
EZ-ISPFTM

User ' s Guide

Release 1.8
January 11, 2007

Referential Systems, LLC
Vienna, Virginia, USA
Phone: (800) 377-2650
Web: <http://www.ez-ispf.com>

EZ-ISPF™, Copyright 2004, by Referential Systems, LLC. All Rights Reserved.

This document, the user's guide for EZ-ISPF™, is an unpublished work fully protected under United States and International copyright law. Permission is hereby granted, to licensed users of EZ-ISPF™ only, to make a limited number of copies of this document for distribution and use within their organizations. No copies may be made for any other reason, nor may this document or any part of it be reproduced or distributed for any other purpose without the permission of Referential Systems, LLC.

The following trademarks are used within this publication:

z/OS and OS/390 are trademarks of International Business Machines

Table of Contents

Preface	Page 6
About This Guide	Page 6
Organization of This Guide	Page 7
Additional Information/Training Tools	Page 8
Chapter One - Introducing EZ-ISPF	Page 9
What is EZ-ISPF?	Page 9
Background	Page 9
EZ-ISPF Overview	Page 10
EZ-ISPF Invocation Method	Page 10
How EZ-ISPF Improves Productivity	Page 11
Eliminating Keystrokes	Page 11
Minimizing System Interactions	Page 13
Fast Access to Data Set Names	Page 15
Summary	Page 16
Chapter Two - How To Use EZ-ISPF	Page 17
EZ-ISPF Demo Mode	Page 18
EZ-ISPF On-Line Tutorial	Page 19
EZ-ISPF as a "Pop-Up" Application	Page 19
EZ-ISPF In New Screen Split	Page 19
EZ-ISPF Real-World Examples	Page 19
Simple Invocation Technique	Page 22
Target String Processing	Page 22
Default Processing For Target Strings	Page 23
Cataloged Data Set Name	Page 23
Partial Data Set Name	Page 23
Non-Cataloged Data Set	Page 24
PDS Member Name	Page 24
Partial Member Name	Page 24
Program Name	Page 24
Partial Program Name	Page 24
JCL Procedure Name	Page 25
Partial Procedure Name	Page 25
ISPF Panel Name	Page 25
Partial ISPF Panel Name	Page 25
DASD Volume Serial	Page 25
Partial DASD Volume	Page 25
DASD UNIT= Keyword	Page 26
DASD Unit	Page 26
JCL VOL=SER= Keyword	Page 26
SYSOUT Character	Page 26

SYSOUT= Keyword	Page 26
MSGCLASS Character	Page 27
MSGCLASS= Keyword	Page 27
JOB CLASS Character	Page 27
JOB CLASS= Keyword	Page 27
Jobname	Page 27
Comprehensive Target String Processing ...	Page 27
Flexible Target String Processing	Page 28
Invocation Parameters	Page 28
Available EZ-ISPF Invocation Parameters ..	Page 29
Using Invocation Parameters	Page 30
Additional Processing Capabilities	Page 32
Invocation Options	Page 32
More Examples Using Invocation Options ...	Page 35
Example 1 - Compressing a PDS	Page 35
Example 2 - Deleting a Data Set	Page 35
Example 3 - Getting Data Set Info ...	Page 35
Assigned Data Set IDs	Page 35
History List	Page 40
Current PDS Search Set	Page 40
Context PDS Search Set	Page 40
Last Indicated PDS Search Set	Page 41
Panel Display PDSs	Page 43
 Chapter Three - Installing EZ-ISPF	 Page 44
Introduction	Page 44
Step 1 - Define EZ/EZS Commands	Page 45
Step 2 - Install EZ-ISPF Under SMP/E	Page 46
Step 3 - Apply Expiration Date Zap	Page 46
Step 4 - Implement ISPF ELEMENTS	Page 47
Step 4.1 - Using LIBDEF Invocation	Page 47
Step 4.2 - Using Normal Invocation	Page 48
Step 5 - Test EZ-ISPF Installation	Page 49
Step 6 - Additional Considerations	Page 50
 Appendix A - EZ-ISPF Messages	 Page 51
 Appendix B - EZ-ISPF vs. REFLIST	 Page 68
 Appendix C - Create/Update Site Command Table	 Page 70
Step C.1 - Determine Commands To Be Defined ...	Page 70
Step C.2 - Determine if Updating or Creating .	Page 70
Step C.3 - Allocate PDS For Site Table	Page 71
Step C.4 - Copy Existing Site Command Table ...	Page 71
Step C.5 - Create or Update Site Table	Page 72
Step C.6 - Implement Site Command Table	Page 74

Step C.7 - Verify Site Command Table	Page 75
Appendix D - SMP/E Install	Page 77
Step D.1 - SMPALLOC	Page 77
Step D.2 - SMPCSI	Page 77
Step D.3 - SMPDDDEF	Page 78
Step D.4 - SMPRECV	Page 78
Step D.5 - SMPAPPLY	Page 78
Step D.6 - SMPACCEPT	Page 79
Appendix E - Modifying IBM ISPF Panels	Page 80
Step E.1 - Allocate Panel Library	Page 80
Step E.2 - Modify IBM ISPF Panels	Page 80
Appendix F - Modifying Options Table	Page 83
Appendix G - Advantages of Using EZ-ISPF	Page 84
Saves Time/Improves Productivity	Page 85
Intangible Benefits	Page 89
Ease of Use/Training	Page 90
Comparison To Other Tools	Page 91
Summary	Page 93
Appendix H - Cost Justification	Page 94
Appendix I - User Feedback	Page 95

Preface

About This Guide

This is the user's guide for Release 1.8 of EZ-ISPF™, an ISPF productivity improvement tool. This guide will help you to understand the structure and function of EZ-ISPF, and will also show you how using EZ-ISPF will make you much more effective while you are performing your daily tasks with IBM's Interactive System Productivity Facility (ISPF®) product.

If you need additional copies of this guide, it is available in Adobe Acrobat™ format on our web site at www.ez-ispf.com. The Adobe Acrobat™ reader is free and can be downloaded from the Adobe web site at www.adobe.com.

Organization of This Guide

This user's guide for EZ-ISPF™ is organized as follows:

- Chapter One provides an introduction to EZ-ISPF.
- Chapter Two provides more details on how to use EZ-ISPF.
- Chapter Three describes how to install the product.

In addition, there are several appendices:

- Appendix A describes EZ-ISPF messages.
- Appendix B describes differences in EZ-ISPF and the ISPF REFLIST facility.
- Appendix C describes how to create or update the ISPF site command table.
- Appendix D describes how to install the product using SMP/E.
- Appendix E describes how to modify IBM ISPF panels.
- Appendix F describes how to modify the EZ-ISPF options table.
- Appendix G describes the advantages of using EZ-ISPF.
- Appendix H provides cost justification information.
- Appendix I provides user feedback information.

Appendices C, D, E, and F describe optional installation procedures.

You can use the information in Appendices G and H as the basis for a written summary for or a presentation to upper management.

Additional Information/Training Tools

In addition to this user's guide, EZ-ISPF includes a self-guiding on-line demo as well as a complete on-line tutorial.

Once the product is installed, you can invoke the demo, which will show you how to use the most basic features of EZ-ISPF. Each user can invoke the demo using his own data sets and PDS member names. Since the demo actually executes EZ-ISPF as it runs, each individual user can actually see EZ-ISPF in action against his own data sets. This makes for a great training tool.

Please note that the purpose of this user guide is to introduce EZ-ISPF and to provide a general understanding of how EZ-ISPF works. As such, it does not provide a complete description of each and every EZ-ISPF feature and facility. Once the product is installed, you can review the on-line tutorial for a complete description of all EZ-ISPF features and facilities.

Users will be able to learn how to use EZ-ISPF by simply running the on-line demo and then by reviewing the on-line tutorial from time to time. This means that there probably is no need to copy and distribute this user's guide.

Since EZ-ISPF supplements (rather than replacing) ISPF functions, users can continue to use ISPF as they always have. This means there is no "conversion" that immediately affects all users, and users do not have to be heavily trained before EZ-ISPF is installed. Each user can learn and then start benefitting from EZ-ISPF at his own pace and as time permits.

Chapter One - Introducing EZ-ISPF

What is EZ-ISPF?

EZ-ISPF, which is short for Easy Zoom Into ISPF and is pronounced as "Easy ISPF", is an ISPF application that provides significant productivity improvements and time savings to ISPF users on IBM's OS/390 and z/OS systems.

Background

Although a great deal of today's mainframe data processing tasks involve the use of personal computers and software that reside outboard of the mainframe environment, ISPF is still heavily used and is in fact the most effective tool to use when performing many common mainframe programming and operations tasks. ISPF has evolved a great deal since its inception in the 1970s, but many of the tasks ISPF users perform still involve a large amount of typing and command entry. Data set names, JCL, and various menu options and ISPF and TSO/E commands must still be remembered and typed into various screens under ISPF in order to use it effectively.

Unfortunately, the pool of individuals with mainframe programming and operations skills is dwindling rapidly, primarily because many individuals with mainframe skills are beginning to near retirement age, and very few colleges and universities currently offer mainframe-oriented skills training. A lack of new workers with mainframe knowledge has caused many industry analysts to predict a near-term skills shortfall. This trend, documented in a Computerworld¹ article, is expected to continue for the foreseeable future. A dwindling mainframe talent pool forces those individuals who are still working in a mainframe environment to become more productive and to "work smarter".

¹Computerworld, "Mainframe Brain Drain Looms", March 31, 2003,
<http://www.computerworld.com/careertopics/careers/story/0,10801,79849,00.html>

In the z/OS and OS/390 environments, any means that can be found to improve ISPF users' productivity will certainly aid in improving overall job productivity, since ISPF is so heavily used. EZ-ISPF is a tool that provides such a productivity boost. EZ-ISPF was designed and written by MVS software developers with over 25 years experience using ISPF on a daily basis. EZ-ISPF was born from experience gained through thousands of hours using ISPF. ISPF is very powerful, but it does have its shortcomings. EZ-ISPF was born from the need to address those shortcomings in order to vastly improve productive use of ISPF.

EZ-ISPF Overview

EZ-ISPF executes as a traditional ISPF application under any release of OS/390 or z/OS, and provides significant time savings for ISPF users. EZ-ISPF saves time and improves productivity under ISPF in three ways:

- by eliminating ISPF user keystrokes
- by eliminating or minimizing user-to-ISPF system interactions
- by providing fast access to recently or frequently used data set names

How does EZ-ISPF provide each of these benefits? Before we discuss this, a brief explanation of how EZ-ISPF is invoked and operates will provide helpful background information.

EZ-ISPF Invocation Method

EZ-ISPF is invoked under ISPF when a user types the 'EZ' command on the command line of any ISPF panel or presses a PF key that has been equated to the 'EZ' command. The EZ command may be followed by an optional one- or two-character invocation parameter that tells EZ-ISPF which function it should perform. Certain types of invocation parameters may be followed by one or more invocation subparms. Once started, EZ-ISPF immediately determines what type of character string, known as the "target string", the ISPF display screen cursor is on. The type of target string and the context in which that string appears, together with any specified invocation parameter and

invocation subparms, determines the action EZ-ISPF is to take.

If, for example, the ISPF screen cursor is on a cataloged data set name and the ISPF user does *not* type an invocation parameter when EZ-ISPF is invoked, then EZ-ISPF will instantly invoke ISPF Edit for that data set.

If the target string is a PDS member name and the user types invocation parameter 'B' (for Browse) then EZ-ISPF will invoke ISPF Browse for the member.

If the invocation parameter is specified as '1C' and the target string is a data set name, then EZ-ISPF invokes ISPF option 3.1 (ISPF Library Utility) and within 3.1, the 'C' option (Compress data set).

If the target string is a DASD volume serial number and no invocation parameter is specified, then a DASD Information Display will be produced for the specified DASD volume. The DASD Information Display shows DASD characteristics and free space on the specified volume. If the target string is a DASD volume serial number and invocation parameter '4' is specified, then EZ-ISPF will invoke ISPF option 3.4 and the specified DASD volume serial will be shown in the Volume Serial field on the ISPF option 3.4 Data Set List Utility panel.

Now that you have a better understanding of how EZ-ISPF operates, let's use that information to highlight the ways in which EZ-ISPF makes ISPF users more productive.

How EZ-ISPF Improves Productivity

Eliminating Keystrokes

A major feature of EZ-ISPF is its ability to sense the location of the ISPF display screen cursor and to perform different types of processing depending upon the type of character string that the cursor is on. Here's a specific example showing how, using this feature of EZ-ISPF, an ISPF user can save time by eliminating keystrokes:

Suppose an ISPF user is editing a PDS member containing JCL, and he wants to examine one of the data sets contained in the JCL to verify its contents (this is a common task performed by programmers under ISPF). The table which follows shows the sequence of ISPF commands that must be typed in order to accomplish this simple operation, both with and without EZ-ISPF:

	Without EZ-ISPF	With EZ-ISPF
1) Split Screen	Press 'SPLIT' PF key	Move cursor to data set name and press 'EZ' PF key.
2) Invoke ISPF Edit	Type '2' and press ENTER	
3) Type name of data	'SYS2.PROD.DEBITSYS.CYCLE2.JCL'	
4) Invoke Edit	'ENTER' Key	
Keystroke Total:	4 + number of characters in data set name	1

Table 1 - EZ-ISPF Usage Example

As is shown in the table above, one way EZ-ISPF saves time is by eliminating keystrokes. A large part of the keystroke savings comes from the fact that the ISPF user does not have to type the data set name, since EZ-ISPF captures it by sensing the current cursor location on the ISPF display.

For the sake of clarity, the use ISPF command chaining and "copy and paste" techniques were not included in our comparison. Even if they were, the time savings with EZ-ISPF is significant.

Here's another example describing a commonly performed task. Suppose an ISPF user is editing a member in a Partitioned Data Set and then attempts to save his changes, but the PDS is full, so the save fails with a SE37 abend. The table which follows shows how this problem is normally solved, both with and without EZ-ISPF:

	Without EZ-ISPF	With EZ- ISPF
1) Split Screen	Press 'SPLIT' PF key	Type '1C', move cursor to data set name and press 'EZ' PF key.
2) Select ISPF Option 3.1	Type '3.1' and press ENTER	
3) Type name of data set to compress	'PROD.BATCHRUN.JCLLIB'	
4) Invoke Compress	'ENTER' Key	
Keystroke Total:	6 + number of characters in data set name	3

Table 2 - EZ-ISPF Usage Example

As shown in the table above, a significant amount of time is saved by using EZ-ISPF.

Minimizing System Interactions

Without EZ-ISPF, when an ISPF user wants to invoke a second ISPF menu option without leaving the current one and he wants some portion of the current display screen to remain visible, he must use the ISPF "SPLIT" command to start a second ISPF logical screen, then invoke the desired function on the new logical screen. Both of the examples above show this technique at work. In the first case, ISPF Edit was invoked on the new screen. In the second case, the ISPF Library Utility (option 3.1) was invoked.

EZ-ISPF eliminates or reduces user-to-ISPF system interactions by functioning as a **bridge** between commonly used ISPF functions on the same logical screen, without the need to create a second logical screen, and with a minimum number of keystrokes being required to invoke the desired function. Once EZ-ISPF is installed, an ISPF user can invoke the most commonly used ISPF functions directly from the application that is currently in use. This can be seen in the rightmost column of tables one and two above. The current ISPF application is halted when EZ-ISPF starts, and EZ-ISPF then invokes the desired second ISPF function on the same logical screen, "on top of" the current ISPF application. After the ISPF user is finished with the ISPF function which EZ-ISPF invoked, the "END" command will

return him to the application that was active when EZ-ISPF started.

Here's another example of how EZ-ISPF can non-disruptively aid an ISPF user: suppose an ISPF user has invoked Edit to change and then submit JCL for a batch job that will sort some data. He has specified DASD volume TSO005 for the resulting output file but questions whether or not this volume contains enough free space. He also wants to check the list of data sets already residing on TSO005. He can get an instant answer to his question about the amount of free space on DASD volume TSO005 by placing the cursor on volser TSO005 in his job's JCL and invoking EZ-ISPF. EZ-ISPF will determine that the screen cursor is on a DASD volume serial and will immediately produce a DASD Information Display for that volume. If the target string is a volume serial prefix (e.g., TSO*), then EZ-ISPF will instead show a DASD Information Display for all DASD volumes whose serial numbers match that prefix.

The ISPF user can then press the END PF key to return to ISPF edit, or, to obtain a list of the data sets on a particular DASD volume, he can type '4' on the ISPF command line, place the cursor on a DASD volume serial, and invoke EZ-ISPF. EZ-ISPF will then bridge to ISPF option 3.4 and the resulting Data Set List Utility panel will show the specified DASD volume serial in the Volume Serial field. To see a list of all data sets on the chosen DASD volume, the user can simply press Enter after making sure the Dsname Level field is blank. The End PF key can then be used to return to ISPF Edit.

The example above shows once again that EZ-ISPF provided the information that was needed, without having to split the screen and without having to type the DASD volume serial. Both keystrokes and system interactions were eliminated through the use of EZ-ISPF.

Below is a **partial list** of the ISPF options and EZ-ISPF functions currently supported by EZ-ISPF together with the EZ-ISPF invocation parameter used to invoke each one. The ISPF user simply types the desired invocation parameter and any needed invocation subparms and presses the "EZ" PF key to have EZ-ISPF non-disruptively invoke the desired ISPF function ***on top of*** the current ISPF application.

ISPF Option or EZ-ISPF Function	EZ-ISPF Invocation Parameter
1 (Browse)	B
1 (View)	V
2 (Edit)	E (default)
3.1 (Library Utility)	1
3.2 (Data Set Utility)	2
3.3 (Move/Copy Utility)	3
3.4 (Data Set List Utility)	4
SDSF Log Function	\$ LOG
SDSF Hold Queue	\$ H
Submit a job	M
Produce DASD Information Display	D

Table 3 - Some of the available EZ-ISPF invocation parameters

Fast Access to Data Set Names

EZ-ISPF's ability to capture and save the string that the ISPF display screen cursor is on allows it to capture and remember the data set names for which EZ-ISPF is invoked. When EZ-ISPF is invoked and the target string is a data set name, it adds the data set name under the cursor to its 'history' list. The history list contains information about the last 32 times EZ-ISPF was invoked. Since most ISPF users tend to spend their time using the same small group of data sets day after day, EZ-ISPF's history list is of great value. The EZ-ISPF user can display the history list using invocation parameter S (for Show Miscellaneous Information) with invocation subparm H (for History), as shown below:

S H

Once the history list is displayed, the user can place the cursor on any data set name in the list and invoke EZ-ISPF recursively to utilize the most commonly used ISPF functions against that data set. The need to remember the data set name or locate a data set name through research (i.e., split screen, ISPF 3.4) is eliminated and time is saved for more important work.

In addition to remembering data set names for later recall, EZ-ISPF also allows the user to assign a "data set ID" of

up to eight characters to each frequently used data set name. This feature allows the ISPF user to process a frequently used data set by the assigned data set id, rather than by trying to remember and type the full data set name, which may be quite lengthy.

EZ-ISPF allows a user to assign a data set id by placing the cursor under any cataloged data set name on any ISPF display screen and invoking EZ-ISPF with invocation parameter A, as shown below:

A iiiiii

where "iiiiiii" is an invocation subparm specifying the unique 1- to 8-character data set id to assign to that particular data set. The data set id can consist of any combination of alphanumeric and special characters (#,\$,@).

For a data set name like 'SYS2.PAYROLL.REL14.JCL', a user might assign a data set ID like 'P14J'.

Once a data set ID is assigned, EZ-ISPF can be invoked against the data set by typing just the data set ID, and not the full data set name. EZ-ISPF's list of assigned data set IDs and their corresponding data set names can easily be recalled and viewed at any time.

Using the data set ID instead of the actual data set name is another way EZ-ISPF saves keystrokes.

Summary

EZ-ISPF is an easy-to-use tool that provides significant productivity improvements for programmers and operators under ISPF. Through its extended awareness of the ISPF screen cursor location and its knowledge of the types of target strings the cursor may be on, EZ-ISPF is able to provide a bridge to rapidly invoke the most frequently used ISPF functions and menu options. A simple and easy-to-remember set of invocation parameters adds even more flexibility, permitting experienced ISPF users to rapidly improve their productivity under ISPF. EZ-ISPF invocation parameters and options can be quickly learned, ensuring that users of the product rapidly become experts in its use.

Chapter Two - How To Use EZ-ISPF

In this chapter, you will learn how to use EZ-ISPF effectively. EZ-ISPF offers flexible, comprehensive, automatic, context-driven default processing for more types of target strings than any other ISPF productivity tool.

EZ-ISPF offers a set of comprehensive one or two character invocation parameters and many of these "mirror" existing ISPF options, so you do not need to learn a completely new command set to use EZ-ISPF effectively.

EZ-ISPF offers invocation options which provide for immediate processing (saving additional system interactions). These invocation options "mirror" existing ISPF panel options, so they are easy to remember.

The use of invocation parameters and options which "mirror" existing ISPF facilities means that EZ-ISPF supplements ISPF rather than replacing parts of it. You speed up your work with ISPF without having to learn a new methodology and without having to repeatedly type a new set of complicated commands.

When you access a data set or a member which you are not allowed to update due to security restrictions, ISPF warns you about this only after you change the item and then attempt to save it. EZ-ISPF warns you about such security restrictions as soon as you access a restricted data set or member. This keeps you from wasting time typing in changes to an item you are not allowed to update.

EZ-ISPF offers additional innovative productivity and training features, including:

- support for assigning data set ID's to frequently used data sets
- ability to list frequently used data sets
- a history list showing context in which target string appears
- support for immediate job (JCL member) submission without

- typing data set name
- simple invocation of SDSF with jobname as optional target string
 - flexible determination of the PDS(s) to associate with a specified PDS member name
 - a built-in display of DASD information and device characteristics

EZ-ISPF is comprehensive (it addresses most aspects of ISPF), it supplements rather than replaces ISPF facilities, it is extremely easy to learn, and it has more innovative productivity features than any other ISPF productivity tool. These productivity features have a tremendous impact on the speed and efficiency of ISPF. These factors make EZ-ISPF the ISPF productivity tool of choice.

EZ-ISPF includes two features that facilitate effective use of the product: an online 'demo' mode, which uses actual data set names from your system to guide you through an overview of EZ-ISPF's capabilities, and a built-in online tutorial.

We will now show you how to use each one of EZ-ISPF's features and facilities effectively.

EZ-ISPF Demo Mode

You can invoke the EZ-ISPF demo with invocation parameter S(for Show Miscellaneous Information) and invocation subparm D (for Demo), as shown below:

S D

The demo guides you through a set of simple steps and examples that help you to understand what EZ-ISPF is and how it can save you time as you use ISPF. The EZ-ISPF demo is restartable; that is, if you have to leave the demo in the middle of viewing it, when you next invoke the EZ-ISPF demo, you will have the option of picking up where you left off, or starting over from the beginning.

Viewing the EZ-ISPF demo should only take you between 30 and 45 minutes; this is time well spent and will give you a rapid leap forward on the road to using EZ-ISPF effectively.

EZ-ISPF On-Line Tutorial

EZ-ISPF also contains a built-in online tutorial that you can access at any time. The tutorial provides a description of each one of EZ-ISPF's powerful features. You can access the on-line tutorial by specifying invocation parameter H (for Help) when you invoke EZ-ISPF.

EZ-ISPF as a "Pop-Up" Application

When EZ-ISPF is invoked, it "pops-up" over the top of the current application and panel. When EZ-ISPF is terminated, the original application and panel are restored. This feature allows EZ-ISPF to be invoked without interrupting the work that was in progress.

EZ-ISPF In New Screen Split

If the person who installed EZ-ISPF at your site chooses to implement the 'Ezs' command, then in addition to invoking the 'EZ' command under ISPF to start EZ-ISPF as a "pop-up" application, you can also use the 'Ezs' command to invoke EZ-ISPF by itself in a new ISPF logical screen split. You might want to use 'Ezs' to invoke EZ-ISPF when you want it to stay active so you can swap back and forth to it.

EZ-ISPF Real-World Examples

As a mainframe programmer or operator, you are well aware that ISPF is still the main method for getting programming and operations work accomplished on an IBM mainframe computer under the OS/390 and z/OS operating systems. However, ISPF has a number of time-consuming deficiencies:

- it requires too much typing (for data set names, PDS member names, etc.)
- it requires too many system interactions to get certain tasks accomplished
- it sometimes requires a complicated command chaining specification to transfer to a certain option

As examples of the first two deficiencies above, suppose you

are browsing some JCL containing the line below:

```
//SYSUT1 DD DSN=SYS2.ACCTAB03.PAYROLL.CONTRL(SUMRPT4),DISP=SHR
```

and you decide that you want to Edit member SUMRPT4 in data set SYS2.ACCTAB03.PAYROLL.CONTRL. Now suppose you have not referenced this data set recently and do not have it in a personal ISPF data set reference list (REFLIST).

This is a common situation that programmers and operators frequently encounter.

Assuming you do not want to write down or remember this rather complicated data set and member name combination, you would need to do the following:

- split the screen
- possibly scroll the original display so the data set name is visible after the split
- type 2 in the command line in the new screen split and press Enter
- type 'SYS2.ACCTAB03.PAYROLL.CONTRL(SUMRPT4)' in the data set name field of the Edit menu panel and press Enter

If you are logged on to TSO using a TN3270 terminal emulator like Attachmate®, Hoblink®, IBM PCOMM®, or Vista® under Microsoft Windows, you can highlight the desired data set name with your mouse and Cut and Paste the name, but you still need to use the ISPF SPLIT command and spend time Cut'ing and Paste'ing. This task, with or without a mouse, requires 3 or more system interactions and up to 40 keystrokes.

You can accomplish this same task using EZ-ISPF with a single keystroke! Place the cursor on member name SUMRPT4 in the JCL line above and press the 'EZ' PF key, and EZ-ISPF automatically invokes ISPF edit for this member in data set SYS2.ACCTAB03.PAYROLL.CONTRL.

As an example of the third ISPF deficiency above (it sometimes requires a complicated command chaining specification to transfer to a certain option) suppose you are editing some JCL containing the job card shown below:

```
//RSC002W JOB (PJFS1234),'ASSEM2',CLASS=X,NOTIFY=RSC002
```

Now suppose that:

- you have just submitted the JCL stream containing the above JOB statement for execution
- this job runs so quickly it completes immediately
- you want to use SDSF to check the result of the job
- you already have a number of jobs on the SDSF held output queue; to avoid having to scroll the list of held jobs when you get to SDSF, you want to restrict the list to jobs with job name RSC002W
- you invoke SDSF on your system by using option M off of the main ISPF menu, followed by option 5

Again, this is a common situation that programmers frequently encounter. In fact, a programmer in a heavy testing environment might encounter this situation dozens of times in a single day.

In the above mentioned situation, you could press Enter after typing the following command chaining specification:

```
=M.5;H RSC002W
```

This would get you immediately to the portion of the SDSF held output list containing job RSC002W, but requires a somewhat complicated command chaining specification and quite a few keystrokes.

You can accomplish this same task using EZ-ISPF with two keystrokes by typing invocation parameter '\$' on the command line, placing the cursor on the jobname in the JOB statement shown above, and then pressing the 'EZ' PF key.

The time wasted on the type of repetitive tasks described above, even for an installation with a relatively small number of programmers, adds up to a significant amount of lost productivity. For this reason, IBM has made a few productivity enhancements to ISPF over the years, a number of ISPF productivity tools been developed by independent software vendors, and quite a few installations have developed their own in-house tools to deal with this problem.

However, none of these products or approaches have been particularly successful. There are several reasons for this. Some of these products are not comprehensive enough. They address only a limited portion of ISPF (like Edit or Browse). Some completely replace significant portions of

ISPF. This means they require too much retraining and do not allow you to take advantage of future ISPF enhancements. Some rely on techniques that require additional system interactions (which means they are slow). An example of this is the ISPF enhancement involving data set reference lists. To use a list, you must open and then process the list. This requires additional system interactions.

As you will see in the pages that follow, EZ-ISPF is comprehensive (it addresses most aspects of ISPF), it supplements rather than replacing ISPF, it is extremely easy to learn, and it has innovative productivity features that none of the other products or approaches have. These productivity features have a tremendous impact on the speed and efficiency of ISPF, all of which makes EZ-ISPF the ISPF productivity product of choice.

Let's now examine some of EZ-ISPF's key features and how you can use them to improve your ISPF productivity.

Simple Invocation Technique

EZ-ISPF is invoked with a single ISPF command which is generally equated to a PF key. This means that you can invoke EZ-ISPF, from anywhere within ISPF, with a single keystroke. You can also type 'EZ' on the command line of any ISPF panel to invoke EZ-ISPF. EZ-ISPF can be invoked anywhere in ISPF; its use is not limited by the type of ISPF application that is active when EZ-ISPF starts.

Target String Processing

When you place the cursor under a character string on any ISPF panel and invoke EZ-ISPF, EZ-ISPF automatically processes the target string indicated by the position of the cursor. EZ-ISPF automatically determines the type of target string to be processed from the format of the string and the context in which that target string appears on the screen. This feature of EZ-ISPF allows you to process an object like a data set name and/or a PDS member name shown on any ISPF panel without having to actually re-type that data set name or member name.

Default Processing For Target Strings

EZ-ISPF offers default processing for each different type of target string which it recognizes. The default processing for each different type of target string is the type of processing which would normally be required by an ISPF user. If you want the default processing for a given target string, all you have to do is place the cursor under the target string on the ISPF display screen and press the PF key which invokes EZ-ISPF. This means that you can often get the processing you need with a single keystroke.

Below is a **partial list** of the types of target strings recognized by EZ-ISPF showing the default processing provided for each one:

Cataloged Data Set Name

When the target string is a cataloged data set name such as 'SYS2.PAYROLL.R15.SOURCE', EZ-ISPF invokes ISPF Edit against the data set name. This means that to edit a data set when EZ-ISPF is installed, you just put the cursor on the data set name on the ISPF screen and press the 'EZ' PF key.

Partial Data Set Name

When the target string is a partial data set name (i.e., the prefix or high-level portion of a cataloged data set name, such as 'SYS2.PAYROLL'), EZ-ISPF invokes the ISPF Data Set List Service, which shows all cataloged data sets which match the Dsname Level specified by the partial data set name.

Non-Cataloged Data Set

EZ-ISPF handles target strings that are non-cataloged data set names by invoking ISPF option 3.2. The resulting Data Set Utility panel contains an 'A' in the Option field along with a message indicating that the data set is not cataloged. You can then press Enter to allocate the data set or change the Option to 'C' to catalog the data set.

PDS Member Name

When the target string is a PDS member name, EZ-ISPF invokes ISPF Edit against the member in the PDS determined by the Context PDS Search Set, the Last Indicated PDS Search Set, or the Panel Display PDS(s). These are all techniques which are used to determine the PDS to search for a given member name and are described later in this chapter.

Partial Member Name

A partial member name is a member name that ends in an asterisk, like 'ABC*'. For this type of target string, EZ-ISPF will invoke ISPF Edit, which will in turn provide a member list containing only the members whose names match that prefix.

Program Name

When the target string is a program name on a JCL EXEC statement, such as PGM=PAY1, then EZ-ISPF invokes ISPF Browse for the load module in the library from which it is to be executed. This can be from a JOBLIB, STEPLIB, or the system linklist.

Partial Program Name

When the target string is a partial program name such as PGM=PAY*, EZ-ISPF will produce an ISPF Browse list of matching member names in the active execution library (JOBLIB, STEPLIB, Link List, etc.).

JCL Procedure Name

When the target string is a JCL procedure name specified on a JCL EXEC statement, such as EXEC ASM15, EZ-ISPF invokes ISPF Edit for the procedure in the library where the JCL procedure resides; that is, the JCL library from which JES will fetch the JCL procedure in order to execute it (the system proclib, a JCLLIB, etc.).

Partial Procedure Name

EZ-ISPF's default processing when the target string is a partial JCL procedure name (EXEC ASM*, for example) is to invoke ISPF Edit. ISPF Edit will produce a list of matching member names from the active proc library (system proclib, JCLLIB, etc.).

ISPF Panel Name

If the cursor is on a target string that EZ-ISPF recognizes as an ISPF panel name, then EZ-ISPF searches the same set of PDS's that ISPF searches to locate the panel definition. If the panel is found, ISPF EDIT is invoked against the panel.

Partial ISPF Panel Name

If a partial panel name is the target string, processing is just like that mentioned for a full panel name, except that the search is for the specified panel name prefix rather than for the complete panel name.

DASD Volume Serial

For a DASD volser target string appearing in JCL, such as VOL=SER=TST001, EZ-ISPF will show the DASD Information Display for the specified DASD volume.

Partial DASD Volume

If the target string is a partial DASD volser ending in an asterisk such as VOL=SER=TST*, EZ-ISPF will show the DASD

Information Display for all DASD volumes whose volsers match the given volser prefix.

DASD UNIT= Keyword

If the target string is the JCL UNIT= keyword, then EZ-ISPF will present the DASD Information Display for all online DASD volumes attached to the system on which EZ-ISPF is invoked.

DASD Unit

When the target string is a DASD unit (UNIT=3390 for example), then EZ-ISPF's DASD Information Display will be shown and include a line for each DASD volume of that unit type. The UNIT= specification can be a generic or esoteric unit name such as 3390 or SYSDA.

JCL VOL=SER= Keyword

If the target string is the JCL keyword VOL=SER=, then EZ-ISPF will present the DASD Information Display for all online DASD volumes attached to the system on which EZ-ISPF is invoked.

SYSOUT Character

A target string consisting of the value specified for a JCL SYSOUT= keyword, such as SYSOUT=A, causes EZ-ISPF to display the JES SYSOUT class characteristics for the specified SYSOUT class. The cursor must be on the actual class value specified by the SYSOUT= keyword.

SYSOUT= Keyword

If the target string is the JCL SYSOUT= keyword, then information for all of the active JES SYSOUT classes will be displayed.

MSGCLASS Character

A target string consisting of the value specified for a JCL MSGCLASS= keyword, such as MSGCLASS=A, causes EZ-ISPF to display the JES SYSOUT class characteristics for the specified message class. The cursor must be on the actual class value specified by the MSGCLASS= keyword.

MSGCLASS= Keyword

If the target string is the JCL MSGCLASS= keyword, then information for all of the active JES SYSOUT class characteristics will be displayed.

JOB CLASS Character

When the target string is the value specified for the CLASS= keyword on a JCL JOB statement, then EZ-ISPF will display the JES job class characteristics for the specified job class. The cursor must be on the actual class value specified by the CLASS= keyword.

JOB CLASS= Keyword

If the target string is the CLASS= keyword on a JCL JOB statement, then EZ-ISPF will display the JES job class characteristics for all job classes.

Jobname

If the cursor is on the jobname specified on a JCL JOB statement, such as the 'RSC002W' value in the statement '//RSC002W JOB...', then EZ-ISPF will invoke the SDSF Held Output Display for the specified job.

Comprehensive Target String Processing

EZ-ISPF provides comprehensive processing for target strings. For example, a target string like a cataloged data set name can be processed by ISPF Edit, ISPF Browse, ISPF

View, ISPF options 3.1, 3.2, 3.3, 3.4, 3.5, 3.12, 3.13, 3.14, and 3.15. As another example, a target string like a DASD volume serial can be used to generate a DASD Information Display for that specific volume or to display the data sets on that specific volume.

Flexible Target String Processing

EZ-ISPF is flexible in its processing of target strings; a target string can appear in any portion of any ISPF panel that is displayed as output by ISPF.

As an example of EZ-ISPF target string flexibility, in a JCL line like that shown below:

```
//SYSUT1 DD DSN=SYS2.PAYROLL.R15.SOURCE(PAYRLL1),DISP=SHR
```

you can process just the data set name, just the PDS member name, or both the data set name and the member name together.

EZ-ISPF even allows you to process the member name in a line like that shown above with a different PDS than the one shown on this line. This type of situation comes up quite frequently. For example, suppose, while browsing this line, you need to know if program PAYRLL1 existed in the previous release of the payroll system. So you want to browse for member PAYRLL1 in PDS SYS2.PAYROLL.R14.SOURCE. With EZ-ISPF, you can use PAYRLL1 in the line above as the target string and specify some other PDS as the one to be searched for this member.

As another example of EZ-ISPF target string flexibility, in a list like that shown below:

```
WORK01    WORK02    WORK03
```

you can treat each target string in this list as if it were a PDS member name or as if it were a DASD volume serial.

Invocation Parameters

When you want something other than the default processing for a given target string, EZ-ISPF provides a simple set of one or two character invocation parameters that you can use

to dictate the processing to be done. To specify an invocation parameter, you simply type the invocation parameter on the command line before invoking EZ-ISPF. When an invocation parameter is specified and the cursor is placed on a target string, EZ-ISPF uses the invocation parameter and the target string to determine what processing is to be performed.

For example, the invocation parameter used to request ISPF Browse processing is B. So, if you are reviewing some JCL containing the following line:

```
//SYSUT1 DD DSN=SYS2.PAYROLL.R15.SOURCE,DISP=SHR
```

and you want to browse data set SYS2.PAYROLL.R15.SOURCE, all you have to do is type B in the command line, place the cursor under data set name SYS2.PAYROLL.R15.SOURCE in the JCL line above, and press the PF key which invokes EZ-ISPF.

EZ-ISPF's invocation parameters let you override its default processing for a particular target string. If you know a target string is a DASD volser, for example, you can force EZ-ISPF to treat it as such by typing 'D' on the ISPF command line (the invocation parameter which requests the DASD Information Display).

Available EZ-ISPF Invocation Parameters

Here is a list of the EZ-ISPF invocation parameters:

- A Assign data set ID
- B Request ISPF Browse
- C Show DASD device characteristics
- D Show DASD Information Display
- E Request ISPF Edit
- F Define 'From' data set and/or member for copy and compare operations
- H Display EZ-ISPF On-Line Tutorial
- L Define Last Indicated PDS Search Set
- M Submit job to JES
- N Nullify Last Indicated PDS Search Set
- O Define Last Indicated PDS Search Set as 'Only' Search Set
- S Show Miscellaneous Information
- T Define 'To' data set and/or member for copy, compare,

```
    create, and replace operations
V  Request ISPF View
1  Request ISPF Option 3.1  (Library Utility)
2  Request ISPF Option 3.2  (Data Set Utility)
3  Request ISPF Option 3.3  (Move/Copy Utility)
4  Request ISPF Option 3.4  (Data Set List Utility)
5  Request ISPF Option 3.5  (Reset ISPF Statistics)
8  Request ISPF Option 3.8  (Outlist Utility)
12 Request ISPF Option 3.12 (SuperC Compare Utility)
13 Request ISPF Option 3.13 (SuperCE Compare Utility)
14 Request ISPF Option 3.14 (Search-For Utility)
15 Request ISPF Option 3.15 (Extended Search-For)
$  Invoke SDSF Display
```

Using Invocation Parameters

If you want to browse, edit, or view a target string, you specify invocation parameter B, E (not normally needed as it is the default), or V.

If you want to use ISPF Option 3.1 to process a target string, you specify invocation parameter 1; if you want to use ISPF Option 3.2 to process a target string, you specify invocation parameter 2, etc.

For invocation parameters associated with ISPF options 3.1, 3.2, etc., EZ-ISPF invokes the corresponding ISPF option and inserts the target string into the appropriate field on the corresponding ISPF panel. For example, a target string in the form of a data set name will generally be shown in the "Other Data Set Name Field"; a target string in the form of a PDS member name in "dsn(mbr)" format will generally be shown in "dsn(mbr)" format in the "Other Data Set Name Field", so you can process the member name in conjunction with the data set. A target string in the form of a standalone member name will generally be shown in the "Member Name" field; a target string in the form of a volume serial will be shown in the "Volume" field of the ISPF Option 3.2 panel, etc.

When you get to the requested ISPF Option, you can then apply any of the associated processing options to the specified target string.

For example, if you are reviewing some JCL containing the

following statement:

```
//SYSUT1 DD DSN=SYS2.PAYROLL.R15.SOURCE,DISP=SHR
```

and you want to process data set SYS2.PAYROLL.R15.SOURCE using the ISPF Option 3.1 Library Utility, all you have to do is type 1 in the command line, place the cursor under data set name SYS2.PAYROLL.R15.SOURCE in the JCL line above, and press the PF key which invokes EZ-ISPF. This will take you to ISPF Option 3.1 and the Library Utility panel will show 'SYS2.PAYROLL.R15.SOURCE' in the "Other Data Set Name" field. You can then apply any needed Library Utility processing to this data set. For example, if you want to get a Library Utility member list for this data set, you press Enter. If you want to compress this data set, you type C in the Option field and press Enter.

As another example, if you are reviewing some JCL containing the following statement:

```
//SYSUT1 DD DSN=SYS2.PAYROLL.R15.SOURCE(PAYONE),DISP=SHR
```

and you want to change member name PAYONE in SYS2.PAYROLL.R15.SOURCE to member name PAY1, all you have to do is type 1 in the command line, place the cursor under member name PAYONE in the JCL line above, and press the PF key which invokes EZ-ISPF. This will take you to ISPF Option 3.1 and the Library Utility panel will show 'SYS2.PAYROLL.R15.SOURCE(PAYONE)' in the "Other Data Set Name" field. You then type R (for rename) in the Option field, type the new member name in the "New Name" field, and press Enter.

As you can see from the examples above, if you are already familiar with the ISPF utility functions, using the EZ-ISPF invocation parameters shown above requires little or no additional training.

In addition, EZ-ISPF invocation parameters are simple to type and simple to remember, and they save you keystrokes each and every time you have to dictate the processing to be done.

EZ-ISPF's method of utilizing on-screen target strings and simple, one or two character invocation parameters allows EZ-ISPF to improve ISPF user productivity, and thereby

reduces the amount of typing and the number of system interactions required. The speed with which you can accomplish work under ISPF is greatly enhanced.

Additional Processing Capabilities

In addition to processing options that invoke ISPF menu options directly, EZ-ISPF also supports additional parameters that cause it to perform its own processing directly. The following list shows some of these additional EZ-ISPF invocation parameters:

- C Show DASD Device Characteristics
- D Request DASD Information Display
- F Define 'From' Data Set and/or Member
- M Submit Job
- T Define 'To' Data Set and/or Member

Invocation parameter C will cause EZ-ISPF to display DASD device characteristics (cylinders/device, tracks/cylinder, optimum block sizes, etc.) for the specified DASD device type.

Invocation parameter D requests the EZ-ISPF DASD Information Display which, as stated earlier, shows the amount of free space and other DASD information for specified DASD volumes.

Invocation parameters F and T define data set and/or member names to be used with subsequent copy, create, replace, or compare operations.

Invocation parameter M allows you to submit a job without typing the full data set name and member name containing the execution JCL.

Invocation Options

EZ-ISPF also supports various invocation options which are appended to the invocation parameter to which they apply. EZ-ISPF automatically processes the specified invocation option as if it were typed into the Option field of the requested ISPF panel and "Enter" were pressed. This feature allows you to request immediate processing for certain ISPF options without waiting for the associated ISPF panel to be

displayed. The associated ISPF panel will not be displayed when the processing is requested or when it is completed; this saves at least one and sometimes two system interactions! The invocation options are exactly the same as the ISPF menu options defined for the ISPF function that EZ-ISPF invokes.

Here's an example that shows how EZ-ISPF invocation options work: if you press the PF key which invokes EZ-ISPF after typing invocation parameter 1 in the ISPF command line and place the cursor under the cataloged PDS data set name in the SYSOUT line below:

```
IEC031I B37-04,.....,SYS2.PAYROLL.SOURCE,...
```

EZ-ISPF will invoke ISPF Option 3.1 (Library Utility) with 'SYS2.PAYROLL.SOURCE' shown in the "Other Data Set Name" field. If you want to compress SYS2.PAYROLL.SOURCE, all you have to do then is type 'C' (for Compress) in the Option field and press ENTER, and ISPF will compress the 'SYS2.PAYROLL.SOURCE' data set.

You can speed things up further by using the 'C' invocation option with invocation parameter 1. Just type '1C' in the ISPF command line, place the the cursor under the data set name, and then press the PF key which invokes EZ-ISPF.

EZ-ISPF will again invoke the ISPF Option 3.1 Library Utility, but this time the 3.1 utility will compress the 'SYS2.PAYROLL.SOURCE' immediately, and the ISPF Option 3.1 Library Utility panel will not even be displayed. In this way, an additional system interaction is eliminated.

Most of the suboptions associated with the ISPF 3.x menus can be used as EZ-ISPF invocation options.

EZ-ISPF also supports use of the 'blank' menu option available for certain ISPF options like 3.1, 3.2, 3.4 and 3.8. As an example of this, here is a portion of the ISPF 3.1 Library Utility panel options list:

blank	Display member list	I	Data set information
C	Compress data set	S	Short data set information
X	Print index listing	E	Edit member

As shown above, if you press ENTER on the ISPF Option 3.1 Library Utility panel and leave the Option field blank, a Library Utility member list will be displayed.

To facilitate using a blank as an invocation option, EZ-ISPF treats invocation option K as a blank.

Suppose you are reviewing some JCL containing the following line:

```
//SYSIN DD DSN=SYS2.PAYROLL.SOURCE,DISP=SHR
```

and you want to get an ISPF Option 3.1 Library Utility member list for the cataloged PDS data set name shown on this line.

To do this without specifying an invocation option, you could press the PF key which invokes EZ-ISPF after typing invocation parameter 1 in the command line and placing the cursor under the cataloged data set name in the JCL line above. The ISPF Option 3.1 Library Utility panel will then be displayed with 'SYS2.PAYROLL.SOURCE' shown in the "Other Data Set Name" field. If you leave the Option field blank and press Enter, a Library Utility member list will be produced for this data set.

As an example of using invocation parameter 1 with invocation option K, if you type invocation parameter 1 in the command line with invocation option K appended to it, as shown below:

1K

and press the PF key which invokes EZ-ISPF after placing the cursor under the cataloged data set name in the JCL line below:

```
//SYSIN DD DSN=SYS2.PAYROLL.SOURCE,DISP=SHR
```

the ISPF Option 3.1 Library Utility panel will be processed as if Enter was pressed after typing 'SYS2.PAYROLL.SOURCE' in the "Other Data Set Name" field (and leaving the Option field blank). This will produce a Library Utility member list for the specified data set and the ISPF Option 3.1 Library Utility panel will not even be displayed. The ISPF Option 3.1 Library Utility panel will not even be displayed

(as it normally is) when you exit out of the Library Utility member list. So, in this case, you save two system interactions.

More Examples Using Invocation Options

Below are examples of using EZ-ISPF invocation parameters and invocation options together to accomplish everyday ISPF tasks.

Example 1 - Compressing a PDS

If you want to invoke EZ-ISPF and request that a data set be compressed, then type 1C on the ISPF command line ('1' for ISPF option 3.1 and 'C' for 'Compress'), put the cursor on the data set name, and press the PF key equated to the 'EZ' command.

Example 2 - Deleting a Data Set

If you want to invoke EZ-ISPF and request that a data set be deleted, then type 2D on the ISPF command line ('2' for ISPF option 3.2 and 'D' for 'Delete'), put the cursor on the data set name, and press the PF key equated to the 'EZ' command.

Example 3 - Getting Data Set Info

If you want the ISPF 'Data Set Information' display for a data set, type 2K on the ISPF command line ('2' for ISPF option 3.2 and 'K' to represent a blank), put the cursor on the data set name, and press the PF key equated to the 'EZ' command.

Assigned Data Set IDs

EZ-ISPF allows you to assign a data set ID consisting of up to eight characters to each one of your frequently used data sets. You can then process the associated data set without typing the entire data set name, even when it is not currently on display, by typing the assigned data set ID following the invocation parameter which dictates the

processing to be done for that data set.

The data set ID assigned to a given data set can consist of from one to eight alphanumeric or national characters, as long as the data set ID is unique.

For example, if you do most of your work in a payroll system with data sets like the following:

```
SYS2.PAYROLL.R14.SOURCE
SYS2.PAYROLL.R14.LINKLIB
SYS2.PAYROLL.R14.PANELS
SYS2.PAYROLL.R14.MACROS
SYS2.PAYROLL.R14.JCL
```

You might then assign EZ-ISPF data set IDs like those below:

Data Set Name	Assigned Data Set ID
SYS2.PAYROLL.R14.SOURCE	P14S
SYS2.PAYROLL.R14.LINKLIB	P14L
SYS2.PAYROLL.R14.PANELS	P14P
SYS2.PAYROLL.R14.MACROS	P14M
SYS2.PAYROLL.R14.JCL	P14J

Here is a list of EZ-ISPF invocation parameters which can be specified along with an assigned data set ID:

B	Request ISPF Browse
E	Request ISPF Edit
F	Define 'From' Data Set and/or Member
M	Submit Job
T	Define 'To' Data Set and/or Member
V	Request ISPF View
1	Request ISPF Option 3.1 (Library Utility)
2	Request ISPF Option 3.2 (Data Set Utility)
3	Request ISPF Option 3.3 (Move/Copy Utility)
4	Request ISPF Option 3.4 (Data Set List Utility)
5	Request ISPF Option 3.5 (Reset ISPF Statistics)
8	Request ISPF Option 3.8 (Outlist Utility)
12	Request ISPF Option 3.12 (SuperC Compare Utility)
13	Request ISPF Option 3.13 (SuperCE Compare Utility)
14	Request ISPF Option 3.14 (Search-For Utility)
15	Request ISPF Option 3.15 (Extended Search-For)

You use these EZ-ISPF invocation parameters to process an

assigned data set ID in just the same way you would a target string in the form of a data set name. The only difference is typing the assigned data set id rather than placing the cursor under the target string.

The examples that follow all assume that data set id S14 has been assigned to SYS2.PAYROLL.R14.SOURCE.

If you want to browse SYS2.PAYROLL.R14.SOURCE, you type:

B S14

in the ISPF command line and press the PF key which invokes EZ-ISPF.

If you want to compress SYS2.PAYROLL.R14.SOURCE, you type:

1 S14

in the ISPF command line and press the PF key which invokes EZ-ISPF. You then type C in the option field of the ISPF Option 3.1 Library Utility panel and press Enter.

If you want to compress SYS2.PAYROLL.R14.SOURCE using an invocation option, you type:

1C S14

in the ISPF command line and press the PF key which invokes EZ-ISPF. The specified data set will be immediately compressed without the ISPF Option 3.1 Library Utility panel being displayed.

You can also follow an assigned data set ID associated with a PDS with a full or partial PDS member name contained in that PDS.

For example, to edit member PAYONE in SYS2.PAYROLL.R14.SOURCE, you type:

E S14 PAYONE

in the ISPF command line and press the PF key which invokes EZ-ISPF.

You can also specify a target string in the form of a PDS

member name in conjunction with an assigned data set ID.

For example, if you are browsing some documentation under ISPF which contains the following line:

```
PAYONE is the first payroll program.
```

then you could browse PAYONE in SYS2.PAYROLL.R14.SOURCE by typing:

B S14

in the command line, placing the cursor under PAYONE in the line above, and pressing the PF key which invokes EZ-ISPF. You could also type:

B S14 PAYONE

and press the 'EZ' PF key.

As another example, if you are browsing the following JCL line:

```
//SYSUT1 DD DSN=SYS2.PAYROLL.R15.SOURCE(PAYONE),DISP=SHR
```

and you want to see if PAYONE existed in Release 14 of the payroll system, you can type:

B S14

in the ISPF command line, place the cursor under PAYONE in the JCL line shown above, and press the PF key which invokes EZ-ISPF. In this case, the specified assigned data set id overrides the "context" PDS SYS2.PAYROLL.R15.SOURCE and you end up browsing PAYONE in SYS2.PAYROLL.R14.SOURCE.

Now suppose you have the JCL required to reassemble and link program PAY1 stored in member PAY1 in PDS SYS2.PAYROLL.R14.JCL which is assigned to data set ID J14. You can then submit the JCL required to reassemble and link program PAY1 by typing:

M J14 PAY1

in the ISPF command line and pressing the PF key which invokes EZ-ISPF.

You can also assign a data set ID to a **partial** data set name. Given the data sets associated with Release 14 of the Payroll system outlined below:

SYS2.PAYROLL.R14.SOURCE	S14
SYS2.PAYROLL.R14.LINKLIB	L14
SYS2.PAYROLL.R14.PANELS	P14
SYS2.PAYROLL.R14.MACROS	M14
SYS2.PAYROLL.R14.JCL	J14

you might assign data set id A14 (for All Release 14 payroll data sets) to **partial** data set name **SYS2.PAYROLL.R14**. You could then type:

4 A14

in the command line and press the PF key which invokes EZ-ISPF to get the ISPF Option 3.4 (Data Set List Utility) panel with SYS2.PAYROLL.R14 shown in the Dsname Level field. If you then press Enter after ensuring that the Volume Serial field is blank, an ISPF option 3.4 data set list will be produced showing all of the Payroll Release 14 data sets. You then have all the options offered by ISPF Option 3.4 available to process the data sets on the resulting list.

This reveals the true power of EZ-ISPF; that is, using a short sequence of keystrokes and a data set ID or target string to drive an existing ISPF option. So you can utilize the full power of ISPF, capitalize on your existing knowledge of ISPF, at the same time, minimize the amount of typing and the number of system interactions normally required to get a task done under ISPF.

As you use EZ-ISPF's assigned data set ID feature more and more, you will at some point want to display a list of all assigned data set Ids. You can do this by typing invocation parameter S (for Show Miscellaneous Information) with invocation subparm A (for Assigned data set id's) in the command line when you invoke EZ-ISPF, as shown below:

S A

The displayed list includes the data set name or data set name prefix assigned to each data set id. The data set ID list will probably be a list of the data sets you use most

frequently. Since EZ-ISPF can be invoked recursively, you can use any data set shown on the list as a target string in order to process that data set again with EZ-ISPF.

As shown above, the assigned data set id technique provides a flexible, comprehensive, extremely powerful facility for processing frequently used data sets.

History List

EZ-ISPF provides a list of the invocation parameter and the target string specified the last 32 times you invoked EZ-ISPF. You can display this list by invoking EZ-ISPF with invocation parameter S (for Show Miscellaneous Information) with invocation subparm H (for History), as shown below:

S H

Current PDS Search Set

The Current PDS Search Set is the set of one or more cataloged, concatenated PDS's to be searched to find a target string in the form of a PDS member name.

EZ-ISPF provides a number of ways to determine the Current PDS Search Set, including:

- the Context PDS Search Set
- the Last Indicated PDS Search Set
- Panel Display PDS(s)

Each of these is described below.

Context PDS Search Set

When the target string takes the form of a PDS member name, EZ-ISPF uses a concept called the Context PDS Search Set. The Context PDS Search Set is the set of one or more cataloged PDS's that, based upon the context in which that member name appears, is the "best" or "most likely" set of PDS's to search to find that member name. EZ-ISPF

automatically determines the Context PDS Search Set from the context in which the target string appears.

For example, in the JCL line below:

```
//SYSIN DD DSN=SYS2.PAYROLL.R14.JCL(JOB1),DISP=SHR
```

the Context PDS Search Set for PDS member name JOB1 is the single PDS SYS2.PAYROLL.R14.JCL.

Concatenated PDSs are also processed by EZ-ISPF, if all of the data sets in the concatenation appear on the ISPF screen at the same time, as in the JCL fragment below:

```
//JS10 EXEC PGM=PAY1  
//STEPLIB DD DSN=MY.TEST.LINKLIB,DISP=SHR  
//          DD DSN=SYS2.PAYROLL.R14.LINKLIB,DISP=SHR
```

For the JCL above, EZ-ISPF will automatically treat MY.TEST.LINKLIB and SYS2.PAYROLL.R14.LINKLIB as the Context PDS Search Set for executable program name PAY1, since it is obvious from the context that PAY1 will be executed from the PDS's concatenated to the STEPLIB DD statement. Specifying executable program name PAY1 as the target string will cause EZ-ISPF to search for member name PAY1 in the set of data sets concatenated to the STEPLIB DD statement above.

The same process applies to JCL procedures. For example, in the JCL statement below:

```
//JS10 EXEC PROC=PROC1
```

EZ-ISPF will automatically treat the system proclibs as the Context PDS Search Set for executable procedure name PROC1. Specifying executable procedure name PROC1 as the target string will cause EZ-ISPF to search for member name PROC1 in the system procedure libraries defined to JES.

Last Indicated PDS Search Set

EZ-ISPF allows you to designate a set of one or more concatenated PDS's as the set of PDS's to be searched for a PDS member name. This set of one or more PDS's is referred to as the Last Indicated PDS Search Set.

For example, in the JCL below:

```
//JS10 EXEC PGM=IEBCOPY
//INPUT1 DD DSN=TEST.PDS1,DISP=SHR
//      DD DSN=TEST.PDS2,DISP=SHR
//      DD DSN=TEST.PDS3,DISP=SHR
//OUTPUT1 DD DSN=OUTPUT.PDS,DISP=SHR
//SYSIN DD *
C O=OUTPUT1,I=((INPUT1,R))
S M=DLYPAY
```

suppose you want to check member DLYPAY to see which PDS it will be fetched from when this JCL is executed, and to see exactly what the DLYPAY member contains.

If you place the cursor under DD name INPUT1 and press the PF key which invokes EZ-ISPF, the Last Indicated PDS Search Set will be defined as data sets TEST.PDS1, TEST.PDS2, and TEST.PDS3. If you then place the cursor under PDS member name DLYPAY and press the PF key which invokes EZ-ISPF, EZ-ISPF will automatically search for DLYPAY in these three PDS's and will invoke ISPF Edit against the member in the first PDS which contains it.

If you want to determine whether PDS member DLYPAY is already contained in data set OUTPUT.PDS, you can place the cursor under DD name OUTPUT1 and press the PF key which invokes EZ-ISPF. The Last Indicated PDS Search Set will then be defined as the single PDS OUTPUT.PDS. If you then place the cursor under PDS member name DLYPAY and press the PF key which invokes EZ-ISPF, EZ-ISPF will automatically search for DLYPAY in data set OUTPUT.PDS.

Note the difference between the Context PDS Search Set and the Last Indicated PDS Search Set. The Context PDS search set is automatically determined by EZ-ISPF. The Last Indicated PDS Search Set is explicitly defined by the user.

Panel Display PDSs

Panel Display PDS(s) refers to the set of one or more cataloged PDS data set names that may appear on the current panel display.

When the target string is a PDS member name and no Context PDS Search Set is available and no Last Indicated PDS Search Set has been defined by the user, then EZ-ISPF will use the Panel Display PDSs to process the specified member name. If only one PDS data set name appears on the current panel display, then EZ-ISPF will search that PDS for the specified member name. If more than one PDS data set name appears on the screen, then EZ-ISPF will prompt the user to determine which PDS to search for the specified member name.

As an example, in the JCL below:

```
//JS10 EXEC PGM=PGM1
//INPUT1 DD DSN=TEST.PDS1,DISP=SHR
//SYSIN DD *
MEMONE
```

If you place the cursor under MEMONE and invoke EZ-ISPF, then ISPF Edit will be invoked against member name MEMONE in PDS TEST.PDS.

Chapter Three - Installing EZ-ISPF

Introduction

These install instructions assume that you have already obtained and unzipped the EZ-ISPF Release 1.8 installation ZIP archive file and that you have uploaded and executed the installation startup REXX exec. Depending on the options you selected, the startup REXX exec may or may not have completed the initial installation of EZ-ISPF. If you did not allow the startup REXX exec to complete the initial installation of EZ-ISPF, then you can follow the instructions which follow to complete the initial installation. Once the initial installation is completed, you can use the instructions which follow to make changes to your initial installation or to complete the permanent installation of EZ-ISPF.

All EZ-ISPF data set names are shown in these instructions as follows:

SYS2.EZISPF.R18.xxxxxxxxx

where xxxxxxxx represents the low-level node(s) that make each name unique.

Anytime you deviate from the "recommended" data set name (as you may have done when you ran the startup REXX exec), you must substitute the name you used in place of the recommended data set name when it appears in these instructions.

At this point, browse the installation log file (SYS2.EZISPF.R18.INSTALL.LOG) to see the installation work completed by the startup REXX exec.

Step 1 - Define EZ/EZS Commands

Review members CMDINFO, LIBDEF, and EZANDEZS in the EZ-ISPF JCL library. You must understand the information in these members before continuing.

If the EZ and EZS commands are already properly defined to ISPF, either by the startup REXX exec or by installation of a prior release of EZ-ISPF, then you should skip this step and proceed to Step 2.

If a previous release of EZ-ISPF was installed and you want to change the way in which the EZ and EZS commands are currently defined to ISPF, then see "Create/Update Site Command Table" in Appendix C.

If this is the first time EZ-ISPF has been installed and the EZ and EZS commands were not defined by the startup REXX exec or you want to change the way they were defined by the startup REXX exec, then continue with this step.

In what follows, if an ISPF site command table is present, then it is considered to be the active command table. Otherwise, the ISPF command table is considered to be the active command table.

If the startup REXX exec could not find an EZ command in the ISPF or site command table, then it produced two updated copies of the active command table. One updated copy defines the EZ and EZS commands for LIBDEF invocation of EZ-ISPF; the other defines the commands for normal (non-LIBDEF) invocation. Browse the installation log file to see if the active command table was replaced by the startup REXX exec and, if so, which updated copy was used.

If the active command table was not replaced or you want to change the way in which the EZ and EZS commands are currently defined, then, if you want to use LIBDEF invocation for EZ-ISPF, submit the JCL in member COPYLD in the EZ-ISPF JCL library. If you want to use normal (non-LIBDEF) invocation, submit the JCL in member COPYNL in the JCL library.

Step 2 - Install EZ-ISPF Under SMP/E

If you want to use SMP/E to complete the installation of EZ-ISPF, see "SMP/E Install" in Appendix D. Then continue with Step 3 below.

Step 3 - Apply Expiration Date Zap

You must apply an expiration date zap to ensure that EZ-ISPF continues to execute on your system. You will receive the expiration date zap as part of the EZ-ISPF installation package or via e-mail.

If you are not using SMP/E for the install, the expiration date zap must be applied using JCL in member EXPIREDT in the EZ-ISPF JCL library. The expected return code for this job is 0.

If you are using SMP/E for the install, you must RECEIVE and APPLY the expiration date zap as a USER MOD by using member SMPEXPD in the EZ-ISPF JCL library. The expected return code for this job is 0.

Step 4 - Implement ISPF ELEMENTS

You must now implement the various EZ-ISPF components into ISPF.

If you are not using SMP/E for the install, these include the following EZ-ISPF libraries that were created and loaded by the startup REXX exec:

	Recommended Data Set Name

EZ-ISPF link library	SYS2.EZISPF.R18.LINKLIB
EZ-ISPF panels library	SYS2.EZISPF.R18.PANELS
EZ-ISPF messages library	SYS2.EZISPF.R18.MESSAGES
EZ-ISPF command table library	SYS2.EZISPF.R18.TABLES

If you are using SMP/E for the install, these include the following target libraries defined via SMP/E:

	Recommended Data Set Name

EZ-ISPF target link library	SYS2.EZISPF.R18.XEZLINK
EZ-ISPF target panels library	SYS2.EZISPF.R18.XEZPNL
EZ-ISPF target messages library	SYS2.EZISPF.R18.XEZMSG
EZ-ISPF target command table library	SYS2.EZISPF.R18.XEZTBL

The EZ-ISPF components also include the modified copies of the IBM ISPF panels that were created by the startup REXX exec and stored in data set:

SYS2.EZISPF.R18.ISPFrr.PANELS (where rr is the ISPF
release number)

If you intend to use normal (non-LIBDEF) invocation, skip Step 4.1 below and proceed to Step 4.2.

Step 4.1 - Using LIBDEF Invocation

You must now modify the EZ LIBDEF clist so that it specifies the correct data set names. Changes to the data set names may already have been made by the startup REXX exec. In any case, review member EZ in the EZ-ISPF JCL library and modify the data set names if required.

Once you ensure that the required modifications have been made to the EZ LIBDEF clist, you must copy it to a library

in your TSO logon procedure's SYSPROC concatenation. Browse the installation log file to see if this copy operation was completed by the startup exec. If not, member COPYEZ in the JCL library may have been updated by the startup REXX exec as required to copy the EZ LIBDEF clist to the first PDS in your SYSPROC concatenation. If so, you may use member COPYEZ to do this copy operation. Otherwise, ISPF option 3.3 can be used for this purpose. Once a properly modified copy of the EZ LIBDEF clist is in the SYSPROC concatenation, skip Step 4.2 and proceed to Step 5.

Step 4.2 - Using Normal Invocation

You must now copy REXX exec EZSN from the EZ-ISPF JCL library to a library in your TSO logon procedure's SYSPROC or SYSEXEC concatenation. Browse the installation log file to see if this copy operation was done by the startup REXX exec. If not, member COPYEZSN in the JCL library may have been updated by the startup REXX exec as required to copy EZSN to the first PDS in your SYSPROC concatenation. If so, you may use member COPYEZSN to do this copy operation. Otherwise, ISPF option 3.3 can be used for this purpose.

You must now modify your TSO logon proc so that each library from the appropriate list in Step 4 above is concatenated to the appropriate ISPF DD statement. Browse the installation log file to see if a test copy of your logon proc was updated by the startup REXX exec.

In any case, here are the recommended data set names and the DDname of the TSO logon proc DD statement each must be concatenated to:

DDname	Recommended Data Set Name (Non-SMP/E Install)	Recommended Target Library Data Set Name (SMP/E Install)
ISPPLLIB	SYS2.EZISPF.R18.LINKLIB	SYS2.EZISPF.R18.XEZLINK
ISPPLIB	SYS2.EZISPF.R18.PANELS	SYS2.EZISPF.R18.XEZPNL
ISPMLIB	SYS2.EZISPF.R18.MESSAGES	SYS2.EZISPF.R18.XEZMSG
ISPTLIB	SYS2.EZISPF.R18.TABLES	SYS2.EZISPF.R18.XEZTBL

DDname	Recommended Data Set Name (Either Type of Install)
ISPPLIB	SYS2.EZISPF.R18.ISPFrr.PANELS

(where rr is the ISPF release number)

Note that DDname ISPPLIB must have two data sets concatenated to it: one containing the EZ-ISPF panels and one containing the modified IBM ISPF panels. The library containing the modified IBM ISPF panels must be concatenated ahead of the PDS containing the original IBM panels.

Step 5 - Test EZ-ISPF Installation

After logging out of TSO and logging back on, type:

EZ S D

in the command line of any ISPF panel and press Enter. This will start the EZ-ISPF demo, which provides a good testing tool. If the demo does not work correctly, review/repeat the installation steps above. Also review the trouble-shooting tips in member TRBLTIPS in the EZ-ISPF JCL library.

Step 6 - Additional Considerations

Review member OPTDESC in the JCL library to determine if you need to modify the EZ-ISPF options table. If so, see "Modifying Options Table" in Appendix F.

If you used SMP/E for the install, you should ACCEPT the USER MOD for the expiration date zap using the JCL in member SMPACEXP in the JCL library.

If you used SMP/E for the install and you modified the EZ-ISPF options table, you should ACCEPT the USER MOD for the options table using the JCL in member SMPACOPT in the JCL library.

You may want to apply a USER MOD to ISPF for the modified IBM panels (so they will be protected from inadvertent updates) by using member SMPPNLS in the JCL library. Before submitting this job, specify the correct FMID for the associated release of ISPF in member XEZPNLS in the JCL library.

To implement EZ-ISPF into production for all ISPF users, make sure the modified ISPF system or site command table containing the EZ and/or EZS commands is included in the concatenation for your production ISPF table library (ISPTLIB) and that it replaces or is concatenated ahead of the non-modified version. If you are not using LIBDEF to invoke EZ-ISPF, make sure the EZ-ISPF panels library, messages library, linklib, and application command table (QWIKCMDS) are included in the ISPPLIB, ISPMLIB, ISPLLIB, and ISPTLIB concatenations, respectively, and that the EZSN REXX exec is stored in your production REXX exec library. If you are using LIBDEF, make sure the EZ LIBDEF clist is stored in your production clist library.

Notify your ISPF users about EZ-ISPF using member USERMEMO in the JCL library.

If you install a new version of ISPF and need to redo the IBM ISPF panel modifications, see "Modifying IBM ISPF Panels" in Appendix E.

Appendix A - EZ-ISPF Messages

The following messages are issued by EZ-ISPF:

Message Number: XEZM000S

Message text: 'XEZM000S - XEZOPTS is from an older release and cannot be used'

Explanation: The version of the EZ-ISPF options table XEZOPTS is from an older release of the product and is not compatible with your release of EZ-ISPF.

User Action: Reassemble the XEZOPTS table, using current versions of the XEZOPT macro.

Message Number: XEZM001S

Message text: 'XEZM001S - File allocation/open failure; check file attributes'

Explanation: EZ-ISPF could not open or allocate a needed data set.

User Action: Check for other operating system messages indicating which type of error led to this condition. Make sure that the correct data set name is specified.

Message Number: XEZM002I

Message text: 'XEZM002I - Please wait; the dump you requested is now being produced'

Explanation: The EZ-ISPF ESTAE exit is producing a dump after being requested to do so.

User Action: Save the dump and forward it to the EZ-ISPF Support Group at Referential Systems, if requested to do so.

Message Number: XEZM003I

Message text: 'XEZM003I - Dump complete; press ENTER to resume normal termination processing'

Explanation: The EZ-ISPF ESTAE exit is producing a dump after being requested to do so.

User Action: Save the dump and forward it to the EZ-ISPF Support Group at Referential Systems, if requested to do so.

Message Number: XEZM004E

Message text: 'XEZM004E - Specified invocation parm is not valid'

Explanation: The invocation parameter(s) used when EZ-ISPF was invoked are invalid.

User Action: Use the 'EZ HP' command to get a list of valid EZ-ISPF invocation parameters, then invoke EZ-ISPF again with a valid parameter.

Message Number: XEZM005S

Message text: 'XEZM005S - Return code nn received from cursor parsing module XEZCURSR'

Explanation: An error return code was passed back from the EZ-ISPF on-screen cursor location parsing module.

User Action: Attempt the operation again; if the problem persists, contact the EZ-ISPF Support Group at Referential Systems

Message Number: XEZM005E

Message text: 'XEZM005E - Specified invocation parm requires at least one subparm'

Explanation: The user invoked EZ-ISPF with a parameter that also requires the use of at least one subparameter.

User Action: Invoke EZ-ISPF with a correct invocation parameter/subparameter pair.

Message Number: XEZM006E

Message text: 'XEZM006E - Specified invocation parm requires target string'

Explanation: The cursor must be placed on an on-screen target (non-blank) string when EZ-ISPF is invoked with this type of parameter.

User Action: Invoke EZ-ISPF again with the same parameter, but place the cursor on an on-screen target string first.

Message Number: XEZM007E

Message text: 'XEZM007E - Specified invocation parm requires subparm or target string'

Explanation: EZ-ISPF was invoked with a parameter that also requires either a subparameter, or that the cursor be placed on a non-blank on-screen target string.

User Action: Invoke EZ-ISPF again with the same parameter,
but place the cursor on an on-screen target string
or specify a subparameter first.

Message Number: XEZM008E

Message text: 'XEZM008E - Specified target string is not
recognized'

Explanation: The cursor was placed on a type of on-screen
non-blank target string that is in a format which
EZ-ISPF cannot recognize in the context in which
EZ-ISPF was invoked.

User Action: Invoke EZ-ISPF again with the cursor on a valid
target string.

Message Number: XEZM009E

Message text: 'XEZM009E - dsn is a PDS which contains no
members'

Explanation: EZ-ISPF was invoked to perform an action on a
Partitioned Data Set, but the data set is empty;
that is, it contains no members.

User Action: Ensure that the type of action you want EZ-ISPF
to perform or to request ISPF to perform is
appropriate for the PDS.

Message Number: XEZM010E

Message text: 'XEZM010E - Member or dataset in use, YES
specified for LOCK parm, or request disallowed by ISPF
config table'

Explanation: EZ-ISPF cannot gain access to the data set or
member requested because the data set or member is
under the exclusive control of another job or user
or because YES was specified for the LOCK parm or
the processing you requested is not allowed by the
ISPF config table.

User Action: If the member or dataset is in use, wait until
the data set or member is freed by the owning job
or user.

Message Number: XEZM011I

Message text: 'XEZM011I - Data set not cataloged; you may
allocate or catalog it here'

Explanation: The data set EZ-ISPF has been invoked to
process is not cataloged; you may use the current
on-screen facility to catalog the data set if
desired.

User Action: Catalog the data set if desired.

Message Number: XEZM012E
Message text: 'XEZM012E - Specified data set ID too long or contains invalid characters'
Explanation: EZ-ISPF was invoked with a data set ID that is in an invalid format.
User Action: Invoke EZ-ISPF again with a valid data set ID. EZ-ISPF data set IDs can be up to 5 characters long, and can consist of any combination of alphanumeric and special characters (#,\$,@).

Message Number: XEZM013E
Message text: 'XEZM013E - Data set id id already defined'
Explanation: EZ-ISPF was invoked to assign a data set ID to a data set, but the ID requested has already been assigned to this or another data set by EZ-ISPF.
User Action: Invoke EZ-ISPF again using a different data set ID if desired.

Message Number: XEZM014E
Message text: 'XEZM014E - Invocation parm A requires data set name to be assigned'
Explanation: You invoked EZ-ISPF with the 'A' parameter to assign a data set ID to a data set, but no data set name was specified.
User Action: Specify the data set name as a target string or as an invocation subparm.

Message Number: XEZM015I
Message text: 'XEZM015I - Previous invocation of EZ-ISPF has completed processing'
Explanation: EZ-ISPF has completed processing successfully.
User Action: None.

Message Number: XEZM016I
Message text: 'XEZM016I - Data set id xx assigned to dsn'
Explanation: The indicated data set ID was associated with the indicated data set.
User Action: None.

Message Number: XEZM017I
Message text: 'XEZM017I - Last indicated PDS search set defined as specified...'
Explanation: EZ-ISPF has honored a request to establish a set of PDS's to use as its search set for subsequent requests.
User Action: None.

Message Number: XEZM018E
Message text: 'XEZM018E - Indicated target contains non-PDS or non-cataloged data set dsn'
Explanation: A data set that was specified as a target string is non a Partitioned Data Set or is not cataloged.
User Action: Respecify the request with a PDS that is cataloged.

Message Number: XEZM019E
Message text: 'XEZM019E - Indicated target associated with more than 32 PDS's'
Explanation: EZ-ISPF was invoked for target string that is part of a PDS concatenation consisting of more than 32 PDS's. Concatenations of this size are not supported by EZ-ISPF.
User Action: Invoke EZ-ISPF for the indicated data set using some other mechanism, such as assigning a data set ID to it.

Message Number: XEZM020E
Message text: 'XEZM020E - Specified target string not valid with invocation parm L or O'
Explanation: The type of target string that the cursor is on is not compatible with EZ-ISPF invocation parameter 'L' (Define Last Indicated PDS Search Set) or 'O' (Define Last Indicated PDS Search Set As Only Search Set).
User Action: Reposition the cursor onto a valid PDS data set name on the screen, or use a different EZ-ISPF invocation parameter.

Message Number: XEZM021I
Message text: 'XEZM021I - No last indicated PDS search set is now defined'
Explanation: Your request that the last indicated PDS search set be nullified has been honored.
User Action: None.

Message Number: XEZM022E
Message text: 'XEZM022E - No current PDS search set available or specified'
Explanation: EZ-ISPF could not act upon the requested member because there no current PDS search set was

specified and none is visible on the current display.

User Action: Provide a current PDS search set for the member by specifying an assigned data set id or by using the 'L' or 'O' parameters.

Message Number: XEZM023W

Message text: 'XEZM023W - type memname not found in ...'

Explanation: The requested member could not be found.

User Action: Try again with a different member name.

Message Number: XEZM024E

Message text: 'XEZM024E - type memname not found; cannot create with EDIT in ... dsn '

Explanation: The requested member could not be located in the PDS, nor could it be created.

User Action: Try again using a different member name.

Message Number: XEZM025I

Message text: 'XEZM025I - dsn is a partial data set name'

Explanation: The indicated target string is a partial data set name.

User Action: None.

Message Number: XEZM026E

Message text: 'XEZM026E - Program memname not found in xxxLPA or Link List'

Explanation: EZ-ISPF could not locate the specified program anywhere on the system link list nor in LPALIB.

User Action: Make sure a valid program name is specified and retry the request.

Message Number: XEZM027I

Message text: 'XEZM027I - type memname found in ... dsn'

Explanation: EZ-ISPF found the requested member in the data set specified.

User Action: None.

Message Number: XEZM028W

Message text: 'XEZM028W - type memname not found; you may create it in ... dsn ...'

Explanation: The specified member name was not found. You may create it in the indicated PDS.

User Action: Enter data for the indicated member name if desired.

Message Number: XEZM029W

Message text: 'XEZM029W - Pattern cannot be searched for in LPA or Link List'

Explanation: EZ-ISPF cannot search for a partial program name in the LPA or Link List.

User Action: Try the search using some other technique.

Message Number: XEZM030W

Message text: 'XEZM030W - PDS search set defined by parm O; use S L to review; N to nullify'

Explanation: EZ-ISPF has established the given PDS set as the 'only' search set to use when looking for a target member to act upon. Only the indicated search set will be used, and any on-screen context PDS will not be used.

User Action: None.

Message Number: XEZM031E

Message text: 'XEZM031E - Data set dsn cannot be opened; check for security violation'

Explanation: The indicated data set could not be opened, probably because the user does not have sufficient authority, as specified by the local security system (e.g., RACF).

User Action: Retry the request using a data set name that you are authorized to open.

Message Number: XEZM032E

Message text: 'XEZM032E - Last Indicated PDS search set contains invalid data set'

Explanation: A data set defined as being part of the PDS search set established via EZ-ISPF parameter 'L' or 'O' and the on-screen cursor position is not a PDS.

User Action: Retry the request using a different on-screen search set context or search set. Only PDS's are supported by EZ-ISPF for this purpose.

Message Number: XEZM033W

Message text: 'XEZM033W - message'

Explanation: This message displays variable EZ-ISPF warning data that is self-explanatory.

User Action: Take action based upon the warning in the displayed text.

Message Number: XEZM034E

Message text: 'XEZM034E - text ddname missing or invalid'

Explanation: EZ-ISPF cannot find the indicated JES2 or JES3 ddname.

User Action: Contact the EZ-ISPF Support Group at Referential Systems for assistance.

Message Number: XEZM035E

Message text: 'XEZM035E - Internal storage area not large enough; information truncated'

Explanation: The EZ-ISPF function requested produced a display that was too large for the virtual storage area defined.

User Action: Increase the size of EZ-ISPF's virtual storage display build area using a larger value for the TEXTSTR= keyword in the EZ-ISPF options table XEZOPTS, or redrive the EZ-ISPF request with a parameter or subparameter value that will result in less data being displayed.

Message Number: XEZM036E

Message text: 'XEZM036E - STEPLIB contains non-PDS or non-cataloged data set dsn'

Explanation: EZ-ISPF was invoked to establish a PDS search set using the list of data sets concatenated to an on-screen STEPLIB DD statement, but one or more of the data set names appearing on screen was not cataloged or not a PDS.

User Action: Retry the request using a valid STEPLIB concatenation.

Message Number: XEZM037E

Message text: 'XEZM037E - JOB CLASS definition(s) could not be located'

Explanation: EZ-ISPF was invoked with the cursor placed on a JES job class, but EZ-ISPF could not determine what the JES2 or JES3 definitions were for that job class.

User Action: Retry the request with a valid job class. The on-screen value that the cursor is on may not be a valid JES2 or JES3 job class.

Message Number: XEZM038E

Message text: 'XEZM038E - type DD statements not valid or not positioned correctly'
Explanation: EZ-ISPF cannot determine the PDS's concatenated to the indicated DD statement.
User Action: Make sure all portions of the indicated DD statement are visible on the current display and that the indicated DD statement is formatted correctly and contains cataloged data set names.

Message Number: XEZM039E

Message text: 'XEZM039E - Target statements not valid or not positioned correctly'
Explanation: EZ-ISPF cannot determine the PDS's concatenated to the target DD statement.
User Action: Make sure all portions of the target DD statement are visible on the current display and that the target DD statement is formatted correctly and contains valid cataloged data set names.

Message Number: XEZM041E

Message text: 'XEZM041E - Requested volume is not mounted, or unit name is invalid'
Explanation: EZ-ISPF was invoked with a DASD volser or DASD unit name as a target string, but the requested volser or unit name is not valid on this system.
User Action: Retry the request with a valid DASD volser or unit name.

Message Number: XEZM042S

Message text: 'XEZM042S - Internal error - Invalid line length was calculated'
Explanation: An internal EZ-ISPF logic error has occurred.
User Action: Contact the EZ-ISPF Support Group at Referential Systems for assistance.

Message Number: XEZM043E

Message text: 'XEZM043E - Sysout Class definition(s) could not be located'
Explanation: EZ-ISPF was invoked with the cursor placed on a JES SYSOUT class, but EZ-ISPF could not determine what the JES2 or JES3 definitions were for that SYSOUT class.
User Action: Retry the request using a valid SYSOUT class.

Message Number: XEZM044S

Message text: 'XEZM044S - Internal error -Invalid branch table offset calculated'

Explanation: An internal EZ-ISPF logic error has occurred.
User Action: Contact the EZ-ISPF Support Group at
Referential Systems for assistance.

Message Number: XEZM045S
Message text: 'XEZM045S - Internal error - Column specified
is not in branch table'
Explanation: An internal EZ-ISPF logic error has occurred.
User Action: Contact the EZ-ISPF Support Group at
Referential Systems for assistance.

Message Number: XEZM046E
Message text: 'XEZM046E - The SORT command can be used only
when DASD info is displayed'
Explanation: You used the SORT command in an invalid context
within EZ-ISPF. Only EZ-ISPF's on-screen DASD free
space display supports the SORT command.
User Action: Retry the request in a valid context.

Message Number: XEZM047E
Message text: 'XEZM047E - Required operand is missing or
invalid on SORT command'
Explanation: One or more operands on the SORT command were
invalid.
User Action: Retry the request with valid operands.

Message Number: XEZM048E
Message text: 'XEZM048E - Specified sort field has been
omitted from display'
Explanation: You attempted to sort the EZ-ISPF DASD free
space display using a column number that is
invalid; the column has been "turned off" for
display by EZ-ISPF via a parameter in the EZ-ISPF
options table, XEZOPTS.
User Action: Sort on a different column number, or reinstate
the DASD free space column requested and try the
request again.

Message Number: XEZM049I
Message text: 'XEZM049I - SYSOUT chr * OR \$ means use
MSGCLASS; put cursor under MSGLCASS chr'
Explanation: EZ-ISPF was invoked with a SYSOUT character as
the target string, but the character the cursor is
on is a '*' or '\$'. These two characters tell JES
to use the MSGCLASS that is specified as the SYSOUT
class.

User Action: If you want to see the JES definition for that SYSOUT class, put the cursor on the value specified for the MSGCLASS= keyword on the JOB statement and invoke EZ-ISPF again.

Message Number: XEZM050E

Message text: 'XEZM050E - Target, invocation parm, or dataset/member inconsistent or in-use'

Explanation: EZ-ISPF was invoked with a target string, an invocation parm, or a dataset or member that is inconsistent or in-use. For example, requesting a DASD free space display via EZ-ISPF invocation parameter 'D' when the cursor is on a data set name (instead of a DASD volser) is an inconsistent request.

User Action: Retry the request with a consistent target string, invocation parm, dataset, and member which are not currently in-use.

Message Number: XEZM051E

Message text: 'XEZM051E - Specified data set ID is not an assigned data set ID'

Explanation: EZ-ISPF was invoked with a data set ID that was not previously defined via the 'A' invocation parameter.

User Action: Retry the request using a valid data set ID.

Message Number: XEZM052E

Message text: 'XEZM052E - dsn is a physical seq data set with no records and/or RECFM=U'

Explanation: The named data set is either an empty sequential data set or is a RECFM=U file, which is invalid for the context in which EZ-ISPF was invoked.

User Action: Retry the request using a valid data set name.

Message Number: XEZM053E

Message text: 'XEZM053E - type memname found in ... dsn - but contains no records'

Explanation: You have asked EZ-ISPF to act upon an empty PDS member.

User Action: Retry the request with a different requested function or a different member name.

Message Number: XEZM054E
Message text: 'XEZM054E - Invocation parm S requires subparm A, H, I, or L'
Explanation: You must supply a valid subparameter when the 'S' invocation parameter is used.
User Action: Retry the request using a valid subparameter.

Message Number: XEZM055E
Message text: 'XEZM055E - Maximum limit of 100 assigned data set ID's has been reached'
Explanation: 100 data set IDs have been defined to EZ-ISPF, and you requested that another data set ID be defined. 100 is EZ-ISPF's design limit for this feature.
User Action: Invoke EZ-ISPF again like this: "EZ S A", then delete one or more assigned data set IDs. You will then be able to define the desired ID.

Message Number: XEZM056I
Message text: 'XEZM056I - No data set ID's are currently assigned'
Explanation: No data set IDs have yet been assigned.
User Action: Define one or more data set IDs by placing the cursor on a valid data set name on-screen and then invoking EZ-ISPF with the 'A' invocation parameter. You will then be able to display the data set IDs using the "EZ S A" command.

Message Number: XEZM057E
Message text: 'XEZM057E - XEZOPTS SDSF SELECT Service Specification is invalid'
Explanation: The SELECT Service specification used for SDSF in the XDKOPTS table is invalid.
User Action: Correct the XEZOPTS SDSF SELECT Service specification and reassemble and relink the XEZOPTS table.

Message Number: XEZM058I
Message text: 'XEZM058I - Specified data set ID(s) were deleted'
Explanation: The selected data set ID(s) were deleted from the defined set, as requested.
User Action: None.

Message Number: XEZM059I

Message text: 'XEZM059I - No Last Indicated PDS Search Set is currently defined'

Explanation: You requested a display of the last indicated PDS search set, but no last indicated PDS search set is currently defined.

User Action: None.

Message Number: XEZM060I

Message text: 'XEZM060I - ''From'' now defined as member memname'

Explanation: The indicated member has been defined as the 'from' member for a subsequent EZ-ISPF invocation of an ISPF option.

User Action: None.

Message Number: XEZM061I

Message text: 'XEZM061I - ''From'' now defined as type dsn'

Explanation: The indicated data set has been defined as the 'from' data set for a subsequent EZ-ISPF invocation of an ISPF option.

User Action: None.

Message Number: XEZM062I

Message text: 'XEZM062I - ''To'' now defined as member memname'

Explanation: The indicated member has been defined as the 'to' member for a subsequent EZ-ISPF invocation of an ISPF option.

User Action: None.

Message Number: XEZM063I

Message text: 'XEZM063I - ''To'' now defined as type dsn'

Explanation: The indicated data set has been defined as the 'to' data set for a subsequent EZ-ISPF invocation of an ISPF option.

User Action: None.

Message Number: XEZM064E

Message text: 'XEZM064E - Invalid command; commands are HELP, FIND, SORT, ZPRINT'

Explanation: The EZ-ISPF command you entered is valid.

User Action: Retry the request using one of the listed valid EZ-ISPF commands.

Message Number: XEZM065E

Message text: 'XEZM065E - No string to find was specified'
Explanation: A FIND command was issued without a string to
find being specified.
User Action: Retry the FIND request and specify a string
that EZ-ISPF is to look for.

Message Number: XEZM066I
Message text: 'XEZM066I - Characters "xxxxxxx" found'
Explanation: The requested character string was found by EZ-
ISPF.
User Action: None.

Message Number: XEZM067W
Message text: 'XEZM067W - Characters "xxxxxxx" not found'
Explanation: The requested character string was not found by
EZ-ISPF.
User Action: Retry the request with another string if
desired.

Message Number: XEZM068W
Message text: 'XEZM068W - Bottom of data reached'
Explanation: The requested string was not found prior to the
bottom of the text being searched.
User Action: Retry the FIND command from the top of the data
if desired.

Message Number: XEZM069E
Message text: 'XEZM069E - SDSF is not supported in this
installation of EZ-ISPF'
Explanation: EZ-ISPF cannot be used in your TSO address
space to invoke SDSF.
User Action: SDSF may not be available, or it may be
protected by your installation's security system.
Contact your local System Programmer that is
responsible for SDSF if you feel that you should
have access to it.

Message Number: XEZM070W
Message Text: 'XEZM070W - Browse substituted for text'
Explanation: Browse was substituted for indicated request
because indicated request is not valid for
specified item.
User Action: None required.

Message Number: XEZM071S
Message Text: 'XEZM071S - Last accessed IBM panel not modified as required for EZ-ISPF'
Explanation: The last accessed IBM panel has not been modified as required for EZ-ISPF processing.
User Action: Check your ISPF panel definitions/allocations and make sure you are running with the modified versions of the IBM panels that must be modified for EZ-ISPF. If necessary, use REXX exec EZPNLS as described in Appendix E to modify the required IBM panels.

Message Number: XEZM072E
Message Text: 'XEZM072E - Specified member name is not valid'
Explanation: The specified member name is not valid.
User Action: Specify a valid member name.

Message Number: XEZM075E
Message text: 'XEZM075E - Cannot specify both USER ID and WRITER with ZPRINT command'
Explanation: You entered a EZ-ISPF ZPRINT command but you specified both the USERID and the WRITER parameters, which are mutually exclusive.
User Action: Retry the ZPRINT request with valid parameters.

Message Number: XEZM077I
Message text: 'XEZM077I - Any changes made to the profile were applied temporarily'
Explanation: Profile changes specified (for ZPRINT for example) are only temporary in use for this ISPF session.
User Action: None.

Message Number: XEZM078I
Message text: 'XEZM078I - The profile was permanently updated'
Explanation: The ISPF profile values EZ-ISPF uses for this option were saved for use in subsequent EZ-ISPF sessions.
User Action: None.

Message Number: XEZM079S
Message text: 'XEZM079S ALLOC FAILED S99ERR=&S99E S99INFO=&S99I'

Explanation: A dynamic allocation request failed with the SVC 99 error and information codes displayed.

User Action: Use the given error information to determine the cause of the error and correct the problem.

Message Number: XEZM080W

Message text: 'XEZM080W - There is not yet any history to display'

Explanation: You requested a display of the EZ-ISPF target string history list, but no history list has yet been established by EZ-ISPF for your TSO userid.

User Action: Retry the list later after you have more of a history of using EZ-ISPF. Placing the cursor on valid on-screen target strings and then invoking EZ-ISPF with valid parameters will cause EZ-ISPF to save a history list of your target string use.

Message Number: XEZM081I

Message text: 'XEZM081I - Text was successfully written to the ZPRINT DD statement'

Explanation: The ZPRINT command you entered was honored.

User Action: None.

Message Number: XEZM082E

Message text: 'XEZM082E - The WRITER operand on the ZPRINT command is invalid'

Explanation: The WRITER operand specified an invalid value for the ZPRINT command, as entered.

User Action: Retry the ZPRINT with a valid parameter.

Message Number: XEZM083E

Message text: 'XEZM083E - The FCB operand on the ZPRINT command is invalid'

Explanation: The FCB operand specified an invalid value for the ZPRINT command, as entered.

User Action: Retry the ZPRINT with a valid parameter.

Message Number: XEZM084E

Message text: 'XEZM084E - The USER operand on the ZPRINT command is invalid'

Explanation: The USER operand specified an invalid value for the ZPRINT command, as entered.

User Action: Retry the ZPRINT with valid parameter.

Message Number: XEZM085E

Message text: 'XEZM085E - Specified invocation subparm is too long'
Explanation: You specified a subparameter that is too long for the parameter it appeared with.
User Action: Retry the request with a valid subparameter.

Message Number: XEZM086E
Message text: 'XEZM086E - CLASS or COPIES operand on the ZPRINT command is invalid'
Explanation: The CLASS or COPIES keywords you specified on the ZPRINT command were invalid.
User Action: Retry the request with valid keywords.

Message Number: XEZM087E
Message text: 'XEZM087E - The DEST or FORM operand on the ZPRINT command is invalid'
Explanation: The DEST or FORM keywords you specified on the ZPRINT command were invalid.
User Action: Retry the request with valid keywords.

Message Number: XEZM088S
Message text: 'XEZM088S - The ZPRINT SYSOUT file could not be opened - try again'
Explanation: ZPRINT could not honor a request to print to the specified SYSOUT file.
User Action: Retry the request with a valid SYSOUT specification.

Message Number: XEZM089W
Message text: 'XEZM089W - One or more ZPRINT output records were truncated'
Explanation: ZPRINT executed a print request, but one or more output print lines were truncated.
User Action: Retry the ZPRINT using an output specification that will not cause truncation.

Appendix B - EZ-ISPF vs. REFLIST

ISPF users familiar with the REFLIST function of ISPF will see some similarities in EZ-ISPF and REFLIST. There are key differences that make EZ-ISPF vastly superior to the REFLIST function. REFLIST's chief purpose is to provide for the storage and retrieval of data set names that have previously been allocated by ISPF.

Here are some points that highlight the differences between REFLIST and EZ-ISPF:

REFLIST can only be used on certain panels; EZ-ISPF can be invoked from *any* ISPF panel.

REFLIST can be used to get a list of the last 30 referenced data sets; EZ-ISPF provides a similar "history" list which also shows which ISPF function was last invoked by EZ-ISPF for that data set name.

REFLIST can be used to get a list of frequently used data sets, but to retrieve a name from the list, you generally have to:

specify a command indicating the particular list you want as well as know or remember the "number" of the data set you want from that list

or

specify a command indicating the particular list you want displayed and then place the cursor under the data set you want to use

With EZ-ISPF, frequently used data sets can be retrieved by simply specifying an easy to remember data set ID, or by placing the cursor on the data set name and pressing the EZ PF key.

REFLIST is of no use at all when looking at an ISPF screen containing a data set name that has never been allocated by ISPF. With EZ-ISPF, a data set name that has never been allocated by ISPF can still be processed by simply placing the cursor under the data set name wherever it happens to appear on the display screen and pressing the EZ PF key.

REFLIST can only be used to retrieve a data set name. With EZ-ISPF, you can retrieve and process the data set name in one step.

Effective REFLIST use requires many different ISPF commands. The list of REFLIST fastpath commands includes REFLISTD, REFLISTL, REFOPEND, REFOPENL, REFACTD, REFACTL, REFADDD, REFADDL, and NRETRIEV. Most users won't assign all of these command names to PF keys, so the command names must be remembered and typed when needed. This degrades productivity and can even confuse the user if the commands are not frequently used, since the command names are so similar. EZ-ISPF's simple command structure and cursor location awareness provides a rapid productivity boost without the need to keep track of so many different command names.

Appendix C - Create/Update Site Command Table

This appendix describes how you can use the EZCMDS clist to create or update the ISPF site command table so that the two commands used to invoke EZ-ISPF (EZ and/or EZS) are properly defined to ISPF.

EZCMDS should only be used if a previous release of EZ-ISPF was installed and you want to change the way in which the EZ and EZS commands are currently defined.

Step C.1 - Determine Commands To Be Defined

You must now decide if you want to add both commands (EZ and EZS) to the site command table. To determine this, review the information in the member named EZANDEZS in the EZ-ISPF JCL library.

Step C.2 - Determine if Updating or Creating

In this step, you will determine if you are going to be updating an existing site command table or creating the site command table for the first time.

Site command tables are documented in the ISPF Planning and Customization Guide. Recent releases of ISPF use option 3.9 to show you the application ID assigned to your site command table (if one already exists) on the right hand side of the option 3.9 display. You should now use ISPF option 3.9 to determine whether your installation already has an ISPF site command table defined; if it does, copy down its application id (i.e., applid) from the ISPF 3.9 display.

If you are creating a new site command table, then you will have to determine the applid to be assigned to the new site command table. It is recommended that you use '**SITE**' as the applid for the new site command table. This means that the PDS member name of the member containing the site command

table will be **SITECMDS**. This makes the name of the site command table member easy to recognize and easy to remember.

Step C.3 - Allocate PDS For Site Table

In this step, you will allocate the table library used to store the updated or newly created site command table.

If you are creating a new site command table, the EZ-ISPF support group recommends that you name this library **SYS2.EZISPF.R18.NEW.SITECMDS** so there will be no confusion later about where the newly created site command table is stored. If you are updating an existing ISPF site command table, the EZ-ISPF support group recommends that you name this library **SYS2.EZISPF.R18.UPD.SITECMDS** so there will be no confusion later about where the updated site command table is stored.

JCL to allocate this library can be found in member **EZAT** in the EZ-ISPF JCL library. Be sure to read the comments in this JCL member before submitting the job for execution. Once this job executes with a return code of zero you can proceed to the Step C.4.

Step C.4 - Copy Existing Site Command Table

If you are creating a new site command table, you should skip this step and proceed to Step C.5.

If you are updating an existing site command table, use ISPF option 3.3 to copy your installation's current site command table member from the production ISPF table library to the table library you just allocated in Step C.3 above. This copy of the existing site command table is referred to as the "working copy" of the site command table.

If you used the recommended name for the table library allocated in Step C.3, the name of the library you will be copying to is:

SYS2.EZISPF.R18.UPD.SITECMDS

The site command table member you should copy can be found in the set of PDS's concatenated to DDname **ISPTLIB** in your

TSO LOGON procedure(s). You can determine which PDS's are concatenated to DDname ISPTLIB in your current ISPF environment by executing the LISTFILE REXX exec in the EZ-ISPF JCL library. For a description of LISTFILE, see the member named LFDESC in the EZ-ISPF JCL library.

Step C.5 - Create or Update Site Table

In this step, you will use the EZCMDS clist to create a new site command table or to update the existing site command table.

EZCMDS is stored in the EZ-ISPF JCL library. This clist will automatically add the EZ and/or EZS commands used to invoke EZ-ISPF to the updated or newly created site command table.

EZCMDS will first check to see if a site command table is already defined to ISPF. If so, you will be asked to confirm the applid id assigned to the site command table and EZCMDS will assume you are updating an existing site command table.

If EZCMDS determines that no site command table is currently defined to ISPF, then it will assume you are creating a new site command table and it will ask for the applid id to be assigned to the new site command table. The recommended applid for the site command table is 'SITE'.

The EZCMDS clist will then ask you for the following additional information:

- if you are updating an existing site command table, then EZCMDS will ask for the name of the PDS containing the working copy of the site command table (if you used the recommended name when this PDS was created in Step C.3, the name is SYS2.EZISPF.R18.UPD.SITECMDS)
- if you are creating a new site command table, then EZCMDS will ask for the name of the PDS where you want the new site command table member to be stored (if you used the recommended name when this PDS was created in Step C.3, the name is SYS2.EZISPF.R18.NEW.SITECMDS)

- whether or not you will be using the LIBDEF method to invoke EZ-ISPF; if you don't know or are not sure, see the member named LIBDEF in the EZ-ISPF JCL library
- whether or not you want to add the EZ command to the site command table being updated or created
- whether or not you want to add the EZS command to the site command table being updated or created

Note: Do not invoke the EZCMDS clist until you are certain of the values that you need to specify, as outlined above.

You must invoke the EZCMDS clist under ISPF using the TSO EXEC command. To do this, go to ISPF option 6 and type:

```
EXEC 'SYS2.EZISPF.R18.JCL(EZCMDS)'
```

and press ENTER. Reply to the prompts issued by EZCMDS.

Based upon your responses to the questions which it asks, EZCMDS will either update the working copy of the site command table or create a new site command table and place it in the specified output PDS.

Note: The data set updated by the EZCMDS clist is not closed by the ISPF table update services invoked by the EZCMDS clist. For this reason, you must log off of TSO and log back on again if you want to invoke the EZCMDS clist for a second time.

Step C.6 - Implement Site Command Table

In this step, you will implement the site command table which you created or updated in Step C.5 above.

If you updated an existing site command table, then you need to replace the existing site command table in the production ISPF table library with the newly updated site command table. If you used the recommended data set name, the PDS containing the updated site command table member is named:

SYS2.EZISPF.R18.UPD.SITECMDS

Use ISPF option 3.3. to replace the existing site command table member in the production ISPF table library. This is the PDS from which you pulled your working copy of the site command table in Step C.4.

WARNING: It's a good idea to make a backup copy of the existing site command table before you replace it with the updated site command table.

If you updated an existing site command table, then you should skip the rest of this step and proceed to Step C.7.

If you created a new site command table, then you need to add the new site command table to the **ISPTLIB** concatenation. You can do this in one of two ways:

- by modifying your TSO logon proc so that the PDS containing the new site command table member is added to the ISPTLIB concatenation
- by copying the new site command table into a library already in the ISPTLIB concatenation

The first option above has the advantage that, if the PDS containing the new site command table is given a meaningful name, it may be easier to locate the site command table in the future. The second option has the advantage that it does not require a logon proc change.

If you used the names recommended in the steps above, the PDS containing the new site command table is named SYS2.EZISPF.R18.NEW.SITECMDS and the member name assigned to the new site command table is SITECMDS.

If you want to copy the new site command table into a library already in the ISPTLIB concatenation, you can use the LISTFILE REXX exec to determine the data sets already concatenated to ISPTLIB. See the member named LFDESC in the EZ-ISPF JCL library for a description of LISTFILE.

The site command table member must be the first member of that name in the PDS's concatenated to ISPTLIB. This means that, unless you have a good reason for using some other library in the ISPTLIB concatenation, the safest thing to do

is copy the site command table member into the first library in the ISPTLIB concatenation. Once you have determined which library in the ISPTLIB concatenation you are copying to, you can use ISPF option 3.3 to perform the copy.

Once you have added the new site command table to the ISPTLIB concatenation, you need to define your new site command table to ISPF. Since the method for doing this involves modifying the ISPF configuration table, you will have to refer to the ISPF Planning and Customization Guide for information on how to do this. Once you have modified the ISPF configuration table as required to define the new site command table to ISPF, you can proceed to Step C.7 below.

Step C.7 - Verify Site Command Table

In this step, you will validate the implementation of the newly created or updated site command table.

Logoff of TSO and then log back on again. Go to ISPF option 3.9 and enter the application id assigned to the site command table.

If you created a new site command table and followed the recommendation in Step C.5, the applid for the newly created site command table will be 'SITE'.

While viewing the site command table, you should be able to see the EZ-ISPF commands (EZ and/or EZS) which you added to the site command table. If not, you need to go back and review and/or repeat installation Steps C.1 through C.7.

Appendix D - SMP/E Install

This appendix describes how you can install EZ-ISPF using SMP/E.

Please note that the use of SMP/E is completely optional. Unless your installation has a standard requiring the use of SMP/E or you are a knowledgeable SMP/E user, use of SMP/E is not recommended. If you are required to use SMP/E due to an installation standard and are not a knowledgeable SMP/E user, then it is recommended that you find an experienced SMP/E user to assist you.

You can install EZ-ISPF into existing SMP/E data sets or create new ones for EZ-ISPF. The EZ-ISPF Support Group recommends that you create a new SMP/E environment for each release of EZ-ISPF; this will facilitate an easier transition between releases.

Although it is not recommended, if you want to use an existing SMP/E environment for this release of EZ-ISPF, skip Steps D.1 through D.3 below and proceed to Step D.4. Otherwise, continue with Step D.1.

Step D.1 - SMPALLOC

In this step, you will allocate the EZ-ISPF target and distribution libraries.

You do this by using the JCL in member SMPALLOC in the EZ-ISPF JCL library. Be sure to read the comments in this JCL member before submitting the job for execution.

The expected return code for this job is 0.

Step D.2 - SMPCSI

In this step, you will allocate the CSI and other SMP/E data sets required for EZ-ISPF.

You do this by using the JCL in member SMPCSI in the EZ-ISPF JCL library. Be sure to read the comments in this JCL member before submitting the job for execution.

The expected return code for this job is 0.

Step D.3 - SMPDDEF

In this step, you will define zones and add DDDEF entries to the CSI.

You do this by using the JCL in member SMPDDEF in the EZ-ISPF JCL library. Be sure to read the comments in this JCL member before submitting the job for execution.

The highest expected return code for this job is 4.

Step D.4 - SMPRECV

In this step, you will perform the SMP/E RECEIVE function on the EZ-ISPF FUNCTION SYSMOD.

You do this by executing the JCL in member SMPRECV in the EZ-ISPF JCL library. Be sure to read the comments in this JCL member before submitting the job for execution.

The expected return code for this job is 0.

Step D.5 - SMPAPPLY

In this step, you will perform the SMP/E APPLY function on the EZ-ISPF FUNCTION SYSMOD.

You do this through use of the JCL in member SMPAPPLY in the EZ-ISPF JCL library. Be sure to read the comments in this JCL member before submitting the job for execution.

The expected return code for this job is 0.

Step D.6 - SMPACCEPT

In this step, you will perform the SMP/E ACCEPT function on the EZ-ISPF FUNCTION SYSMOD.

You should now edit and then submit the JCL in member SMPACCEPT in the EZ-ISPF JCL library. Be sure to read the comments in this JCL member before submitting the job for execution.

The highest expected return code for this job is 4.

Appendix E - Modifying IBM ISPF Panels

You can use the information in this appendix to redo the IBM ISPF panel modifications required for EZ-ISPF. This should only need to be done if you install a new version of ISPF which includes significant changes to the following IBM ISPF panels:

ISRUDA1 ISRUDA2S ISRUDLP ISRUMC1 ISRUMC2A ISRUMC2B
ISRECPY1 ISRECRA1 ISRURSP ISRUOLP1 ISRSSNEW ISRSSOLD
ISRSEPRI ISRSFSPR ISRSFPRI ISRSFCON

Step E.1 - Allocate Panel Library

In this step, you will allocate the panel library used to store the modified IBM ISPF panels. The recommended data set name for this library is as follows:

SYS2.EZISPF.R18.ISPFrr.PANELS

where rr is the release number for the release of ISPF containing the panels to be modified.

JCL to allocate this library can be found in member **EZAP** in the EZ-ISPF JCL library. Be sure to read the comments in this JCL member before submitting the job for execution. Once this job executes with a return code of zero you can proceed to Step E.2.

Step E.2 - Modify IBM ISPF Panels

In this step, you will modify copies of the IBM ISPF panels listed above.

Please note that you will be making modifications to copies of these panels. The original version of these panels as received from IBM will not be modified.

The required panel modifications will be performed by REXX exec EZPNLS, which is stored in the EZ-ISPF JCL library. This REXX exec will automatically make the needed panel modifications and will ensure that they are done correctly. The changes must be applied to the versions of the IBM panels that you are currently using. These will be the first versions of these panels found by EZPNLS in the ISPLIB concatenation.

EZPNLS will search for the required IBM panels in the PDS's currently allocated to DDname ISPLIB. When it finds the first PDS containing the required panels, you will be given the opportunity to confirm that this PDS is the one containing the IBM panels to be modified. In most normal ISPF environments, you will want to confirm the use of this PDS as the one containing the IBM panels to be modified. However, in case you are using an unusual configuration, you will also be given the opportunity to specify some other PDS as the one containing the IBM panels to be modified.

You will also need to specify the name of the PDS where the modified panels are to be stored. This library was allocated in Step E.1 above. If you used the data set name recommended in Step E.1, the data set name for this library is as follows:

SYS2.EZISPF.R18.ISPFrr.PANELS

where rr is the release number for the release of ISPF containing the panels to be modified.

You must execute the EZPNLS REXX exec under ISPF using the TSO EXEC command. To do this, go to ISPF option 6 and type the command below:

EXEC 'SYS2.EZISPF.R18.JCL(EZPNLS)'

and press ENTER. Read and reply to each prompt issued by the EZPNLS exec.

Once EZPNLS indicates that it has completed processing, you can examine the modifications that were made by typing 'F EZ-ISPF' in the command line and pressing ENTER while browsing the modified panels.

Please note that the modified IBM panels produced by EZPNLS for any one particular release of ISPF must be used when executing EZ-ISPF under that release of ISPF. Using modified IBM panels from a different release of ISPF may produce erroneous results.

Appendix F - Modifying Options Table

You can use the information in this appendix to modify the EZ-ISPF options table. This should only be done if, after reviewing member **OPTDESC** in the EZ-ISPF JCL library, you determine that modifications are required.

If you do want to modify the EZ-ISPF options table, then you should review and change member **XEZOPTS** in the EZ-ISPF Source Library. This member contains the source code for the EZ-ISPF options table.

After making the necessary changes to the XEZOPTS source member, you need to reassemble and relink the options table. If you are not using SMP/E for the install, you do this by using the JCL in member **EZOPTASM** in the EZ-ISPF JCL library. Be sure to read the comments in this JCL member before submitting the job for execution.

If you are using SMP/E for the install, you must RECEIVE and APPLY the XEZOPTS changes as a USER MOD. You do this by using the JCL in member **SMPOPTS** in the EZ-ISPF JCL library. Be sure to read the comments in this JCL member before submitting the job for execution.

Appendix G - Advantages of Using EZ-ISPF

This appendix contains a summary of the advantages of using EZ-ISPF. If needed, it can be used as the basis for a written summary for or a presentation to upper management.

Saves Time/Improves Productivity

EZ-ISPF saves time and improves the productivity of each ISPF user by:

● Saving Keystrokes

- Uses cursor position to designate item or "target string" to be processed. Target string can be any item (data set name, PDS member name, etc.) appearing on any ISPF display panel.
- Automatically determines target string type from context.
- Provides default processing for over two dozen different types of target strings.
- If item to be processed appears on screen and you want to use default processing, just position cursor and press PF key which invokes EZ-ISPF - No typing required!
- Provides for data set id of 8 characters or less to be assigned to each frequently used data set name. You can then process the data set by specifying the data set id rather than typing the entire name. So you save keystrokes each time you access a frequently used data set.
- Provides one or two character invocation parameter to invoke the most frequently used ISPF options. For example, instead of typing =3.1 to go directly to ISPF option 3.1, you type invocation parameter 1. So you save additional keystrokes almost every time you invoke an ISPF option.
- Warns you about making changes to a protected data set as soon as you access the data set. ISPF normally waits until you have wasted time typing changes and attempt to save the changes to let you know that you are not authorized to update the data set.

- Provides additional options which allow:
 - o Immediate job submission - Submit job without editing, browsing, or typing data set name containing JCL.
 - o Single character invocation of SDSF with jobname as optional target string.
 - o Several ways to relate a PDS member name to a single PDS or a set of concatenated PDS's - without typing the PDS data set name(s).
 - o Specification of data set and/or member name to be copied from, copied to, created, replaced, searched, or compared without typing data set name or member name.

- **Fast Access To Recently Used Data Set or Member Names**

EZ-ISPF provides a History List showing target string or data set id specified the last 32 times EZ-ISPF was invoked. You can then process any name shown in the History List without retyping it.

This saves research needed to determine or remember recently used data set or member names as well as saving additional keystrokes.

- **Fast Access To Frequently Used Data Sets or Member Names**

In addition to processing a frequently used data set via the assigned data set id, you can also list and process all data set names that have been assigned to a data set id.

You can also create and process both dynamic and static data set and member name lists.

EZ-ISPF even allows you to process data set and member name lists provided by ISPF itself or by any other ISPF product.

So you can process any data set or member name on any of these types of lists without retyping the name.

These facilities save research needed to determine or remember frequently used data set or member names as well as saving additional keystrokes.

- **Eliminating System Interactions**

EZ-ISPF eliminates the need to split the screen or invoke a menu panel when you want to request some other ISPF option and then return to the current display.

It also allows certain ISPF options to be processed without displaying the associated ISPF menu panel.

This saves one system interaction almost every time EZ-ISPF is invoked and, when the menu panel is normally shown both before and after the requested processing has taken place, it sometimes saves two system interactions.

Intangible Benefits

With EZ-ISPF, users spend less time on tedious tasks involving typing, ISPF navigation, research to determine data set and member names to be processed, waiting for the system to respond, etc. They also spend less time correcting errors due to typing mistakes.

All of this reduces fatigue and leads to greater job satisfaction.

This is reflected in the nearly unanimous opinion of experienced EZ-ISPF users who say they "never want to use ISPF again without having EZ-ISPF available."

Ease of Use/Training

- **Invocation Parameters "Mirror" Existing ISPF Options**

There are no complicated commands to learn or type. If you are familiar with ISPF menus and options, you will be able to begin using EZ-ISPF invocation parameters immediately.

- **Acts As "Front-End" To ISPF**

You can still use ISPF as you always have. EZ-ISPF does not replace ISPF; it supplements ISPF in order to speed it up. So there is no big "learning curve" and no "conversion date" where every user has to be "up to speed". Users can begin to use EZ-ISPF a little at a time and learn as they go.

- **Provides On-Line Demo and On-Line Tutorial**

Users can learn how to use EZ-ISPF using the on-line demo and the on-line tutorial. No formal training sessions are needed.

Comparison To Other Tools

Due to the excessive typing and the number of system interactions required to get certain tasks done under ISPF, IBM has made a few productivity enhancements to ISPF over the years, a number of ISPF productivity tools been developed by independent software vendors, and quite a few installations have developed their own in-house tools to deal with this problem.

However, none of these products or approaches have been particularly successful. There are several reasons for this:

- **These Products Are Not Comprehensive**

They generally address only a limited portion of ISPF (like Edit and Browse).

They offer target string processing or data set id processing but not both.

EZ-ISPF is comprehensive. It addresses all of the most frequently used portions of ISPF and it offers both target string and data set id processing.

- **Some Replace Significant Portions Of ISPF**

This means they require too much retraining and do not allow you to take advantage of future ISPF enhancements.

EZ-ISPF supplements ISPF rather than replacing it.

- **Some Require Additional System Interactions**

This means they are slow. An example of this is the IBM ISPF enhancement involving data set reference lists. To use a list, you must open and then process the list. This requires additional system interactions.

Although EZ-ISPF offers facilities like the ISPF data set reference list (and even allows you to process the ISPF data set reference list using EZ-ISPF), it offers many other time-saving facilities over and above simple "reference lists".

- **They Are Not Flexible**

For example, for a member name like PAY1 in the line below:

```
SYS2.PAYROLL.SOURCE(PAY1)
```

some other products might allow target string processing for PAY1 but would restrict it so that it can be processed only in conjunction with data set SYS2.PAYROLL.SOURCE.

EZ-ISPF allows for PAY1 to be processed as a target string in conjunction with SYS2.PAYROLL.SOURCE or with virtually any other PDS that you might want to specify. This is just one example of the type of flexible processing provided by EZ-ISPF.

Summary

EZ-ISPF is more flexible, more comprehensive, and more powerful than any other ISPF productivity product. It supplements rather than replacing ISPF. This means that it is extremely easy to learn and that it requires no massive "conversion" or "retraining" effort. It has innovative productivity features that no other ISPF productivity product offers. These productivity features have a tremendous impact on the speed and efficiency of ISPF, all of which makes EZ-ISPF the ISPF productivity product of choice.

Appendix H - Cost Justification

Assumptions:

- 10 programmers, operators, or other ISPF users who are at least moderate users of ISPF
- \$50 per man-hour
- EZ-ISPF saves an average of 30 minutes per day per user

With these assumptions, the approximate monthly savings would be \$1250 per month as shown by the calculation below:

$$.5 \text{ hrs/day} \times 20 \text{ days/month} \times 10 \text{ users} \times \$50/\text{hr} = \$5000/\text{month}$$

With these assumptions, the initial licence fee for EZ-ISPF would be paid for in 2 or 3 months and EZ-ISPF would then save about \$60,000 per year.

The table below shows approximate annual savings based on the number of at least moderate users of ISPF:

Number of Moderate ISPF Users	Approximate Annual Savings
-----	-----
10	\$60,000
20	\$120,000
30	\$180,000
50	\$300,000
100	\$600,000

Appendix I - User Feedback

The EZ-ISPF support group at Referential Systems wants to ensure that EZ-ISPF is as useful to you as possible. If you have an idea for enhancing or extending the capabilities of EZ-ISPF, or if you want to report a program defect or bug, please send your comments to us using e-mail, fax, or a voice communication. Please include your name, the name of your enterprise, and an e-mail address or phone number that we can use to acknowledge your request.

Contact us using one of the methods below:

Via postal mail: EZ-ISPF Support Group
Referential Systems
P. O. Box 452
Vienna, Virginia 22183-0452
USA

Via e-mail: support@ezispf.com

Via telephone: (800) 377-2650

Web site: <http://www.ez-ispf.com>