with the Permanent Command List 172  
TSO LISTDS command 124  
TSO LOGON Procedure to invoke ISPF Productivity Tool 230  
TSO shell options 170

## U

UDO 2  
UDO and SDSF invocations 59  
UPDATE command 42  
use PDSs rather than PDSEs 87  
user commands from an Object List 285

## V

validate command 24  
View a group of datasets 30  
View Command 186  
VSAM data set interface 133  
VSAM file from the IPT Object List 135  
VSAM interface to the IBM File Manager for z/OS 133  
VTOC of a disk volume 22

## X

XMOUT command 293

## Z

OLIST 1  
PDS 1  
PDSE 1  
SCLM 2

Archived



## Improving Your Productivity with the ISPF Productivity Tool V5.9 on z/OS

(0.5" spine)  
0.475" <-> 0.873"  
250 <-> 459 pages







# Improving Your Productivity with the ISPF Productivity Tool V5.9 on z/OS



**Easy access to ISPF environment objects from any application**

**Enhance your system navigation and search capabilities**

**Point-and-Shoot object names to launch applications**

In this IBM® Redbooks® publication, we introduce the IBM Interactive System Productivity Facility Productivity Tool (ISPF-PT) Version 5, Release 9 for z/OS®. The ISPF Productivity Tool operates as a seamlessly integrated front-end to ISPF:

- ▶ The ISPF-PT functionality is available from any panel without modifying any ISPF Primary Option Menus. All ISPF-PT functions are totally integrated.
- ▶ The ISPF-PT combines separately provided ISPF utility functions and new ISPF Productivity Tool features into the Object List (OLIST) and Member Selection List (MSL).
- ▶ The ISPF-PT relates objects to applications similarly to the way a PC performs OLE (Object Linking and Embedding).
- ▶ The ISPF-PT provides extensive search capabilities that are both rapid and intuitive.
- ▶ The ISPF-PT extends the ISPF Action Bar with options that provide access to new functionality, so that you do not have to learn new commands or syntax.
- ▶ The ISPF-PT provides integrated and enhanced IBM Software Configuration and Library Manager (SCLM) support within the standard member and data set lists.
- ▶ The ISPF-PT includes built-in interfaces to various IBM and ISV products.

Practical scenarios, accompanied by detailed screen captures and coding examples, demonstrate how to take advantage of the ISPF-PT enhanced functionality at every case.

## INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

### BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

**For more information:**  
[ibm.com/redbooks](http://ibm.com/redbooks)