Connect:Express™ OS/390

Utilities Guide

Version 4.2.0

Connect:Express for OS/390 Utilities Guide
Version 4.2.0
First Edition

This documentation was prepared to assist licensed users of the Connect:Express system ("Sterling Commerce Software"). The Sterling Commerce Software, the related documentation and the information and know-how it contains, is proprietary and confidential and constitutes valuable trade secrets of Sterling Commerce, Inc., its affiliated companies or its or their licensors (collectively "Sterling Commerce"), and may not be used for any unauthorized purpose or disclosed to others without the prior written permission of Sterling Commerce. The Sterling Commerce Software and the information and know-how it contains have been provided pursuant to a license agreement which contains prohibitions against and/or restrictions on its copying, modification and use. Duplication, in whole or in part, if and when permitted, shall bear this notice and the Sterling Commerce, Inc. copyright legend.

Where any of the Sterling Commerce Software or Third Party Software is used, duplicated or disclosed by or to the United States government or a government contractor or subcontractor, it is provided with RESTRICTED RIGHTS as defined in Title 48 CFR 52.227-19 and is subject to the following: Title 48 CFR 2.101, 12.212, 52.227-19, 227.7201 through 227.7202-4, FAR 52.227-14(g)(2)(6/87), and FAR 52.227-19(c)(2) and (6/87), and where applicable, the customary Sterling Commerce license, as described in Title 48 CFR 227-7202-3 with respect to commercial software and commercial software documentation including DFAR 252.227-7013(c) (1), 252.227-7015(b) and (2), DFAR 252.227-7015(b)(6/95), DFAR 227.7202-3(a), all as applicable.

The Sterling Commerce Software and the related documentation are licensed either "AS IS" or with a limited warranty, as described in the Sterling Commerce license agreement. Other than any limited warranties provided, NO OTHER WARRANTY IS EXPRESSED AND NONE SHALL BE IMPLIED, INCLUDING THE WARRANTIES OF MERCHANTABILITY AND FITNESS FOR USE OR FOR A PARTICULAR PURPOSE. The applicable Sterling Commerce entity reserves the right to revise this publication from time to time and to make changes in the content hereof without the obligation to notify any person or entity of such revisions or changes.

References in this manual to Sterling Commerce products, programs, or services do not imply that Sterling Commerce intends to make these available in all countries in which Sterling Commerce operates.

Printed in the United States of America.

Copyright © 2003, 2007. Sterling Commerce, Inc. All rights reserved.

Connect:Express is a registered trademark of Sterling Commerce. All Third Party Software names are trademarks or registered trademarks of their respective companies. All other brand or product names are trademarks or registered trademarks of their respective companies.

Contents

Preface

The *Connect:Express OS/390 Utilities Guide* is for programmers and network operations staff who automate and organize file transfer operations with Connect:Express OS/390.

Chapter Overview

The *Connect:Express OS/390 Utilities Guide* is organized into the following chapters and appendices:

Chapter/Appendix	Description
1 The Utilities Package	This chapter provides an overview of the Utilities Package for Connect:Express and discusses how to integrate the utilities into the monitor configuration and the user environment.
2 Automatic Monitor Checking and Alert Generation	This chapter describes the utilities that enable you to send end of transfer notifications, check connections and operations, and manage a remote Partner.
3 Integrating Transfer Operations into the User Environment	This chapter describes the utilities that enable you to activate user procedures, delete checkpoint files, purge transfer requests, and manage Partner and File directories.
A Supporting Lists, Tables, and Examples	This appendix provides reference lists of action keywords, dynamic variables, and error and severe messages. There are also examples of a SYSUE1 file and a SYSOUT file.

Connect:Express Documentation

Connect:Express documentation consists of the following manuals:

❖ The *Connect:Express OS/390 Version 4.2 Release Notes* lists maintenance updates and any important notes.

❖ The *Connect:Express OS/390 4.2 Installation Guide* describes the planning and installation of Connect:Express.

❖ The *Connect:Express OS/390 4.2 User Guide* includes general information on using the TSO/ISPF interface, and serves as a reference of user and environment commands.

❖ The *Connect:Express OS/390 4.2 Utilities Guide* describes the optional Utilities package that you can integrate with Connect:Express.

❖ The *Connect:Express OS/390 4.2 FTP Guide* provides you with the information that you need to use Connect:Express with the FTP protocol.

❖ The *Connect:Express OS/390 4.2 Administration Guide* provides detailed information about transfer operations for system administrators and other advanced users of Connect:Express.

❖ The *Connect:Express OS/390 4.2 Options Guide* provides information about the CICS, IMS, and RJE interfaces available for Connect:Express.

❖ The *Connect:Express OS/390 4.2 PeSIT User Fields Guide* describes how you can exchange the PeSIT Pi37 and Pi99 fields with any PeSIT software.

❖ The *Connect:Express HTTP Option Implementation Guide* provides you with the information that you need to implement HTTP access to the Connect:Express OS/390 repository.

❖ The *Connect:Express OS/390 4.2 Etebac3 User Guide* provides you with the information that you need to use Connect:Express with the Etebac3 protocol.

❖ The *Connect:Express OS/390 4.2.0 SSL Guide* includes general information on implementing secured file transfers.

❖ The *Connect:Express OS/390 4.2.0 Sysplex Supervision Guide* includes general information on implementing a group of Connect:Express Plex managers under control of a Connect:Express Plex supervisor.

Getting Support for Sterling Commerce Products

Sterling Commerce provides intuitive technical products and superior Help and documentation to enable you to work independently. However, if you have a technical question regarding a Sterling Commerce product, use the Sterling Commerce Customer Support Web site.

The Sterling Commerce Customer Support Web site at *www.sterlingcommerce.com* is the doorway to Web support, information, and tools. This Web site contains several informative links, including a solutions database, an issue tracking system, fix information, documentation, workshop information, contact information, sunset and retirement schedules, and ordering information. Refer to the Customer Support Reference Guide at *www.sterlingcommerce.com/customer/tech_support.html* for specific information on getting support for Sterling Commerce products.

Conventions Used in This Guide

The *Connect:Express OS/390 Utilities Guide* uses certain notational conventions. This section describes the conventions used in this guide.

Convention	Description
UPPERCASE LETTERS	Uppercase letters in the command format indicate that you type in information as shown.
Lowercase letters	Lowercase letters or words in commands or syntax boxes require substitution by the user. For example, index1.index2.PARMLIB indicates that you must provide the first and second indexes of the string. "PARMLIB" is mandatory.
Bold Letters	Bold print in syntax boxes indicates Connect:Express commands and required parameters. For example, PLEX=N indicates that the parameter PLEX must be set to N.

Convention	Description
Underlined Letters	Underlining indicates default values for parameters and subparameters. For example, PLEX=Y|<u>N</u> specifies that the default for PLEX is N.
Vertical Bars (|)	Vertical bars indicate that you can supply one of a series of values separated by the vertical bars. For example RUN=H|C specifies that H or C is valid.
`Monospaced characters` (characters of equal width)	`Monospaced characters` represent information for screens, commands, Processes, and reports.
Punctuation	Code all commas and parentheses as they appear.
£ or #	The Pound character (£) and the hash character (#) are equivalent.

Chapter 1

The Utilities Package

This chapter provides an overview of the Utilities Package for Connect:Express and discusses how to integrate the utilities into the monitor configuration and the user environment.

Overview

This guide covers the Utilities Package for Connect:Express OS/390 and is for system administrators or other advanced users of Connect:Express.

Note: You must have a license to use the Utilities Package. Contact your Sterling Commerce representative for information about obtaining a license.

The Utilities Package enables you to integrate file transfer operations in your environment and adds a layer of high-level functionality to the monitor's basic file transfer services. The Utilities Package option is under the control of the asset protection file, and includes programs and parameter files that provide monitor management and file transfer operations management. The Utilities Package provides functionality in two areas:

❖ Automatic Monitor Checking and Alert Generation

❖ Integrating Transfer Operations into the User Environment

Each utility receives input parameters and/or files and then delivers actions, notifications, and reports. WTO Messages are issued so that you can store them in a SYSPRINT file or trap them with Netview. Some utility programs are executed in their own address space, while others are executed in one of the Connect:Express address spaces. The TOM and APM are used for PeSIT and ODETTE transfers, and the AFM and EAS are used for FTP transfers.

Integrating the Utilities Package with Connect:Express

To integrate the Utilities Package into the Monitor configuration, you must position new parameters and specify some functionality in the monitor configuration through the TSO/ISPF operator interface or in the SYSIN file.

Utilities functions are based on standard Connect:Express interfaces--the API interface and the user exit interface. With the API interface, some functions use subsystem interface services such as DISPLAY, TRANSFER, and MODIFY. With the user exit interface, some functions are implemented by standard exits such as, journal, beginning and end of transfer, connection, and selection exits.

You can also integrate the Utilities Package into the user environment so that user procedures start automatically. You can implement external jobs or OS/390 commands for permanent or occasional checking, and setup a utility to send alert messages to a control process. In addition, you can work with vendor utilities like IDCAMS or TSO.

If you use any of the Programs described in the following sections, you will need to define parameters in the Connect:Express configuration or in specific files.

Trapping WTO Messages with NETVIEW

WTO messages are sent to the JES log of the job, not to the OS/390 console, and Netview can trap these messages. If you use this feature, you must update the NETVIEW message table with the specific message headers from the Utilities package modules. Messages display the subsystem name followed by three letters that represent the origin module name. For example, if the subsystem name is TOMP, you would see the following message headers:

```
TOMPSM1  – TOMPJN1  – TOMPRC1
      headers from:
P1GFISM1 - L1GFIJN1 - L1GFIRC1
```

Redirecting Messages to SYSPRINT Files

WTO messages can also be redirected to SYSPRINT files. When a program is executed in a Connect:Express address space, the SYSPRINT file is dynamically created from a prefix and a time stamp during the transfer. If a program is executed on its own, you must add a SYSPRINT DD card to the JCL of the address space where it is executed.

Adding Extra JCL DD Cards

To integrate user processes with transfer operations, you may need to add extra DD cards to the JCL of the address space where it is executed, either TOM, APM, AFM, or EAS. These cards correspond to SYSPRINT files, JCL skeletons, program libraries, trace files, or snap files for debugging. The following table shows the corresponding processes and the DD cards required. The last column shows how the process can be invoked using the general exit L1GFIUE1.

DD Card	Process	L1GFIUE1†
SYSPRINT	P1GFIIDC, P1GFIID0, P1GFICP1	$PGMJ$
SYSJOB	P1B2PJCL	JCL

† L1GFIUE1 and 'xxx' keywords are explained in Chapter 3, Integrating Transfer Operations into the User Environment.

DD Card	Process	L1GFIUE1†
SYSPRT	P1B2PJCL	JCL
SYSPCH	P1B2PJCL	JCL
SYSPR20	P1GFICP1, P1GFIRP1, P1B2P300, P1B2PREQ, P1B2PRQ2	300, REQ, JOB
WTOPRINT	P1GFICP1, P1GFIRP1	
SYSEXEC	REXX Procedure	$REXX$
SYSTSPRT	REXX Procedure	$REXX$
SYSTSIN	REXX Procedure	$REXX$

† L1GFIUE1 and 'xxx' keywords are explained in Chapter 3, Integrating Transfer Operations into the User Environment.

Chapter 2

Automatic Monitor Checking and Alert Generation

This chapter describes the utilities that enable you to send end of transfer notifications, check connections and operations, and manage a remote Partner.

Overview

The Utilities Package includes a program that produces Statistics by analyzing data from the transfer results of one or more monitors. There is another program that you can setup to issue notification messages at the time of an error, or you can monitor Connect:Express by periodically scanning journals and logs. The following utilities can be run externally or online. Each function and module is described in its own section following the table.

Function	Module	Description
Producing Statistics from Journal Files	P1GFIJS2	External tool: Produces statistics based on one partner (or a generic partner), one file (or a generic file), a date/time interval, or transfer failures. Statistics can be made from one or several Journal files, from one or several monitors.
Sending End of Transfer Notifications	L1GFIJN1	Journal exit: Notifications can be sent to NETVIEW at end of transfer from the journal recording process.
	L1GFITS1	External tool: A notification message is sent to the first user from a list of users logged on to TSO.
Checking Connections and Transfer Selections	L1GFICN1	Connection and selection exit: Enables you to detect immediately session failures or transfer negotiation failures.
Checking General Operations	P1GFISM1	External tool: Performs a periodical checking of Connect:Express resources and sends alerts to Netview or to a user exit.
Managing a Remote Monitor	L1GFIRC1	Beginning and end of transfer exit: You can establish a session with a remote Connect:Express and monitor it remotely or check the link.

Producing Statistics From Journal Files (P1GFIJS2)

This utility processes input from one or more journal files. It sorts and merges the files in a sequential file output and then removes any duplicates. You can produce Statistics by Partner name or root, File name or root, or for a designated date and time range. You can also request Statistics about failed transfers. This program can be executed in its own address space.

Note: Journal files can be from different monitors.

Implementing P1GFIJS2

The P1GFIJS2 gets its input parameters from the EXEC PARM field. These parameters include the selection criteria and tell Connect:Express how to format the results. If the program finds a SYSJNL DD card, it will process a unique VSAM journal file. If it finds a CUMULS DD card, it will process a sequential file that results from a previous merge procedure.

```
//CUMJNLS  DD      DISP=SHR,DSN=INDEX1.P1GFIJS2.TEMPIN
               - or -
//SYSJNL   DD      DISP=SHR,DSN=INDEX1.TOM2V410.SYSJNL
```

Note: Connect:Express looks at the SYSJNL card first.

Entering Selection and Output Criteria

You must specify the criteria for the statistics that you want as well as the format for the output. You enter your options in the PARM field, as shown in the following example. Positions of the keywords are fixed.

```
PARM field: 'OPT1',STR='aammjj',END='aammjj',DIR='d', 'OPT2'
```

The following table describes the parameters that you use to enter your selection criteria.

Parameter	Format	Description
OPT1	4 Characters alphanumeric	FULL: header, statistics and Journal record FAIL: header, statistics and Journal record (failures only) STAT: statistics only HEAD: header only
STR	Date – yymmdd	Beginning of transfer date: any transfer started after this date (included) that meets the next condition is processed.
END	Date – yymmdd	End of transfer date: any transfer ended before this date (included) that meets the previous condition is processed.
DIR	1 Character alphanumeric	T = Transmission R = Reception * = Both direction
OPT2 optional	1 to 12 Characters alphanumeric	SPN = Partner name (8 characters) or generic root ('GEN*' or 'GEN' are equivalent). For example, SPN=NEWYORK. SFN = File name (8 characters) or generic root ('GEN*' or 'GEN' are equivalent). For example, SFN=FILE00*

Process Description

P1GFIJS2 is an example of JCL in the *SAMPOPT* library. Steps 1-3 apply when you have several files. For one file, only step 3 applies. The process flows as follows:

1. Connect:Express allocates a sequential file. The space must be sufficient for all the records expected from the journal files.

2. The REPRO IDCAMS and sort procedure is run, then the L1GFIJS2 (E35) module removes any duplicates.

3. The P1GFIJS2 module processes the selection criteria and creates statistics from the sequential file built in steps 1 and 2, according to the OPT1 option.
   OR
   The P1GFIJS2 module processes the selection criteria and creates statistics from one VSAM journal file, according to the OPT1 option.

Viewing the Results

You can format the results with the OPT1 option. The header for all options shows global information, the statistics (FULL, FAIL, and STAT options) show general numbers corresponding to your selection criteria, and the FULL and FAIL options list all selected journal records. The return code indicates if the process was successful or not. The following table lists the possible return codes.

Code	Description
0	Successful
4	'WARNING' – an anomaly has been detected during the process or no record was found. In case of OPT1=HEAD, no record was looked for, and the return code is always 4.
8	ERROR detected in the PARM field
12	FATAL ERROR

WTO Messages

WTO messages are time stamped and can be redirected to a file called WTOPRINT. Journal records can also be redirected to a file called EXTJNL, as shown in the following example.

```
//WTOPRINT DD SYSOUT=*
//EXTJNL   DD SYSOUT=*        FBA LRECL=133
```

OPT1 = Any Value

The header lines of the WTOPRINT file display for any OPT1 value. The first line indicates the last update of the journal file. The next line indicates the statistics mode, multiple Journal files, or one VSAM file. Then

Connect:Express displays the selection criteria and general counts, the total number of records, the number of selected records, the number of successful records, and the return code.

```
01115 182102 P1GFIJS2 UPDATE DATE : 000403
01115 182102 P1GFIJS2 NOTICE  CUMJNLS  JOURNAL  USED          (SEQ.,SORTED,NO-DUP)
01115 182102 P1GFIJS2 NOTICE  HEAD     JOURNAL FOR STR=010401
01115 182102 P1GFIJS2 NOTICE  HEAD     JOURNAL FOR END=010430
01115 182102 P1GFIJS2 NOTICE  HEAD     JOURNAL FOR DIR=*
01115 182102 P1GFIJS2 NOTICE  HEAD      JOURNAL END RC00000004 TOTAL RECORDS=0000
01115 182102 P1GFIJS2 NOTICE  HEAD      ENTRIES MATCH=00000000   ENTRIES OK=0000

--------------------------------------------------------------------------------
*        *TYPE                  COUNT OK  /  MATCHED *  *D*  *PARTNER *
*                               *BEG.DATE       *END-DATE        *DUR.SEC.*RST
--------------------------------------------------------------------------------
```

Note: If OPT1=HEAD, the return code is always equal to 4.

OPT1 = FULL or FAIL

If OPT1 = FULL or FAIL, the header line displays in the WTOPRINT file and the journal record lines appear in the EXTJNL file. The first lines of the EXTJNL file display the job name, date and time, and selection criteria (SEL) parameters in the following order: direction, partner, file, date/time interval. The following screen shows an example.

```
USR4JS2  01115   18210
SEL:          *      SCI                     010401       010430
```

Each journal record is displayed with two lines. Some lines are truncated in the example below.

```
*FILE    *DATA SET NAME                             *T *D*PP*PARTNER *T*L*ST*
*        *REQ.NUM.*FROM    *C*Y*BEG.DATE/HOUR    *END-DATE/HOUR    *DURATION*RST
```

Line 1: *File *Physical Name *File Type *Direction *Presentation *Partner *Partner type *Link *Session Table *Remote Physical Name *TRC *PRC

Line 2: *Request Number *Requestor *Class *Priority *Date/Time Begin. *Date/Time End *Elapse *Retries *Kbytes protocol *Kbytes File *Kbytes/Second * Number of records *Messages Sent/Received *subsystem

The list of journal records displays total counts and average counts as shown in the following example. Some lines are truncated in the figure below.

```
*        *TYPE                  COUNT OK  /  MATCHED *  *D*  *PARTNER *
*        *TOTAL OF TRANSFERS OK:     005 /      005 *  ***  *        *

*        *TOTAL FAIL:      0  *BEG.DATE        *END-DATE        *DUR.SEC.*RST
*                            *010401          *010430          *        4*-
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
*        *TYPE                  COUNT OK  /  MATCHED *  *D*  *PARTNER *
*        *AVERAGE OF TRANSFERS OK:   005 /      005 *  ***  *        *

*                            *BEG.DATE        *END-DATE        *DUR.SEC.*RST
*                            *010401          *010430          *        4*-
```

Line 1: *Total counts *Transfers successfully ended* *Transfers matching the criteria* *Direction criteria* *Partner criteria*

Line 2: *Total of transfers that failed among the total matching (only for Total) *Begin date criteria* *End Date criteria* *Transfer Elapse (seconds)* *Number of retries* *Protocol Kbytes* *File Kbytes* *Kbytes/second* *Number of Records* *Messages Sent/Received* and *Subsystem Name*

OPT1 = STAT

When OPT1 = STAT, the Header lines display in the WTOPRINT file and total counts appear in the EXTJNL file, as shown in the following example.

```
  *        *TYPE                 COUNT OK  /  MATCHED *  *D*  *PARTNER *
  *        *TOTAL OF TRANSFERS OK:     005 /     005 *  *** *        *

  *        *TOTAL FAIL:      0  *BEG.DATE        *END-DATE        *DUR.SEC.*RST
  *                            *010401          *010430          *       4*-
```

Examples of Statistics

OPT1 = HEAD

When OPT1 = HEAD, only the header is displayed, as shown in the following example of a WTOPRINT file.

Note: If the beginning date = 000000 and end date = 999999, then all transfers are included.

```
  01115 182102 P1GFIJS2 UPDATE DATE : 000403
  01115 182102 P1GFIJS2 NOTICE  CUMJNLS  JOURNAL  USED        (SEQ.,SORTED,NO-DUP)
  01115 182102 P1GFIJS2 NOTICE  HEAD     JOURNAL FOR STR=010401
  01115 182102 P1GFIJS2 NOTICE  HEAD     JOURNAL FOR END=010430
  01115 182102 P1GFIJS2 NOTICE  HEAD     JOURNAL FOR DIR=*
  01115 182102 P1GFIJS2 NOTICE  HEAD     JOURNAL END RC00000004 TOTAL RECORDS=0000
  01115 182102 P1GFIJS2 NOTICE  HEAD     ENTRIES MATCH=00000000   ENTRIES OK=0000

  --------------------------------------------------------------------------------
  *        *TYPE                 COUNT OK  /  MATCHED *  *D*  *PARTNER *
  *                            *BEG.DATE        *END-DATE        *DUR.SEC.*RST
  --------------------------------------------------------------------------------
```

In this example, the first line indicates the last modification date of the Journal file, and the second line indicates that statistics are from multiple journal file (CUMJNLS). There is no total count and the return code is 4.

OPT1 = STAT

When OPT1 = STAT, header and total counts are provided, as shown in the following example of a WTOPRINT file.

```
  01115 182102 P1GFIJS2 UPDATE DATE : 000403
  01115 182102 P1GFIJS2 NOTICE  SYSJNL   VSAM    JOURNAL USED
  01115 182102 P1GFIJS2 NOTICE  STAT     JOURNAL FOR STR=000000
  01115 182102 P1GFIJS2 NOTICE  STAT     JOURNAL FOR END=999999
  01115 182102 P1GFIJS2 NOTICE  STAT     JOURNAL FOR DIR=*
  01115 182102 P1GFIJS2 NOTICE  STAT     JOURNAL END RC00000000 TOTAL RECORDS=00001395
  01115 182102 P1GFIJS2 NOTICE  STAT     ENTRIES MATCH=00001393   ENTRIES OK=00000992
```

In this example, the second line indicates that the statistics were produced from a unique VSAM file. This file contains 1395 records, and 1393 records were selected based on criteria. Of the selected records, 992 were successful transfers. The following screen shows these results in the EXTJNL file.

```
 *        *TYPE               COUNT OK  /  MATCHED *  *D*  *PARTNER *
 *        *TOTAL OF TRANSFERS OK:     992 /     1393 *  *** *        *

 *        *TOTAL FAIL:    401  *BEG.DATE       *END-DATE       *DUR.SEC.*RST
 *                             *010401         *010430         *      4*-
```

OPT1 = FAIL

When OPT1 = FAIL, header and total counts are displayed. Journal records are also listed, as long as at least one return code is not null. The following screen shows an example of the WTOPRINT file.

```
01115 182102 P1GFIJS2 UPDATE DATE : 000403
01115 182102 P1GFIJS2 NOTICE  SYSJNL   VSAM    JOURNAL USED
01115 182102 P1GFIJS2 NOTICE  FAIL     JOURNAL FOR STR=000000
01115 182102 P1GFIJS2 NOTICE  FAIL     JOURNAL FOR END=999999
01115 182102 P1GFIJS2 NOTICE  FAIL     JOURNAL FOR DIR=R
01115 182102 P1GFIJS2 NOTICE  FAIL     JOURNAL END RC00000000 TOTAL RECORDS=00001395
01115 182102 P1GFIJS2 NOTICE  FAIL     ENTRIES MATCH=00001393   ENTRIES OK=00000992
```

OPT1 = FULL

When OPT1 = FULL, all line types are displayed. The following example shows journal records for any SCI* partners during the month of April in a WTOPRINT file.

```
01115 182102 P1GFIJS2 UPDATE DATE : 000403
01115 182102 P1GFIJS2 NOTICE  SYSJNL   VSAM    JOURNAL USED
01115 182102 P1GFIJS2 NOTICE  FULL     JOURNAL FOR STR=010401
01115 182103 P1GFIJS2 NOTICE  FULL     JOURNAL FOR END=010430
01115 182103 P1GFIJS2 NOTICE  FULL     JOURNAL FOR DIR=*
01115 182103 P1GFIJS2 NOTICE  FULL     JOURNAL FOR SPN=SCI
01115 182103 P1GFIJS2 NOTICE  FULL     JOURNAL END RC00000000 TOTAL RECORDS=00001395
01115 182103 P1GFIJS2 NOTICE  FULL     ENTRIES MATCH=00000005   ENTRIES OK=00000005
```

The following example shows the EXTJNL file when OPT1 = FULL.

```
    --------------------------------------------------------------------------------
    USR4JS2  01115    18210
    SEL:          *         SCI                        010401        010430
    --------------------------------------------------------------------------------
    *FILE     *DATA SET NAME                                *T *D*PP*PARTNER *T*L*ST*
    *         *REQ.NUM.*FROM     *C*Y*BEG.DATE/HOUR    *END-DATE/HOUR    *DURATION*RST
    --------------------------------------------------------------------------------
    *RC1LOOPX*INDEX1.SCIPSR.RC1LOOP.D010415.B0601332    *S *R*04*SCIPSR  *T*X*32*
    *        *00001332*SCIPSR  *A*0*01/04/15*12:44:59*01/04/15*12:45:01*00:00:02*-
    --------------------------------------------------------------------------------
    *RC1LOOPX*INDEX1.SCIPSR.RC1LOOP.D010415.B0601335    *S *R*04*SCIPSR  *T*X*32*
    *        *00001335*SCIPSR  *A*0*01/04/15*12:45:14*01/04/15*12:45:23*00:00:09*-
    --------------------------------------------------------------------------------
    *RC1LOOPX*INDEX1.SCIPSR.RC1LOOP.D010415.B0601333    *S *R*04*SCIPSR  *T*X*32*
    *        *00001333*SCIPSR  *A*0*01/04/15*12:45:16*01/04/15*12:45:21*00:00:05*-
    --------------------------------------------------------------------------------
    *RC1LOOPX*INDEX1.SCIPSR.RC1LOOP.D010415.B0601337    *S *R*04*SCIPSR  *T*X*32*
    *        *00001337*SCIPSR  *A*0*01/04/15*12:45:22*01/04/15*12:45:25*00:00:03*-
    --------------------------------------------------------------------------------
    *RC1ECHOS*INDEX1.SCIPSR.RC1ECHOS.D010416.B0700006   *S *R*04*SCIPSR  *T*S*32*
    *        *00000006*SCIPSR  *A*0*01/04/16*16:53:08*01/04/16*16:53:08*-       *-
    --------------------------------------------------------------------------------
    --------------------------------------------------------------------------------
    *         *TYPE                   COUNT OK  /  MATCHED *  *D*  *PARTNER *
    *         *TOTAL OF TRANSFERS OK:       005/     0005 *  *** *         *

    *         *TOTAL FAIL:      0   *BEG.DATE         *END-DATE          *DUR.SEC.*RST
    *                              *010401           *010430                 *     57*-
    --------------------------------------------------------------------------------
    *         *TYPE                   COUNT OK  /  MATCHED *  *D*  *PARTNER *
    *         *AVERAGE OF TRANSFERS OK:    005 /     005 *  *** *         *

    *                              *BEG.DATE         *END-DATE          *DUR.SEC.*RST
    *                              *010401           *010430                 *      4*-
```

Total and average numbers are computed from the selected records. In this example, five records were selected and there were no failures among them.

Sending End of Transfer Notifications (L1GFIJN1)

This program is executed in the TOM address space and it is a Journal exit module. This Journal exit enables you to send an alert each time an error is detected at the end of a transfer. Using this exit, you can also produce statistics during the termination of Connect:Express about work done since the initialization or since the last statistics request. Then, the counts are reset. You can request statistics at any time by transferring the symbolic file 'STATS.'

Implementing L1GFIJN1

This program is a Journal exit and is called by the standard Journal procedure to check the Journal record. The Journal record structure is described by the D1B2PJNL structure provided in the *MACLIB* library. The program is either declared in the SYSIN file, as shown in the following example:

```
UEXJNL=L1GFIJN1
```

Or, as shown below, in the SYSINEXT file of the L1B2PDIX journal exit driver which is in the TOM address space JCL.

```
                              UEXJNL=L1B2PDIX
   //SYSINEXT DD
```

The L1GFIJN1 program is authorized (APF) and WTO messages (ROUTCDE=11) are not sent to the OS/390 console.

Process Description

During initialization, a message is sent. At each end of transfer, whether it was successful or not, a message displays the symbolic file name, the partner name, the request number, date and time, and PRC and TRC return codes. During termination or at end of 'STATS' transfer, total counts are displayed and then erased.

All retries for one transfer are counted. For example, if a transfer is interrupted, restarted and successfully ended, the counts display as follows:

```
TOTAL 2  OK=1  KO=1
```

Viewing the Results

The following table lists the valid Return Codes for this program.

Code	Description
0	Successful
4	'WARNING' – An anomaly has been detected during processing.

WTO Messages

A transfer that ended successfully displays the following values:

```
TRC=0000 - PRC=0000
```

The table below describes the conventions for WTO messages sent by this utility.

Character	Description
????????	Status or flag
++++++++	Numeric field, a number, a count, a code
////////	Alphanumeric, partner name for example

The following table lists the possible messages for this utility.

L1GFIJN1 Messages
`TOM?JN10    INIT` The subsystem name is not yet known at this point during initialization.
`TOM+JN11 //////// //////// ++++++++ TRC+++++ PRC++++ RST+++ ++:++:++` Partner Name File Name Request Number TRC PRC Number of retries End time If the transfer is interrupted, the End time is null.

`TOM+JN12 ERROR                          ?                          (INTERNAL ERROR)`
`                                ? = P - Internal error while passing parameters`
`                                ? = S - Invalid subsystem name (must be TOM?)`

`TOM+JN13    TOTAL ++++++++ OK=++++++++ KO=++++++++ P=++++++++ T=++++++++` During termination, the exit shows the counts for total work done since the last time statistics were processed. TOTAL=count1 OK=count2 KO=count3 P=count4 T=count5 Count1: total number of calls to exit Count2: total number of successful transfers Count3: total number of interrupted or rejected transfers Count4: total number of unsuccessful transfers with PRC not null Count5: total number unsuccessful transfers with TRC not null
`TOM+JN14 STATS TOTAL ++++++++ OK=++++++++ KO=++++++++ P=++++++++ T=++++++++` This message is issued at each 'STATS' transfer.

Sending End of Transfer Notifications to a TSO User (L1GFITS1)

This utility provides a way to send a TSO message to the first user in a list of users logged on to TSO. This program can be executed in its own address space.

Implementing L1GFITS1

You can use the L1GFITS1 program either in batch or in the L1GFIUE1 general exit process as the parameter of a $PGMJ$ keyword.

```
SYSUE1 Example---- $PGMJ$  L1GFITS1 'PARMS'
```

The program receives the EXEC PARM field input, as shown in the following example.

```
   ($VAR$    L1 USR0003,USR0004,USR0005,USR0008)
   L1GFITS1 ULN=(RC0,&L1),MSG=&SSN&CT3&FILENAME&DSN,UER=PSR0008
```

The table below describes the parameters for this utility.

Parameter	Description
MSG	Text of the message (any character or blank). Dynamic keywords are supported when processed by L1GFIUE1. In the example above, &SSN will be replaced by the Connect:Express subsystem name (TOM?), &CT3 will be replaced by the call type (IEX or TEX), and &FILENAME will be replaced by the data set name of the file transferred.
ULN	List of TSO users, the first who is found connected will be sent the message. Using the keyword RC0 avoid return code 4 if no user is connected.
UER	Name of the user to receive the message, if no user on the list is connected.

Viewing the Results

The following table lists the possible Return Codes for the L1GFITS1 utility.

Code	Description
0	Successful
4	'WARNING' – no user from the list is logged on. If this is performed during transfer execution this return code would result in a transfer interruption. If you want to skip the error, use the keyword RC0 in the ULN field.
8	ERROR detected during processing

The screen below shows several examples.

```
BATCH:
  //GO        EXEC    PGM=L1GFITS1,
  //  PARM='MSG=HEY JOE ULN=(USR0003,USR0004,USR0005) UER=USR0009'
  //STEPLIB    DD       DISP=SHR,DSN=INDEX1.TOMV???.LOADLIB
  //SYSUDUMP   DD       SYSOUT=*

L1GFIUE1:
  $PGMJ$    L1GFITS1 ULN=(&L1)  MSG=&SSN &FILENAME &DSN  UER=USR0004
```

Checking Connections and Transfer Selections (L1GFICN1)

L1GFICN1 is a connection and transfer selection exit module that is executed in the TOM, AFM, APM or EAS address space. The L1GFICN1 utility enables you to trap incidents and take immediate actions. It provides information like request number, Partner name, TRC, PRC, NRC, and network link type. An example of the source file, L1USRCN1, is provided in the *SAMPLIB* library.

Implementing L1GFICN1

This program is a standard exit that is called during the connection phase or the file transfer selection phase. When monitoring connections, an exit must be declared in the user connection exit table T1B2PCNT (TSO/ISPF Option 3.3.CNT). Input parameters are from the D1B2PCNX structure. When monitoring transfer selections, you must declare this exit in the user selection exit table T1APMSRT (option 3.3 SRT on the Main Menu). Input parameters are from the D1B2RUEX structure provided in the *MACLIB* library.

Viewing the Results

The following table lists the valid Return Codes for this utility.

Code	Description
0	Successful
4	'WARNING' – an anomaly has been detected during processing.

The return code is displayed in the TRC field and in the LOG file as TRC=46RC, like all return codes from a user exit. If an anomaly occurs, a WTO message is sent to the OS/390 LOG.

WTO Messages

The table below describes the conventions for WTO messages sent by this utility.

Character	Description
????????	Status or flag
++++++++	Numeric field, a number, a count, a code
////////	Alphanumeric, partner name for example

The following table describes the possible messages for this utility.

L1GFICN1 Message		
TOM+CN1 NOTICE L1GFICN1 INIT		01/04/25 20:07:33
Initialization of Connect:Express		
TOM+CN1 ERROR //////// mod-type ++++++++ TRC++++ PRC++++ NR++++++ ?/++++++		
Information is: Partner, calling mode– error type, Request number, TRC, PRC, NRC1, link type/NRC2, Time Mod – INP, input call Mod – OUT, output call Error type – CLOS,FAIL during connection Error type – APPL,INIT,END,ABND,FAIL during transfer selection Link type: S=SNA, T=TCP/IP, X=X25, C=LU6.2		
TOM+CN1 NOTICE L1GFICN1 TERM		01/04/25 24:37:05
Connect:Express termination phase		

Error Types

Error types correspond to the call type. The following table lists the connection errors.

Error	Description
CLOS	Session has been interrupted by a protocol message ABORT.
FAIL	Session has been rejected.

The table below describes the possible selection errors.

Error	Description
APPL	Call before the beginning of transfer, before reception demand by caller.
INIT	Call before the beginning of transfer – allocation phase.
END	Call at end of transfer – deallocation phase.
FAIL	Call at end of transfer when transfer is interrupted.
ABND	Call at end of transfer when an abend occurred (Example D37, 913)

Checking General Operations (P1GFISM1)

This utility checks file transfer operations and periodically scans Connect:Express resources such as PCT, FCT, RCT, HCT, XCT, APM, and Network resources. The utility sends an alert message each time an anomaly is detected, and performs more than 31 types of controls. This automatic process performs the operations just as a human operator would using the TSO/ISPF interface. This is a permanent program that runs until a fatal error occurs or until it receives an EXIT or STOP command.

The P1GFISM1 utility runs automatically and uses the Connect:Express API with dynamic calls to the DISPLAY services of the L0B2Z20 interface module. All calls are performed systematically, the answers are analyzed, and you are notified of any anomalies with WTO messages. This program is executed in its own address space.

This program can be running even if Connect:Express is not up. An alert message is sent until the monitor is started.

Implementing the P1GFISM1 Monitor

You can only start the P1GFISM1 program if the subsystem that you want to check has been declared to OS/390. If the TOMP subsystem is unknown, the program stops.

```
   TOMPSM1 ERROR   L0B2Z20   RC00000004              STATE T2908 TOM/Z20
```

The version of Connect:Express is also controlled, and if versions are not compatible, the program stops. The following screen shows an example:

```
   TOMPSM1 ERROR   L0B2Z20   RC00000004              STATE T2916 TOM/Z20
```

The program works even if Connect:Express has been stopped and has not been restarted since the last IPL, or if Connect:Express is terminating. The screen below shows an example:

```
   TOMPSM1 ERROR    L0B2Z20   RC00000004                  STATE T2001 TOM/Z20
```

This message displays when Connect:Express is not running and the program is waiting for next start.

The P1GFISM1 program is authorized (APF), and WTO messages do not appear on the OS/390 console.

Initialization Parameters

You can start the program using the following EXEC parameters:

```
   SSN= 'subsystem name', ITS= Number of seconds, MFY
```

The table below describes the parameters.

Parameter	Description
SSN	Subsystem name in the form TOM?
ITS 6 characters	The program uses a timer for periodic scanning of the resources. This timer is given in seconds. Example: ITS=001800 means 30 minutes. (ITS=000000 is equivalent to asking a check up performed only during termination). Positions of parameters are fixed. The number of seconds must be entered as 6 characters.
MFY (optional)	MODIFY commands can be passed. The default, and standard option, is a REPLY to a WTOR message issued by the program P1GFISM1. If MFY option is chosen, the program expects a 'STOP' command to terminate.

Operational Commands

You can issue commands while the program is running. These commands are sent through MODIFY or REPLY (standard option). The table below lists the valid commands.

Command	Description
STOP	Termination requested (Option 'MFY' active)
EXIT	Termination requested
STAT	This command requests a file transfer using the symbolic file 'STATS'. This file must be defined in the Files Directory of Connect:Express. (See *L1GFIJN1 description*)
LOAD	Load L1USRSM1 user exit
SCAN	This command requests an immediate check up of Connect:Express resources.

A message is displayed each time you enter a command, as shown in the following example.

```
   TOM2SM1 NOTICE   REPLY    STAT
   TOM2SM1 NOTICE   REPLY    EXIT
```

Using NETVIEW CLIST

WTO messages with the parameter ROUTCDE=11 are considered informational and processed by Netview, so you must define a Netview Subsystem. A NETVIEW CLIST can initiate an action depending on the message received. Refer to IBM documentation for more information about Netview.

Implementing the L1USRSM1 User Exit Module

You can implement a user exit module called L1USRSM1 that enables the user to send messages and take an action based on the message received. This exit is called when a WTO message is sent and before it goes to the OS/390 console. An assembly example is in the *SAMPOPT* library.

This exit is dynamically loaded and you can request it with the LOAD command. Input parameters are from the D1GFISM1 structure provided in the *MACLIB* library. The exit is called during initialization and termination of the P1GFISM1 program. When you execute the 'LOAD' command, the current exit is called for termination and the new exit is then loaded and called for initialization.

Viewing the Results

The table below lists the possible return codes for this utility.

Code	Description
0	Successfully processed
4	'WARNING' – one anomaly was detected during process
8	ERROR detected during process
12	FATAL ERROR detected during process

WTO Messages

Messages provide information about actions performed, statistics about objects that have been controlled, and inform you of anomalies found during transfer operations. The WTO messages are time stamped and can be redirected to a file as shown in the screen below.

```
//WTOPRINT DD SYSOUT=*
```

If an abnormal situation occurs, information from the L0B2Z20 interface can be written in a file, and then you can add a SNAP file, as shown in the example below.

```
//SYSPR20 DD SYSOUT=*
//SNAP    DD SYSOUT=*
```

The anomaly message shows the subsystem name, the type of message, the resource involved, and the anomaly detected. The format is 'Ssn'SM1 'Type' 'Resource' 'Message.' The following table describes the variables of the anomaly message.

Variable	Length	Description
Ssn	4 characters	Subsystem name in the form of TOM?

Variable	Length	Description
Type	8 characters	Type of message: NOTICE Provides information about a step while processing. ERROR Indicates that an anomaly has been detected for a global resource (PCT,FCT,RCT, Network.) WARNING Indicates a non-critical anomaly. SEVERE Indicates a situation that is considered to be critical. SCANNED Is followed by statistical numbers.
Resource	8 characters	Connect:Express Resource involved in the following information. Examples: MCHXX, REQUEST, P1GFISM1
Message	92 characters	The message itself

The table below describes the conventions used in WTO messages for this utility.

Character	Description
????????	Status or flag
++++++++	Numeric field, a number, a count, a code
////////	Alphanumeric , partner name for example
P++++	Return code PRC
T++++	Return code TRC
T++++/	Return code TRC and a letter / (L for 'Local', R for 'Remote')

Some messages are critical for Connect:Express processing, like 'ERROR' or 'SEVERE.' These messages appear in bold characters in the following list of messages. Others messages are critical for the utility's processing and need to be included in the alerts processed. One example of this type of message is *outlined* in bold in the list below.

P1GFISM1 Messages

```
TOM+SM1 NOTICE ////////(JOBNAME) READY
    The automate is ready to receive commands from the operator.
TOM+SM1 ERROR   ADDR     ////////   (INTERNAL ERROR->STOP)
    The module name //////// might be L0B2Z20/L1USRSM1
TOM+SM1 ERROR   APM++    ABENDED
    APM++ = APM01 TO 08
TOM+SM1 ERROR   COMMAND  ////       (INVALID COMMAND)
TOM+SM1 ERROR   DISPLAY  APM        (INTERNAL ERROR->STOP)
TOM+SM1 ERROR   DISPLAY  FCT        (INTERNAL ERROR->STOP)
TOM+SM1 ERROR   DISPLAY  GBL        (INTERNAL ERROR->STOP)
TOM+SM1 ERROR   DISPLAY  NET        (INTERNAL ERROR->STOP)
TOM+SM1 ERROR   DISPLAY  PCT        (INTERNAL ERROR->STOP)
TOM+SM1 ERROR   DISPLAY  RCT        (INTERNAL ERROR->STOP)
TOM+SM1 ERROR   DISPLAY  TCT        (INTERNAL ERROR->STOP)
TOM+SM1 ERROR   EMPTY    FCT        (INTERNAL ERROR->STOP)
TOM+SM1 ERROR   EMPTY    PCT        (INTERNAL ERROR->STOP)
TOM+SM1 ERROR   EMPTY    RCT        (INTERNAL ERROR->STOP)
TOM+SM1 ERROR   ENQ                 (USER ERROR->STOP, ALLREADY ACTIVE)
TOM+SM1 ERROR   ESTAE               (INTERNAL ERROR->STOP)
TOM+SM1 ERROR   EXEC L0B2Z20 RC++++++++ STATE T++++ TOM/Z20
    P1GFISM1 stops if the Subsystem is unknown or versions are not compatible.
```

P1GFISM1 Messages (continued)

TOM+SM1 ERROR	**FCT**	**????????**	
TOM+SM1 ERROR	FILE	DISABLED	//////// ++++++++ REQUEST(S)-
TOM+SM1 ERROR	FILE	UNK. STA:? ////////	
TOM+SM1 ERROR	LOAD	//////// (ERROR PRELOAD->STOP,CHECK STEPLIB)	
The module name //////// might be L0B2Z20/L1USRSM1			
TOM+SM1 ERROR	LOOP	FCT	(INTERNAL ERROR->STOP)
TOM+SM1 ERROR	LOOP	PCT	(INTERNAL ERROR->STOP)
TOM+SM1 ERROR	LOOP	RCT	(INTERNAL ERROR->STOP)
TOM+SM1 ERROR	LU 6.2	ABENDED	////////
TOM+SM1 ERROR	MCH++	DISABLED	////////
MCH++ = MCH01 to 16			
TOM+SM1 ERROR	OPEN	//SNAP	(USER ERROR, CHECK JCL)
TOM+SM1 ERROR	OPEN	//WTOPRINT	(USER ERROR, CHECK JCL)
TOM+SM1 ERROR	PARM	?	(USER ERROR->STOP, CHECK PARM)
		? = K- keyword missing or misplaced	
		? = L- PARM field missing or invalid length	
		? = N- invalid numeric field	
		? = M- 'MFY' option expected	
		? = P- parameter missing or misplaced	
TOM+SM1 ERROR	PARTNER	DISABLED	//////// ++++++++ REQUEST(S)-
TOM+SM1 ERROR	PARTNER	UNK. STA:? ////////	
TOM+SM1 ERROR	PCNEPAD	ABENDED	////////
TOM+SM1 ERROR	**PCT**	**????????**	
TOM+SM1 ERROR	RCT	????????	
TOM+SM1 ERROR	REQ-SIT	DISABLED ++++++++ FILE "////////" PARTNER "////////"	
SIT Transfers only			
TOM+SM1 ERROR	REQUEST	UNKNOWN .STA. ++++++++	
TOM+SM1 ERROR	REQUEST	UNKNOWN .ST1.	
TOM+SM1 ERROR	REQUEST	UNKNOWN .ST2.	
TOM+SM1 ERROR	REQUEST	UNUSABLE ++++++++ FILE "////////" PARTNER "////////"	
TOM+SM1 ERROR	REQUEST	T++++P++++ ++++++++ FILE "////////" PARTNER "////////"	
TOM+SM1 ERROR	REQ-SIT	T2077 RET++ ++++++++ FILE "////////" PARTNER "////////"	
RET++ is the number of connection retries,for SIT sessions only.			
TOM+SM1 ERROR	SNAPA	ABENDED	////////
TOM+SM1 ERROR	SNA3270	ABENDED	////////
TOM+SM1 ERROR	**SSN**	**????????**	**(SUBSYS NOT UP BECAUSE 'STOP')**
TOM+SM1 ERROR	TCP-IP	ABENDED	////////
TOM+SM1 ERROR	TIMER	L1B2PTIM	(INTERNAL ERROR->STOP)
TOM+SM1 ERROR	**USRJNL**	**????????**	**(UEXJNL NOT UP BECAUSE 'ABEND')**
SIT transfers.			
TOM+SM1 ERROR	X25DATE	ABENDED	////////
TOM+SM1 ERROR	X25GATE	ABENDED	////////
TOM+SM1 NOTICE	L1USRSM1	INIT (NOT BY WTO BUT BY INTERNAL PARMLIST)	
TOM+SM1 NOTICE	L1USRSM1	TERM (NOT BY WTO BUT BY INTERNAL PARMLIST)	
TOM+SM1 NOTICE	MODIFY	////	
TOM+SM1 NOTICE	P1GFISM1	STARTED V?R? &COMPILE-DATE	
TOM+SM1 NOTICE	P1GFISM1	ENDED	
TOM+SM1 NOTICE	P1GFISM1	ABEND RECOVERY (INTERNAL/USER ERROR->STOP)	
The L1USRSM1 user exit is not informed of the Abend.			
TOM+SM1 NOTICE	REPLY	////	
TOM+SM1 NOTICE	TRF	STATS ??? STATE T++++	
		??? = OK - successful'STATS' transfer	
		??? = KO - error during 'STATS' transfer TRC= ++++	
		??? = REC - 'STATS' transfer request has been registered	
		(C:X is not running) TRC=1000	
TOM+SM1 NOTICE	+++++++	SECONDS TIMER	
TOM+SM1 SCANNED	++++++++ / ++++++++ EFF		
TOM+SM1 SCANNED	++++++++ / ++++++++ MCH		
TOM+SM1 SCANNED	++++++++ FILE(S)		
TOM+SM1 SCANNED	++++++++ PARTNER(S)		
TOM+SM1 SCANNED	++++++++ REQUEST(S)		
TOM+SM1 SEVERE	**APM**	**NONE ENABLE**	

P1GFISM1 Messages (continued)

```
TOM+SM1 SEVERE   EFF       NONE ENABLE
TOM+SM1 SEVERE   MCH       NONE ENABLE
TOM+SM1 SEVERE   P1GFISM1 ERROR  ? ?   (INTERNAL ERROR->STOP)
TOM+SM1 WARNING  EFF       ALL USED
TOM+SM1 WARNING  FILE      DISABLED      ////////
TOM+SM1 WARNING  PARTNER   ++++/++++ LNK ////////
    Only if the number of allocated links is greater than 1 and if the maximum has been
    reached.
TOM+SM1 WARNING  PARTNER   DISABLED      ////////
TOM+SM1 WARNING  REQUEST   COUNT         +++++++
    Only if the count is greater than 700 requests
TOM+SM1 WARNING  REQUEST   T2077 RET++ ++++++++ FILE "////////" PARTNER "////////"
    NOTE: RET++ IS RETRY COUNT OF CONNECTION (NOT SIT).
TOM+SM1 WARNING  REQUEST   DISABLED      ++++++++
TOM+SM1 WARNING  REQUEST   NO APM CLASS  ++++++++
    The transfer class of the request is not served by any APM.
```

Managing a Remote Monitor (L1GFIRC1)

This program is a beginning and end of transfer exit module, and is executed in the APM address space. This utility enables you to monitor operations remotely. For example, you can:

❖ Activate a Connect:Express resource

❖ Check a network link or the status of one Connect:Express resource

❖ Make or purge a file transfer request

An operation is performed by sending a file with a special symbolic file name that activates the L1GFIRC1 exit at the beginning and end of transfer. The exit performs the action based on the symbolic file name. The following table describes the three types of operations that you can perform.

Type of Operation	Symbolic File Name	Action
CMD/RES	RC1CMD RC1RES	Send a command file to a remote Connect:Express monitor. The remote monitor will send back a report file.
ECHO	RC1ECHO RC1ECHOS RC1ECHOX RC1ECHOI	Send a command file to a remote Connect:Express monitor. The remote monitor will send back the file. SNA link is forced. X25 link is forced. TCP/IP link is forced.
LOOP	RC1LOOP RC1LOOPS RC1LOOPX RC1LOOPI	Send a file back and forth between two Connect:Express monitors. To stop the loop, you must disable the partner or the file and purge the transfer requests currently interrupted or waiting. SNA link is forced. X25 link is forced. TCP/IP link is forced.

Each type of operation results in one file transfer in one direction, and one file transfer back. The LOOP operation results in a continued loop. In all cases, the network link type is the same in both directions.

The L1GFIRC1 program is authorized (APF), and WTO messages with the parameter ROUTCDE=11 are not sent to the OS/390 console.

Implementing the L1GFIRC1 Exit

This program is a standard exit that you can activate at the beginning and/or end of transfer. It must be defined in the exit fields of the symbolic file definition involved. Input Parameters are received from the D1B2RUEX structure provided in the *MACLIB* library.

After the symbolic files, RC1CMD, RC1RES, RC1ECHO* and RC1LOOP*, are created in the Files directories of all Connect:Express monitors involved, you can start the process at any time by making a transfer request with the appropriate symbolic file name.

Messages are generated in a SYSPRINT file allocated to the APM address space where the process is running with the DDNAME built. The following screen shows an example.

```
                        SYSPRINT File: RC'mmss'ef'
```

In this example, RC is a constant prefix, 'mmss' represents minutes and seconds, and 'ef' is the number of the effectors where the transfer is executed.

You can access this file under SDSF.DA, with the '?' command. It contains messages about the 'CMD,' 'RES,' or 'ECHO' processes. If an error occurs when accessing this SYSPRINT file, WTO messages are sent.

Viewing the Results

The following table lists the return codes sent by this utility.

Code	Description
0	Successful
4	'WARNING' – invalid command syntax
8	ERROR detected in the parameter list
12	FATAL ERROR

The return code is displayed in the TRC field of the RCT entry in the Journal record and in the Log file. TRC=40RC is used at the beginning of a transfer, and TRC=49RC is used at the end of a transfer. A more detailed message is written in the APM SYSPRINT file.

WTO Messages

The table below describes the conventions for WTO messages sent by this utility.

Character	Description
????????	Status or flag
++++++++	Numeric field, a number, a count, a code
////////	Alphanumeric , partner name for example

The following table lists the possible messages for this utility.

L1GFIRC1 Messages
TOM+RC1 NOTICE //////// TRANSFERED   ////////
This file was transferred to this partner
TOM+RC1 NOTICE //////// RECEIVED   ////////
This file was received from this partner
*TOM+RC1 ERROR   //////// /////////////////////////////
The transfer failed, see SYSMSG and SYSPRINT files

The CMD/RES Process

With a CMD/RES operation, a command file is transferred under the symbolic file name RC1CMD, and then the receiver processes this file and sends a report under the symbolic file name RC1RES. Commands are defined by the sender using a special syntax. These commands represent calls to the standard API interface L0B2Z20.

The remote Connect:Express monitor enters the commands under RACF control. The standard L1B2PTAC module is used with the SAF option. The result of this process is written on the corresponding command line, and then sent back to the originator. Results are analyzed and notifications are sent.

The CMD/RES process uses two files--the command file and the report file. These files are described in the sections that follow.

> **Note:** If 'USER' is not defined ('IRR012I') or access is refused (ABEND913), the command is not processed.

The following table lists the parameters that you can use with RACF controls:

Parameter†	Description
USER	Symbolic name of the caller.
RESOURCE	A file name is built as follows:
	'Subsys.Table.Id'
Subsys	Called Subsystem name
Table	Abbreviated name of the table: FTC/PCT/RCT/TCT/TRF
Id	Name of the entry in the table (blank for TRF table)

ACCESS	Depends on the L0B2Z20 service invoked by the command.
	    READ – Display
	    UPDATE – Enable/Disable/Interrupt/Restart
	    ALTER – Transfer

† These access rules must be declared to RACF on the remote site.

The Command File

You can edit the command file using the 'EDIT' command in the Transfer Request panel of Connect:Express. This enables you to enter a transfer request after creating or updating the command file. This file must have a

record length of 255 characters, with no blocking factor, as shown in the following example. The syntax of this file is based on the L0B2Z20 module use, with a special header line.

```
LRECL=255, BLOCKS=255, RECFM=F or LRECL=259, BLOCKS=263, RECFM=VB
```

Command File Header

The header is composed of two parts, one for the sender (command file) and one for the receiver (report file). The header uses 99 characters, and displays the date and time, USERID to be notified, system name, subsystem name TOM?, and one option from the following list:

Option	Description
FORMAT	Formatted results are notified to the TSO USER.
TRACE	The process is traced in the SYSPRINT file.
NOOPT	Results are provided in the form of the L0B2Z20 module answer.

System and subsystem fields must be the same for the sender and the receiver. The exit module updates the date and time fields, the system and subsystem name fields, and the option field which is changed to RES or CMD depending on the situation.

The screen below shows an example of a command file header, before and after processing.

```
$01/03/24 12:18:14 USR0004  SYSA  TOM1 FORMAT / AA/MM/JJ HH.MM.SS  UUUUUUU  SYSA TOM1
$01/03/24 12:18:14 USR0004  SYSA  TOM1 RES    / 01/13/24 12:18:16  APM0201  SYSA TOM1
```

Command Lines

A command is defined as follows.

```
                    Subsystem Service Table Id Parameters
```

Command	Description
Subsystem	Name of the local subsystem (TOM? Or any TOM*)
Service†	**D**isplay, **H**old, **T**ransfer, **E**nable, **R**estart, **I**nterrupt, Purge
Table†	**F**ct, **P**ct, **R**ct, **T**CT
Parameters	Partner for 'DP', 'HP', 'EP' File for 'DF', 'HF', 'EF' Request number for 'DR', 'DT','ER', 'HR', 'IR', 'PR' Transfer request parameters for 'TR'

† The 'service + table' combinations (DP, HP, EP …) are all combinations supported by the L0B2Z20 module. See D0B2ZEX* corresponding structures provided in the *MACLIB* library.

The Parameters of a transfer request use standard symbols and are defined as follows.

```
  TOM ?TR00000000 0000Link/Class/Priority/Type/Direction/File/Partner/ Data set name
```

The following screen shows an example of a transfer request.

```
   TOM2TR00000000  0000SA1NTF1      PART5    INDEX1.PS.V255.MEGA
```

In the example above, the Transfer Request was made to subsystem **TOM2,** on an **S**NA link, transfer class **A**, priority **1**, **N**ormal request, **T**ransmission of the file **F1** to the partner **PART5**, and data set **INDEX1.PS.V255.MEGA**. The zeros correspond to the request number and return code that are set in the RESCMD report by the receiver.

Example of a Command File

In the example below, the command file is sent from the subsystem TOM1, identified locally by the parameter DPCSID= SCIPSR, to the subsystem TOM2, identified by the partner symbolic name SCIPSR2.

```
   $01/04/25 10:10:10 USR0009  83BA TOM1 FORMAT / AA/MM/JJ HH.MM.SS UUUUUUUU 83BA TOM1
   TOM2HFF1
   TOM2DFF1
   TOM2EFF1
   TOM2HPSCIPSR
   TOM2DPSCIPSR
   TOM2EPSCIPSR
   TOM2HR00000001
   TOM2ER00000001
   TOM2DR00000001
   TOM2TR00000000  0000SA1NTF1     SCIPSR5 INDEX1.PS.V255.MEGA
   *NOTE : YOU CAN USE 'TOM*' FOR ANY REMOTE SUBSYS.
   *TOM*DPSCIPSR
   *---
   TOM3DPSCIPSR
   TOM2DPSCITOMP
   *---
   *XXXX........(EVOLUTION POSSIBLE TO OTHER SUBSYTEM DIALOGS)
```

The following operations are performed:

❖ Disable symbolic file F1

❖ Display symbolic file F1

❖ Enable symbolic file F1

❖ Disable symbolic partner SCIPSR

❖ Display symbolic partner SCIPSR

❖ Enable symbolic partner SCIPSR

❖ Disable request number 00000001

❖ Display request number 00000001

❖ Enable request number 00000001

❖ Transfer the file INDEX1.PS.V255.MEGA to partner SCIPSR5 under file name F1

❖ Display symbolic partner SCIPSR (*subsystem TOM3 is invalid*)

❖ Display symbolic partner SCITOMP (*access to this partner is protected*)

The Report File

The report file is built from the command file. In this file, the results of the command are added to the command lines and the header is updated. The example below is based on the command file from the previous section. Some lines are truncated.

```
$01/04/25 12:18:14 USR0004  83BA TOM2 RES   / 01/04/25 12:18:16 APM0201  83BA  TOM2
TOM2HFF1        0000
TOM2DFF1        0000   ...F1     $$ALL$$ $$ALL$$ *S2HINDEX1.&PARTNID.&FILENAM.
TOM2EFF1        0000
TOM2DFF1        0000   ...F1     $$ALL$$ $$ALL$$ *S2EINDEX1.&PARTNID.&FILENAM.
TOM2HPSCIPSR    0000
TOM2DPSCIPSR    0000   ...SCIPSR  -       THNAMSNX PSR2AP01PSR2AP049922707124
TOM2EPSCIPSR    0000
TOM2HR00000001  2015
TOM2ER00000001  2015
TOM2DR00000001  2900
TOM2TR00000090  0000SA1NTF1      SCIPSR5 INDEX1.PS.V255.MEGA
*---
TOM3DPSCIPSR    9999   ...                                        ER-SSN-CMD/=
TOM2DPSCITOMP   9999   ...                                        SAF-REJECT
```

In this example, some commands were refused and user USR0004 is notified through TSO because of the FORMAT option in the Command file. Results for Display services are written similar to answers from the L0B2Z20 module from column 126, if the return code is equal to 0000. The following list summarizes the results of the command file.

❖ Request 00000001 was not found (TRC=2015 and 2900 following the command).

❖ Transfer request was accepted, the number is 00000090 and the return code is null.

❖ Subsystem TOM3 was not found (TRC=9999).

❖ Access to SCITOMP partner was protected.

Definition of the Symbolic File RC1CMD

```
TOM4120---------------- FILES ATTRIBUTES (2/5) -----------------------------
 OPTION ===>

 SYMBOLIC NAME          : RC1CMD    MODE: NORMAL

 INIT STATE ............ : E         E: IN-SERVICE H: HOLD

 DIRECTION ............. : *         T:TRANSMIT R:RECEIVE *:TRANS./REC.
 RECEIVING PARTNER ..... : *         'NAME', £LIST, */$$ALL$$ OR $$API$$
 SENDING PARTNER ....... : *         'NAME', £LIST, */$$ALL$$ OR $$API$$

 PRIORITY .............. : 1         0:URGENT 1:FAST 2:NORMAL 3:SLOW
 DSN DEFINITION TYPE ... : D         D:DYNAMIC F:FIXED
 ALLOCATION RULE ....... : 2         1:PREALL. 2:TO CREATE 3:EXIT A:AP

 FILE TYPE ............. : S         S/H/M/P/PU/V/VU/UU/SU
 PRESENTATION PROTOCOL . : 04        COMPRESS.,DATA TYPE (01-16)
 UNLOAD/RELOAD MEMBER .. : -         OPTIONAL
 SECURITY TABLE ........ : -         OPTIONAL

 OPTION  : VIEW                      UPDATE: 94/02/03 17:47 PSR0009
 -ENTER- : NEXT SCREEN              -PF3- : CANCEL
```

```
TOM4120---------------- FILES ATTRIBUTES (3/5) -------------------------------
 OPTION ===>

 SYMBOLIC NAME             : RC1CMD    DEF.: D  ALL.: 2  TYPE: S  DIR.: *

 FILE NAME (DSNAME) ....   : INDEX1.&PARTNID.&FILENAM.&REQDAT.&REQTIM
 GDG NUMBER ............   : -          +XX OR -XX

 1 SYMBOLIC UNIT NAME ..   : 3390       'UNITNAME'
   VOLUME NAMES ........   : -        -        -        -        -
 2 SMS DATA-STOR-MGMT ..   : -                 -                 -
 DISPOSITION ...........   : NEW        SHR/OLD/NEW
 ALLOCATION TYPE .......   : -          CYL/TRK
 SPACE PRIM.-SEC. ......   : -     -    1 TO 4 NUMERIC CHARACTERS
 DIRECTORY BLOCKS .....    : -          IF PARTITIONED
 RECORD FORMAT .........   : -          F,FB,FBA,FBM,V,VB,VBA,VBM,VBS,VS,U
 LRECL-BLKSIZE .........   : -     -    1 TO 5 NUMERIC CHARACTERS
 RETENTION (EXPDT/RETPD)   : -          X'CCYYDDD',E'YYDDD'/R'NNNN'

 REMOTE DSN  (FTP) .....   :
 TYPE/STRUCTURE/MODE ...   :            EN/AN/I,F/R,B/C/S
 UNIQUE ................   : N          Y/N
 NOTE   :
```

```
TOM4120---------------- FILES ATTRIBUTES (4/5) -------------------------------
 OPTION ===>

 SYMBOLIC NAME             : RC1CMD    DEF.: D   ALL.: 2   TYPE: S   DIR.: *
                            UPRFCT=
 --- S : SYMBOLS CHECKING
 V
   TRANSMISSION :
   START EXIT .........   : L1GFIRC1   USER EXIT NAME
 _ START COMMAND .......   : -
   END EXIT ...........   : -          USER EXIT NAME
 _ END COMMAND .........   : -

   RECEPTION :
   START EXIT .........   : -          USER EXIT NAME
 _ START COMMAND .......   : -
   END EXIT ...........   : L1GFIRC1   USER EXIT NAME
 _ END COMMAND .........   : -


 OPTION  : VIEW                        UPDATE: 94/02/03 17:47 PSR0009
 -ENTER- : NEXT SCREEN                 -PF3- : PREVIOUS SCREEN
```

Definition of the Symbolic File RC1RES

```
TOM4120---------------- FILES ATTRIBUTES (2/5) -------------------------------
 OPTION ===>

 SYMBOLIC NAME             : RC1RES    MODE: NORMAL

 INIT STATE ............   : E         E: IN-SERVICE H: HOLD

 DIRECTION .............   : *         T:TRANSMIT R:RECEIVE *:TRANS./REC.
 RECEIVING PARTNER .....   : *         'NAME', £LIST, */$$ALL$$ OR $$API$$
 SENDING PARTNER .......   : *         'NAME', £LIST, */$$ALL$$ OR $$API$$

 PRIORITY ..............   : 1         0:URGENT 1:FAST 2:NORMAL 3:SLOW
 DSN DEFINITION TYPE ...   : D         D:DYNAMIC F:FIXED
 ALLOCATION RULE .......   : 2         1:PREALL. 2:TO CREATE 3:EXIT A:AP

 FILE TYPE .............   : S         S/H/M/P/PU/V/VU/UU/SU
 PRESENTATION PROTOCOL .   : 04        COMPRESS.,DATA TYPE (01-16)
 UNLOAD/RELOAD MEMBER ..   : -         OPTIONAL
 SECURITY TABLE ........   : -         OPTIONAL

 OPTION  : VIEW                        UPDATE: 94/02/03 17:47 PSR0009
 -ENTER- : NEXT SCREEN                 -PF3- : CANCEL
```

```
   TOM4120---------------- FILES ATTRIBUTES (3/5) -------------------------------
    OPTION ===>

    SYMBOLIC NAME            : RC1RES    DEF.: D   ALL.: 2   TYPE: S   DIR.: *

    FILE NAME (DSNAME) ....  : INDEX1.&PARTNID.&FILENAM.&REQDAT.&REQTIM
    GDG NUMBER ...........   : -           +XX OR -XX

    1 SYMBOLIC UNIT NAME ..  : 3390        'UNITNAME'
       VOLUME NAMES ........ : -         -          -          -          -
    2 SMS DATA-STOR-MGMT ..  : -                    -                     -
    DISPOSITION ...........  : NEW         SHR/OLD/NEW
    ALLOCATION TYPE .......  : -           CYL/TRK
    SPACE PRIM.-SEC. ......  : -     -     1 TO 4 NUMERIC CHARACTERS
    DIRECTORY BLOCKS .....   : -           IF PARTITIONED
    RECORD FORMAT .........  : -           F,FB,FBA,FBM,V,VB,VBA,VBM,VBS,VS,U
    LRECL-BLKSIZE .........  : -     -     1 TO 5 NUMERIC CHARACTERS
    RETENTION (EXPDT/RETPD)  : -           X'CCYYDDD',E'YYDDD'/R'NNNN'

    REMOTE DSN  (FTP) .....  :
    TYPE/STRUCTURE/MODE ...   :            EN/AN/I,F/R,B/C/S
    UNIQUE ................  : N           Y/N
    NOTE   :
```

```
   TOM4120---------------- FILES ATTRIBUTES (4/5) -------------------------------
    OPTION ===>

    SYMBOLIC NAME            : RC1RES    DEF.: D   ALL.: 2   TYPE: S   DIR.: *
                              UPRFCT=
    --- S : SYMBOLS CHECKING
    V
      TRANSMISSION :
      START EXIT .........   : L1GFIRC1   USER EXIT NAME
    _ START COMMAND .......  : -
      END EXIT ...........   : -          USER EXIT NAME
    _ END COMMAND .........  : -

      RECEPTION :
      START EXIT .........   : -          USER EXIT NAME
    _ START COMMAND .......  : -
      END EXIT ...........   : L1GFIRC1   USER EXIT NAME
    _ END COMMAND .........  : -


    OPTION  : VIEW                        UPDATE: 94/02/03 17:47 PSR0009
    -ENTER- : NEXT SCREEN                 -PF3- : PREVIOUS SCREEN
```

The ECHO Process

This process works like the CMD/RES process, but the option field must be set to ECHO, as shown below.

```
  $01/04/25 12:18:14 USR0004  83BA TOM2 ECHO     / 01/04/25 12:18:16 APM0201  83BA
```

The command file must have a record length of 255 characters, with no blocking factor.

```
  LRECL=255, BLOCKS=255, RECFM=F or LRECL=259, BLOCKS=263, RECFM=VB
```

Using the average sequential file results in a loop that can only be stopped by disabling the Partner or the file.

Definition of the Symbolic Files RC1ECHO*

```
TOM4120---------------- FILES ATTRIBUTES (2/5) ------------------------------
 OPTION ===>

 SYMBOLIC NAME            : RC1ECHO    MODE: NORMAL

 INIT STATE ...........   : E          E: IN-SERVICE H: HOLD

 DIRECTION .............   : *          T:TRANSMIT R:RECEIVE *:TRANS./REC.
 RECEIVING PARTNER .....   : *          'NAME', £LIST, */$$ALL$$ OR $$API$$
 SENDING PARTNER .......   : *          'NAME', £LIST, */$$ALL$$ OR $$API$$

 PRIORITY ..............   : 1          0:URGENT 1:FAST 2:NORMAL 3:SLOW
 DSN DEFINITION TYPE ...   : D          D:DYNAMIC F:FIXED
 ALLOCATION RULE .......   : 2          1:PREALL. 2:TO CREATE 3:EXIT A:AP

 FILE TYPE .............   : S          S/H/M/P/PU/V/VU/UU/SU
 PRESENTATION PROTOCOL .   : 04         COMPRESS.,DATA TYPE (01-16)
 UNLOAD/RELOAD MEMBER ..   : -          OPTIONAL
 SECURITY TABLE ........   : -          OPTIONAL

 OPTION  : VIEW                         UPDATE: 94/02/03 17:47 PSR0009
 -ENTER- : NEXT SCREEN                  -PF3- : CANCEL
```

```
TOM4120---------------- FILES ATTRIBUTES (3/5) ------------------------------
 OPTION ===>

 SYMBOLIC NAME            : RC1ECHO   DEF.: D   ALL.: 2   TYPE: S   DIR.: *

 FILE NAME (DSNAME) ....   : INDEX1.&PARTNID.&FILENAM.&REQDAT.&REQTIM
 GDG NUMBER ...........    : -              +XX OR -XX

 1 SYMBOLIC UNIT NAME ..   : 3390        'UNITNAME'
   VOLUME NAMES ........   : -         -          -          -          -
 2 SMS DATA-STOR-MGMT ..   : -                    -          -
 DISPOSITION ...........   : NEW         SHR/OLD/NEW
 ALLOCATION TYPE .......   : -           CYL/TRK
 SPACE PRIM.-SEC. ......   : -     -     1 TO 4 NUMERIC CHARACTERS
 DIRECTORY BLOCKS .....    : -           IF PARTITIONED
 RECORD FORMAT .........   : -           F,FB,FBA,FBM,V,VB,VBA,VBM,VBS,VS,U
 LRECL-BLKSIZE .........   : -     -     1 TO 5 NUMERIC CHARACTERS
 RETENTION (EXPDT/RETPD)   : -           X'CCYYDDD',E'YYDDD'/R'NNNN'

 REMOTE DSN  (FTP) .....   :
 TYPE/STRUCTURE/MODE ...   :             EN/AN/I,F/R,B/C/S
 UNIQUE ...............    : N           Y/N
 NOTE    :
```

```
TOM4120---------------- FILES ATTRIBUTES (4/5) ------------------------------
 OPTION ===>

 SYMBOLIC NAME            : RC1ECHO   DEF.: D   ALL.: 2   TYPE: S   DIR.: *
                           UPRFCT=
 --- S : SYMBOLS CHECKING
 V
   TRANSMISSION :
   START EXIT .........   : L1GFIRC1   USER EXIT NAME
 _ START COMMAND .......   : -
   END EXIT ...........    : -         USER EXIT NAME
 _ END COMMAND .........   : -

   RECEPTION :
   START EXIT .........   : -         USER EXIT NAME
 _ START COMMAND .......   : -
   END EXIT ...........    : L1GFIRC1  USER EXIT NAME
 _ END COMMAND .........   : -


 OPTION  : VIEW                         UPDATE: 94/02/03 17:47 PSR0009
 -ENTER- : NEXT SCREEN                  -PF3- : PREVIOUS SCREEN
```

The LOOP Process

You can use the LOOP process to load a network link and test it, or to load the Connect:Express monitor. With the loop process, you send back the file you just received. This results in one file created for each transfer. You can delete the transmitted file with an end of transfer command so that it does not waste space on the disk. The LOOP process can be set up using the L1GFIUE1 general exit.

Note: The original file that you transmit is deleted automatically.

The following screen shows an example of an end of transfer command for deleting the file transferred. Refer to the Files Attributes 4/5 screen shown on page 2-24.

```
//PROCDEL    PROC D=
//BR14       EXEC PGM=IEFBR14
//DD1        DD DSN=&D,DISP=(OLD,DELETE)
```

Definition of the Symbolic Files RC1LOOP*

```
  TOM4120---------------- FILES ATTRIBUTES (2/5) ------------------------------
   OPTION ===>

   SYMBOLIC NAME           : RC1LOOP    MODE: NORMAL

   INIT STATE ............  : E          E: IN-SERVICE H: HOLD

   DIRECTION .............  : *          T:TRANSMIT R:RECEIVE *:TRANS./REC.
   RECEIVING PARTNER .....  : *          'NAME', £LIST, */$$ALL$$ OR $$API$$
   SENDING PARTNER .......  : *          'NAME', £LIST, */$$ALL$$ OR $$API$$

   PRIORITY ..............  : 1          0:URGENT 1:FAST 2:NORMAL 3:SLOW
   DSN DEFINITION TYPE ...  : D          D:DYNAMIC F:FIXED
   ALLOCATION RULE .......  : 2          1:PREALL. 2:TO CREATE 3:EXIT A:AP

   FILE TYPE .............  : S          S/H/M/P/PU/V/VU/UU/SU
   PRESENTATION PROTOCOL .  : 04         COMPRESS.,DATA TYPE (01-16)
   UNLOAD/RELOAD MEMBER ..  : -          OPTIONAL
   SECURITY TABLE ........  : -          OPTIONAL

   OPTION  : VIEW                        UPDATE: 94/02/03 17:47 PSR0009
   -ENTER- : NEXT SCREEN                 -PF3- : CANCEL
```

```
  TOM4120---------------- FILES ATTRIBUTES (3/5) ------------------------------
   OPTION ===>

   SYMBOLIC NAME           : RC1LOOP   DEF.: D   ALL.: 2   TYPE: S   DIR.: *

   FILE NAME (DSNAME) ....  : INDEX1.&PARTNID.&FILENAM.&REQDAT.&REQTIM
   GDG NUMBER ............  : -             +XX OR -XX

   1 SYMBOLIC UNIT NAME ..  : 3390        'UNITNAME'
      VOLUME NAMES ........ : -         -         -         -         -
   2 SMS DATA-STOR-MGMT ..  : -                   -         -         -
   DISPOSITION ...........  : NEW         SHR/OLD/NEW
   ALLOCATION TYPE .......  : -           CYL/TRK
   SPACE PRIM.-SEC. ......  : -     -     1 TO 4 NUMERIC CHARACTERS
   DIRECTORY BLOCKS .....   : -           IF PARTITIONED
   RECORD FORMAT ........   : -           F,FB,FBA,FBM,V,VB,VBA,VBM,VBS,VS,U
   LRECL-BLKSIZE ........   : -     -     1 TO 5 NUMERIC CHARACTERS
   RETENTION (EXPDT/RETPD)  : -           X'CCYYDDD',E'YYDDD'/R'NNNN'

   REMOTE DSN  (FTP) .....  :
   TYPE/STRUCTURE/MODE ...  :             EN/AN/I,F/R,B/C/S
   UNIQUE ................  : N           Y/N
   NOTE   :
```

```
 TOM4120---------------- FILES ATTRIBUTES (4/5) -------------------------------
  OPTION ===>

 SYMBOLIC NAME             : RC1LOOP   DEF.: D   ALL.: 2   TYPE: S   DIR.: *
                             UPRFCT=
 --- S : SYMBOLS CHECKING
 V
   TRANSMISSION :
   START EXIT .........   : L1GFIRC1   USER EXIT NAME
 _ START COMMAND .......   : -
   END EXIT ............   : -          USER EXIT NAME
 _ END COMMAND .........   : S PROCDEL, D=&DSN

   RECEPTION :
   START EXIT .........   : -          USER EXIT NAME
 _ START COMMAND .......   : -
   END EXIT ............   : L1GFIRC1   USER EXIT NAME
 _ END COMMAND .........   : -

 OPTION  : VIEW                        UPDATE: 94/02/03 17:47 PSR0009
 -ENTER- : NEXT SCREEN                 -PF3- : PREVIOUS SCREEN
```

Chapter 3

Integrating Transfer Operations into the User Environment

This chapter describes the utilities that enable you to activate user procedures, delete checkpoint files, purge transfer requests, and manage Partner and File directories.

Overview

To integrate file transfer processes into the user environment, you must setup procedures for managing the operational environment and procedures for managing data transfers. For example, you may want to automatically delete obsolete files, setup start procedures, or forward received files. The utilities listed in the table below can help you accomplish this. These modules can run externally or online. Each one is described in its own section following the table.

Function	Module	Description
Activating User Procedures	L1GFIUE1	This exit can be used for conditionally activating one or more user procedures like Start commands, Jobs, Programs, REXX, or CLISTS.
Deleting Checkpoint files	P1GFICP1	External tool. Checkpoint files attached to a transfer request are normally automatically deleted when the request is executed or purged by an operator. In some cases, these files can stay on disk. For example, if the monitor is cold started, the checkpoint files are not deleted. This external tool enables you to perform a controlled and automatic purge of any remaining checkpoint files.
Purging Transfer Requests	P1GFIRP1	This external tool enables you to purge transfer requests by partner name, file name, age or return codes.
Using the IDCAMS Utility	P1GFIIDC & P1GFIID0	These two external tools send an EXEC PARM field to IDCAMS.
Managing Partner and File Directories with Batch Processes	P0B2DIR	This external tool manages Connect:Express Partners and Files directories with a batch process.

Activating User Procedures (L1GFIUE1)

This program can be implemented as a selection exit or a beginning and end of transfer exit module that is executed in the APM or EAS address space. This utility enables you to implement the user exit interface of Connect:Express without coding a program.

L1GFIUE1 exit parameters use a symbolic language to define conditions and actions to be taken. This language has a set of symbolic variables. Some are resolved dynamically from current file transfer information, while others are local or global variables used for internal resolution.

When this exit is activated, it processes a file called SYSUE1, in which all conditions and actions are defined. This source file is accessed automatically and does not need any compilation or link-edits. A TSO/ISPF operator interface is provided (Option 9.3.1) to help you define parameters in the file and simulate the process. Appendix A provides a list of action keywords and dynamic variables that you can use when defining the parameters for this utility. An Example of a SYSUE1 parameters file is in the ***PARMLIB*** library **(**£SYSUE1).

Executing the General Exit

The general exit is called in the APM address space (or the EAS address space for FTP transfers). You must add the SYSUE1 parameter file DD card in the JCL file, as well as all DD cards corresponding to the work files such as SYSPRINT and SYSPR20, that are needed for the called programs or procedures. It can be activated at four stages of a transfer operation. During the initialization and termination stage, the exit can intervene during the processing of the transfer request and then during the final realization phase. At the beginning and end of transfer, it can intervene in the allocation and deallocation phases.

When reading the parameter file *SYSUE1*, and if one of the conditions matches the current situation, L1GFIUE1 will execute the associated actions.

Using the General Exit During the Selection Phase

The general exit can be used during the initialization and termination phase of transfers. Therefore, it is defined as a selection exit declared in the T1APMSRT table (TSO/ISPF Option 3.3.SRT). Since the table T1APMSRT must be processed by the APM, you should activate the execution parameter SRV=L1APMSRV of the APM, as shown in the following example. In the case of FTP transfers, the driver L1APMSRV is always activated.

```
000030 //DEV$400 EXEC      PGM=P1APM001,REGION=0M,ACCT=(PSRACC),
000032 //   PARM=('APM=&APM','SSN=&SSN','MSG=&MSG','EFF=&EFF','ETC=&ETC',
000033 //        'SRV=L1APMSRV'),
000034 //         DYNAMNBR=64,DPRTY=(15,15),TIME=NOLIMIT
```

The phases are executed before the start of transfer (allocation phase) and after the end of transfer (deallocation phase). If the initialization was successful, the exit is called during the termination phase.

For example, in server mode, you can use the exit to ensure that a file is ready and available when the Partner requests the transfer. The exit is activated before transmitting the request to the monitor. Therefore, it is possible to determine the physical file name based on the request parameters, and to prepare a transmission request of type Hold. This request is presented at the moment when the request is processed by the monitor.

Using the termination phase enables you to process allocation errors. An allocation error occurs at the start of transfer. In this case, neither the start of transfer exit nor the end of transfer exit is activated. Only the call to the selection exit in the termination phase can process the error.

The selection exit is called according to the conditions associated with the protocol that you are using (for example, PeSIT E protocol number 5), the usage mode (for example session table 54 or 52), the direction of the session (incoming or outgoing), and the Partner type (Tom or Other).

Using the General Exit During the Transfer Phase

The general exit must be declared at the file definition level, like any other start or end of transfer exit, in transmission or reception. You can also define the exit in the monitor's SYSIN file with the keywords UEXFTS, UEXFTE, UEXFRS and UEXFRE, that correspond to the four types of calls to the exit. In this way, the symbolic file definition enables the deactivation of the exit using the keyword NONE placed in the appropriate field.

The exit is called at the start and end of transfer for all symbolic files for which it was declared. If the start of transfer was successful, the end of transfer exit is called. All errors detected before the start of transfer exit can only be processed by the selection exit during the termination phase.

Setting Up Parameters in the SYSUE1 File

A set of Keywords enables you to define conditions, actions, and variables that can be local, global, or dynamic. The IF keyword declares conditions. Other keywords describe actions and local or global variables. Refer to Appendix A of this book for a list of action keywords and dynamic variables that you can use in the SYSUE1 file.

The SYSUE1 file has the following format:

Keyword	Description
IF	Conditions. For example: For the symbolic file F, all partners, successful end of transfer
$Actionx$	Action to be taken and parameters of the action. For example: Enable symbolic file "FILE"
$Actiony$	Action to be taken and parameters of the action. For example: Delete the data set
IF	Conditions. For example: all files, all partners, transfer failure
$Actionz$	Action to be taken and parameters of the action. For example: Start MYPROC with parameters of MYPROC

For each call, the exit looks at the SYSUE1 file and stops any time that IF conditions match the current transfer situation. When an IF condition is met, Connect:Express executes the associated $actions$. If an $action$ fails, the exit skips any other $action$ declared behind the current IF, and moves on to the next IF keyword.

The L1GFIUE1 exit normally expects to find an IF condition in the SYSUE1 file that matches the current situation. If no match is found, the current transfer is interrupted with the return code TRC=4004 or TRC=4904, and PRC=3312, and the procedure that is declared after the ERR keyword is activated. The following messages are written in a SYSPRINT file whose name has the format 'Upmmsstt'. The exit creates this file

```
   TOM4 L1GFIUE1 NO 1 MATCH  FILENAM
   TOM4 L1GFIUE1 LAST REC.   *-------------------------------------
```

You can use the keywords NC_SEL, NC_IEX, NC_TEX so that errors are not returned just because no condition was met. These keywords should be placed at the beginning of the SYSUE1 file. NC_SEL indicates that no match found during initialization or termination steps is not an anomaly. NC_IEX indicates

that no match found at beginning of transfer is not an anomaly. NC_TEX indicates that no match found at the end of transfer is not an anomaly.

Note: It is better to declare one $action$ per IF condition, even if you duplicate the IF condition line. This way you are sure that no $action$ is skipped if any of them fail.

If one syntax error occurs during the process, the current transfer is interrupted with a return code of TRC=4008 or TRC=4908 and PRC=3312, and that is declared after the ERR keyword is activated.

After all variables in a command have been replaced, if the length of the command is greater than 120 characters, the current transfer is interrupted with a return code of TRC=4008 or TRC=4908 and PRC=3312. The procedure that is declared after the ERR keyword is activated and an error message is sent in a SYSOUT file named 'Upxxxxxx', with the line number of the error. This problem is more common with longer variables such as &LAB (80 characters), &DSN (44 characters), and &PI99 (254).

The following list outlines the rules and guidelines that apply to the SYSUE1 file.

❖   The maximum number of cards in the SYSUE1 file is 1024.

❖   Two successive IF keywords are not allowed.

❖   There is no limit to the number of $actions$ listed behind an IF keyword.

❖   File, Partner, TRC, PRC, SRC, Direction and Type fields of IF conditions support wild chars '*'.

❖   There is no limit to the number of variables that you can declare.

❖ $NOSWAP$, $NOSTAE$, VAR, $CAPS$, $NOCAPS$, $W-VOL$, $W-UNT$ and $W-RET$ cards can be placed anywhere in the SYSUE1 file.

❖ $ERR$$, $NC-IEX$, $NC-TEX$ and $NC-SEL$ keywords must be placed at the top of the SYSUE1 declarations.

❖   The $action$ phrase length cannot exceed 120 characters, after extension of the variables.

❖   As soon as an error is detected while performing an $action$, the next IF keyword is processed. All $action$ cards behind the failed $action$ are skipped.

❖   If an error occurs while processing an $action$, you will have TRC=40rc during the transfer initialization phase or TRC=49rc during the transfer termination phase.

❖   Except for the $NOSWAP$, $NOSTAE$, ERR, $NC-IEX$, $NC-TEX$, $NC-SEL$, $W-VOL$, $W-UNT$ and $W-RET$ cards, no $action$ card can be placed before the first IF keyword.

❖   Parameters of IF keywords are positional. Blanks must be left for further evolutions.

❖   No continuation character exists.

❖   '/*' characters in position one and two indicate the end of file, all cards behind are omitted.

❖   '*' character in the first position indicates a comment line.

Defining the Conditions

Conditions are defined using the IF keyword and can pertain to data flows or operation status. Note that several monitors can share the same SYSUE1 file. Conditions about data flow include:

❖   For one monitor or all monitors

❖   For one symbolic File, multiple symbolic Files, or all symbolic Files.

❖   For one symbolic Partner, multiple symbolic Partners, or all symbolic Partners.

❖   For transmission, reception, or both directions.

❖   For ongoing calls, outgoing calls, or both directions.

❖   For one type of network link, or all types.

Conditions for Connect:Express Operation status include the following:

❖ During initialization, at beginning of transfer, end of transfer, during termination, or all situations (*).

❖ If there is an error, a successful transfer, or both.

❖ If the transfer is retried or not.

❖ For one TRC return code value, a range of values, or all values.

❖ For one PRC return code value, a range of values, or all values.

❖ For one SRC return code value, a range of values, or all values.

The position and order of the parameters in the SYSUE1 file are fixed. The table below lists the position, the condition, and a brief description.

Position	Condition	Description
1	IF	Each IF defines the limit of a list of $actions$ to perform. If there is an error during an $action$, Connect:Express processes the next IF keyword. All IF conditions are processed during a single call of L1GFIUE1 exit.
6	File(s)	Symbolic file name or generic definition including WILD CHAR '*' or $$ALL$$ for all the symbolic files. Examples: FILE = only FILE, FIC***** = all files beginning with 'FIC'
15	Partner(s)	Symbolic partner name or generic definition including WILD CHAR '*' or $$ALL$$ for all the symbolic partners. Examples: PART = only PART, PAR***** = all partners beginning with 'PAR'
24	Monitor(s)	Fourth character of the subsystem name or '*' for all. Example: 1 = TOM1
26	Direction	T = Transmission, R = Reception, * = both directions
28	Type	I = Initialization, S = Start, E = End, T = Termination, * = in any case. For Initialisation and Termination calls, the L1GFIUE1 general exit must be declared in the T1APMSRT table. For Start and End calls , the L1GFIUE1 general exit must be declared in the Start exit and End exit fields, either in the file definition or in the SYSIN file.
30	Mode	D = Done (Successful), F = Failed, * = both cases
32	TRC	Return code or generic value including WILD CHAR '*': this condition is processed only if the mode is Failed. Examples: 5008 = file space full, 5*** = all TRC beginning with 5, **** = all TRC
37	PRC	Return code or generic value including WILD CHAR '*': this condition is processed only if the mode is Failed. Examples: 2211 = I/O error, 221* = all PRC beginning with 221, **** = all PRC
41	Retry	Y = must be performed for retried transfer.
43	File Type	Blank, Sequential, PDS …., or all. Examples: S = Sequential, H= HFS, P = PDS, V = VSAM, PU = PDS unload/reload, VU = VSAM unload, reload, UU = User unload/reload, SU = Sysout unload/reload
46	Type of access	Ongoing or outgoing call, blank, or all. Example: I = Input, O = Output, * = both direction
48	Type of link	Network link used, blank, or all. Examples: X = X25, S = SNA, I = TCP/IP, * = all

Position	Condition	Description
50	SRC	Return code or generic value including WILD CHAR '*': this condition is processed only if the mode is Failed. Examples: 0210 = File in use, 02** = all SRC beginning with 02, **** = all SRC

The following screen shows an example of a condition. Defining the Actions

```
1----6-------15-------24-–28--32---37-–41----46---50
$IF$ $$ALL$$ $$ALL$$  * * * F **** *** Y **  * X ****
For failed transfers on X25 links only, including retried transfers.
```

Each IF keyword is followed by a list of one or more actions. An action consists of a keyword followed by parameters. The action keyword is in column 1, and the parameters begin in column 10. The screen below shows an example of an action.

```
1-------10
$CMD$    SEND '&DIR,&TYP,&LNK,&PARTNID,&FILENAM',USER=(USER4)
Send to user USER4, by a TSO command, a message that indicates the direction, the file type, the network link, the
symbolic Partner name and the symbolic file name.
```

The following table lists the different types of actions that you can include in the parameters of the SYSUE1 file.

Action	Keyword
Simulation	SIM
Generation of error	$ABEND$ - PRC - $WAIT$
Environment	$NOSTAE$ - $NOSWAP$ - $CAPS$ - $NOCAPS$ - $W-VOL$ - $W-UNT$ - $W-RET$
Execution	ERR - $NC-IEX$ - $NC-TEX$ - $NC-SEL$ - CMD - DYA - $EXIT$ - JCL - $NOPI37$ - PGM - $PGMJ$ -$PI37$ - SUB - REQ - 300 - $PI99$ - $PI99WD$ - $PI99RD$ - $SNFY$ - $FNFY$- $PI11$ - $PI12$ - $PI61$ - $PI62$ - LOG
Local or global variables	VAR

You can implement some standard programs or Connect:Express utilities through the L1GFIUE1 general exit. They can be sent as parameters to one of the keywords such as $EXIT$, PGM, or $PGMJ$, or called by actions like JCL, REQ, 300, $SNFY$, $FNFY$, SUB, or LOG. The table below lists the keywords that you can use with standard programs and Connect:Express utilities.

Keyword	Program	Description
300	P1B2P300	Modifies (Enable, Disable, Purge.) of a file, a partner, a request.
$EXIT$	L1EX£AE2 L1GFICN1 L1EXSNAP	Traces PeSIT protocol fields. Sends notification of anomalies. Traces the exit input parameters with SNAPDUMP

Keyword	Program	Description
JCL	P1B2PJCL	Starts a user procedure with one parameter. See EX£PJCL provided in the *SAMPLIB* library..
LOG	L1B2LOG	Send a user message in the SYSLOG file of the APM.
PGM	Any program	Calls a user program with parameters passed through register1.
$PGMJ$	P1GFIIDC P1GFIID0 L1GFITS1 P1GFIRP1	Calls IDCAMS with EXEC PARM. Calls IDCAMS with EXEC PARM, without return code. Sends a TSO message to one user from a list. Purges a transfer request.
REQ	P1B2PREQ	Submits a transfer request with EXEC PARM.
SUB	Any utility	Submits a job from a JCL skeleton in which dynamic variables are replaced by current transfer parameters.
$SNFY$	P1B2PNFY	Sends Connect:Express notifications.
$FNFY$	P1B2PNFY	Sends Connect:Express notifications in a file.

Simulating the L1GFIUE1 Exit

To implement the L1GFIUE1 exit, your license must include the Utilities Package. You can however, simulate the process without an authorized license. With a full implementation of this utility, the SYSUE1 file is processed normally, and all simulation modes are supported. You can simulate one command or the entire process.

> **Note:** You can simulate L1GFIUE1 exits from the TSO/ISPF interface as well. This is available for test by non-authorized users and is described in the next section.

Simulations using the SIM Keyword

You can simulate an L1GFIUE1 exit using the SIM keyword. In this case, a SIM card is placed to simulate an OS/390 command and control the process without performing the action. The result of the process is written in a SYSPRINT file. The file name has the format 'Upxxxxxx'.

> **Note:** This mode is available for authorized users only.

Simulations using the Symbolic File L1GFIUE1

When you use the symbolic file L1GFIUE1 for a transfer, the SYSUE1 file is processed in a simulation mode. All the commands are processed, and the results are written in a SYSPRINT file. The file name has the format 'Upxxxxxx'. In this mode, an ERR action results in the WTO message 'L1GFIUE1 ERROR TOM? (SIMULATE)'.

> **Note:** This mode is available for test by non-authorized users.

Tracking Errors in the SYSPRINT File

When using a simulation mode or when an error occurs, the general exit writes messages in a SYSPRINT file that is created in the address space where the process was executing. You can access the SYSPRINT file through ISF (SDSF) with the option 'INPUT ON.' The file name has the following format.

```
UPmmsstt  - mm = minutes, ss = seconds, tt = thousands
```

On the ISF 'DA' screen, type a question mark '?' in front of the job and select the 'UP'… File corresponding to the transfer. All error messages are indicated on the right side of a SYSUE1 line number, and the message indicates the nature of the error. Error codes and messages are listed in Appendix A.

Creating the SYSUE1 File with the TSO/ISPF Interface

The TSO/ISPF operator interface can help you define the conditions and actions in the SYSUE1 file. From the Connect:Express Main Menu, select option **9.3.1** to display the SYSUE1 Tailoring screen. From this screen, you can:

❖ Define IF keywords and associated $actions$

❖ Define local and global variables

❖ Define the environment $actions$ not associated with an IF condition, for example (ERR, $NOSTAE$,$NOSWAP$)

❖ Process simulations

The figure below shows the options on the SYSUE1 Tailoring screen.

```
TOM4120          SYSUE1 TAILORING
 OPTION ===>

          1  C-CONDITIONS      CHAINING CONDITIONS

          2  V-VARIABLES       GLOBAL/LOCAL VARIABLES

          3  S-SYTEM           SYSTEM PARAMETERS

          4  T-SIMULATE        TEST EXIT


 FILE(MEMBER) ===> INDEX1.TOM4.SYSPRM(SYSUE1)'''''''''''''''' (SYSUE1)
```

Defining Conditions (Option 1)

The Conditions option enables you to define an IF line. Parameters are automatically placed in the correct position for you as you enter them, and values in the fields are controlled.

1. From the Connect:Express Main Menu, type 9.3.1 in the option field and press <Enter> to display the SYSUE1 Tailoring screen.

2. Type 1 in the option field and press <Enter> to define conditions.

```
TOM4120     SYSUE1 LIST PARAMETERS  $IF$                NAMES INITIALIZED    !
 OPTION ===>
           E : EDIT, I : INSERT $IF$, -ENTER- COMMANDS DEFINITION


 MONITOR ===> TOM4
 FILE(MEMBER) ===> INDEX1.TOM4.SYSPRM(SYSUE1)                  (SYSUE1)

  FILE ...... ===> TEST              (SYMBOLIC NAME/MASK/$$ALL$$)
  PARTNER ... ===> SCI*             (SYMBOLIC NAME/MASK/$$ALL$$)
  SUBSYS .... ===> *                (SUBSYS SUFFIX)
  DIRECTION . ===> R                (T/R/*)
  TYPE ...... ===> E                (S/E FILE, I/T SRT, *)
  MODE ...... ===> F                (D/F/*)
  SRC........ ===> 0000             (SRC/MASK TYPE=E/T MODE=F, ****)
  TRC ....... ===> 0000             (TRC/MASK TYPE=E/T MODE=F, ****)
  PRC ....... ===> 000              (PRC/MASK TYPE=E/T MODE=F, ****)
  RESTART ... ===> N                (Y/N)
  CODE ...... ===> **               (FILE CODE/**)
  ACCESS .... ===> *                (I/O/*)
  LINK ...... ===> *                (C/I/N/O/S/T/X/*)
           X EXIT, -PF3- END
```

The following table describes the commands that you can enter in the option field on the first screen.

Command	Description
E	Edit the SYSUE1 file directly
I	Creates a new condition line with an action line or adds an action line behind a condition line. If you specify option 'I' before going to next screen, you will create a condition line and an $action$ line. If you don't specify option 'I', you will only add an $action$ line behind an existing condition line.

3. Type E in the option field and press <Enter> to edit an existing line or type I to insert a new line.

4.   Enter your conditions and press <Enter> to define the actions associated with this condition. You enter one action at a time. Each $Action$ that you enter is kept in the user ISPF profile, so you can enter a new $action$ or select one from a list.

```
SYSUE1 PARAMETER   COMMAND
 OPTION ===>

 V---------------S: ADD, V: CHECK VARIABLES
 _  $SUB$       ===> JOB1''''''''''''''''''''''''''''''''''''''''''''''''''''
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
 _  $CMD$       ===> ''''''''''''''''''''''''''''''''''''''''''''''''''''''''
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
 _  $SIM$       ===> S PSRPROC''''''''''''''''''''''''''''''''''''''''''''''
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
 _  $PGM$       ===> P1B2PREQ   _ $PGMJ$ ===> ''''''''''   _ $EXIT$ ===> ''''''''''
 _  $REXX$      ===> ''''''''''''''''''''''''''''''''''''''''''''''''''''''''
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
 _  $REQ$       ===> ''''''''''''''''''''''''''''''''''''''''''''''''''''''''
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
 _  $JCL$       ===> ''''''''''''''''''''''''''''''''''''''''''''''''''''''''
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
 _  $300$       ===> ''''''''''''''''''''''''''''''''''''''''''''''''''''''''
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
 _  $DYA$       ===> ''''''''''''''''''''''''''''''''''''''''''''''''''''''''
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
 _  $PRC$       ===> '''          _  $WAIT$                 _  $ABEND$
           X EXIT, -PF3- BACK
```

Note: If you do not specify option 'I' and no existing IF is found, the $action$ is just inserted and this can result in an error during execution.

Defining Local or Global Variables (Option 2)

From the TSO/ISPF interface, you can enter one to four global variables and one to four local variables. Only one variable can be entered at a time, however. Variables are stored in the user ISPF profile so when you are defining variables, you can enter a new one or select an existing one from a list.

When you enter or select a variable, it is inserted into the SYSUE1 file and put in the correct place. Once the variable is in the SYSUE1 file, you can update the variable name in the file itself. The following screen shows the SYSUE1 List Variable screen where you can edit or select variables.

```
TOM4120     SYSUE1 LIST  VARIABLE $VAR$                    NAMES INITIALIZED    !
 OPTION ===>
             E : EDIT MEMBER

 MONITOR ===> TOM4
 FILE(MEMBER) ===> INDEX1.TOM4.SYSPRM(SYSUE1)_____ (SYSUE1)
 V---------------S: ADD, V: CHECK VARIABLES
 __ G1          ===> TEST_____
 _____
 __ G2          ===> _____
 _____
 __ G3          ===> _____
 _____
 __ G4          ===> &DSN,&SSN,&DATE,&TIME_____
 _____
 __ L1          ===> _____
 _____
 __ L2          ===> _____
 _____
 __ L3          ===> _____
 _____
 __ L4          ===> _____
 _____
              X EXIT, -PF3- BACK
```

Defining System Parameters (Option 3)

This option enables you to insert $actions$ that are not associated with an IF condition. You use the following screen to define system parameters.

```
TOM4120     SYSUE1 LIST  SYSTEM                           SERIOUS ERROR       !
 OPTION ===>
             E : MEMBER EDIT
 USER<->MONITOR/UTIL. INCOMPATIBLE LEVELS          R15 (DEC): 00000016
 MONITOR ===> TOM4
 FILE(MEMBER) ===> INDEX1.TOM4.SYSPRM(SYSUE1)_____ (SYSUE1)

 V---------------S: ADD, V: CHECK VARIABLES
 _ $ERR$      ! ===> SEND 'UE1 ERROR ON &SSN &AP',USER=(???),LOGON_____
 _____
 _ $NOSTAE$

 _ $NOSWAP$
 _ $NC-IEX$
 _ $NC-TEX$
 _ $NC-SEL$
```

Testing the Execution (Option 4)

This option enables you to simulate the L1GFIUE1 program. During a simulation, the SYSUE1 file is processed and IF lines are compared to the transfer conditions that you defined in the TEST SYSUE1 LIST screen shown below.

```
TEST SYSUE1 LIST                              NAMES INITIALIZED    !
OPTION ===>                         E : EDIT'SYSUE1', -ENTER- : TEST, -PF3- : END


EXIT  ===> L1GFIUE1   (L1GFIUE1/...)
//DDN ===> UPMMSSEE   (UPMMSSEE/...) MONITOR ===> TOM4
FILE(MEMBER) ...===> INDEX1.TOM4.SYSPRM(SYSUE1)               (//SYSUE1)

 FILE ...... ===> FICTST            (SYMBOLIC NAME)
 PARTNER ... ===> SCIPSR            (SYMBOLIC NAME)
 SUBSYS .... ===> 2                 (SUFFIX OF SUBSYS)
 DIRECTION . ===> R                 (T/R/*)
 TYPE ...... ===> E                 (S/E FILE, I/T SRT, *)
 MODE ...... ===> D                 (D/F)
 SRC ....... ===> 0000              (SRC/MASK TYPE=E/T MODE=F, ****)
 TRC ....... ===> 0000              (TRC/MASK TYPE=E/T MODE=F, ****)
 PRC ....... ===> 000               (PRC/MASK TYPE=E/T MODE=F, ****)
 RESTART ... ===> N                 (Y/N)
 FILE CODE . ===> 4040              (HEXA FILE CODE)
 ACCESS .... ===> I                 (I/O)
 LINK ...... ===> I                 (C/I/N/O/S/T/X)
            X EXIT, -PF3- END
```

You must define the SYSUE1 file that you want to test, and enter the monitor subsystem name TOM? and the conditions. Then, press <Enter> to display the process simulation.

```
L1GFIUE1 INDEX1.TOM4.SYSPRM(SYSUE1)
TOM40101TEXI02/01/0703:43:08FICTST  SCIPSR  A1111111.B2222222.C3333333.D4444444
.E5555555RS0S        0000*******

                                                           X3T 1 ORIGIN
      DESTINATION    :

02007 034308 TOM4 L1GFIUE1 PESIT-D1                                          V6R3
02007 034309 TOM4 L1GFIUE1                                                   0002
02007 034309 TOM4 L1GFIUE1 VAR:  $VAR$    G4:&DSN,&SSN,&DATE,&TIME           0002
02007 034309 TOM4 L1GFIUE1                                                   0003
02007 034309 TOM4 L1GFIUE1 IF: $IF$ FICTST  SCI*    * R E D  **** *** N ** * *  0003
02007 034309 TOM4 L1GFIUE1                                                   0005
02007 034309 TOM4 L1GFIUE1 IF: $IF$ $$ALL$$ $$ALL$$  * * * F **** *** Y ** * X  0005
02007 034309 TOM4 L1GFIUE1                                                   0006
02007 034309 TOM4 L1GFIUE1 CMD: (SIMULATE) $SUB$    JOB1                     0006
02007 034309 TOM4 L1GFIUE1 JOB1                                              0006
02007 034309 TOM4 L1GFIUE1                                                   0007
02007 034309 TOM4 L1GFIUE1 RET0 RC=??                     00000000          V6R3
02007 034309 TOM4 L1GFIUE1                                                   V6R3
02007 034309 TOM4 L1GFIUE1 FLG:    ?106?0 /11100/0111 0013/            /    V6R3
02007 034309 TOM4 L1GFIUE1 TOM20101TEXI02/01/0703:43:08FICTST SCIPSR A1111111…  V6R3
02007 034309 TOM4 L1GFIUE1 RS0S       00000000000                          V6R3
02007 034309 TOM4 L1GFIUE1 INT/EXT ID  -       00000000  ****/0000 ****/0000  V6R3
```

The first line indicates which SYSUE1 file is being processed. The next lines show the D1B2RUEX structure that is given to the general exit for testing conditions and replacing dynamic variables. The line number displays on the right side, and the Julian date and time displays on the left side. The L1GFIUE1 level is indicated on the right side (V6R3).

In the example above, a variable was found on line 0002, the condition, "the criteria doesn't match the test criteria" was found on line 0003, and the condition, "the criteria matches the test criteria and action line 6 is simulated," was found on line 0005. The last lines show the internal status for further debugging.

Note:	If you enter E in the option field, you can edit and update the SYSUE1 file that you selected. Then, press <PF3> to exit the file and start the simulation process.

Deleting Checkpoint Files (P1GFICP1)

A request checkpoint file is created for each file transfer in Connect:Express. When a transfer is successful or when the request is purged by an operator, this file is deleted. If the request is interrupted, this file stays on the disk until the transfer is restarted and successfully ended, or until the request is purged by an operator. If the monitor is cold started, however, the checkpoint files for interrupted requests are not automatically deleted. The requests are omitted and lost, but they are not purged. You can use this utility to delete any remaining checkpoint files.

This program is executed in its own address space.

Implementing the P1GFICP1 Utility

This program calls the IDCAMS utility, so a SYSPRINT file is required, as shown below.

```
//SYSPRINT DD SYSOUT=*
```

The P1GFICP1 program is authorized (APF), and WTO messages (ROUTCDE=11) are not sent to the OS/390 console.

Initialization Parameters

The program gets its parameters from the EXEC PARM field, as follows.

```
SSN= 'subsystem Name' , DELETE , CKI= 'Checkpoint files Index'
```

The table below describes the parameters in the EXEC PARM field.

Parameter	Description
SSN	Subsystem name in the form of TOM?.
DELETE	Action to be performed.
CKI Up to 17 characters	First index of the checkpoint files to be deleted. The full name structure is: 'cki'.'ssn'*.* Example†: If SSN=TOM2 and CKI=PROD.CHK, the checkpoint name would be PROD.CHK.TOM2CHKP.A0000125. A0000125 is the transfer request number.

† Positions of parameters are fixed.

Process Description

First, the program builds a list of all files with a checkpoint file name that matches the pattern below.

```
'cki'.'ssn'*.*
```

Connect:Express checks each file to see if it is associated with a request number, and any file with no request in the request table is deleted. Connect:Express looks for the requests with the L0B2Z20 module, and the file is deleted using IDCAMS.

Note: Connect:Express must be running to perform this process.

Viewing the Results

The table below lists the possible return codes sent by this utility.

Code	Description
0	Successful
4	'WARNING' – (invalid parameter, Monitor is not running …)
8	ERROR (the maximum of checkpoint files entries is reached, invalid subsystem name, …)
12	FATAL ERROR during processing

WTO Messages

The following table describes the conventions for WTO messages sent by this utility.

Character	Description
????????	Status or flag
++++++++	Numeric field, a number, a count, a code
////////	Alphanumeric , partner name for example
CK	'Checkpoint'

The table below lists the possible messages for this utility.

P1GFICP1 Messages

```
TOM+CP1 ERROR   ////////.////////     LOCATE RC(DEC)=++++++++ R0=++++++++
TOM+CP1 ERROR   ////////.////////     OBTAIN RC(DEC)=++++++++
TOM+CP1 ERROR   IDCAMS ++++++++       CHECK //SYSPRINT
        The following message found in the SYSPRINT file is normal:
        'IKJ56225I DATA SET X.Y.Z ALREADY IN USE, TRY LATER...'
        The file is still allocated by the monitor although the transfer request is ended.
TOM+CP1 ERROR   LIMITED TO STORAGE               (INTERNAL LIMIT->STOP)
TOM+CP1 ERROR   LIMITED TO ++++++++ ENTRIES     (RUN AGAIN THE JOB)
    NOTE: ONLY ++++++++ ENTRIES WERE PROCESSED
TOM+CP1 ERROR   LOGIC?                           (INTERNAL ERROR->STOP)
```

P1GFICP1 Messages (continued)

TOM+CP1 ERROR	OPEN	//SNAP	(USER ERROR, CHECK JCL)
TOM+CP1 ERROR	OPEN	//WTOPRINT	(USER ERROR, CHECK JCL)
TOM+CP1 ERROR	PARM	?	(USER ERROR->STOP, CHECK PARM)
		? = K - keyword missing or misplaced	
		? = L - PARM field missing or invalid length	
		? = N - invalid numeric field	
		? = M - 'MFY' option expected	
		? = P - parameter missing or misplaced	
TOM+CP1 LIMITED TO ++++++++ ENTRIES-		(RERUN ONE MORE TIME)	
TOM+CP1 NOTICE	DELETE APM CHECKPOINTS FOR CKI=/		
TOM+CP1 NOTICE	DELETED NO REQUEST ACTIVE	////////./////////	
TOM+CP1 NOTICE	KEPT REQUEST ACTIVE	////////./////////	
TOM+CP1 NOTICE	NO DSN FOUND		
TOM+CP1 NOTICE	NO MATCH		
TOM+CP1 NOTICE	PROGRAM EXIT		
TOM+CP1 NOTICE	TOTAL ++++++++ DSN(S)		
TOM+CP1 NOTICE	TOTAL ++++++++ CK(S) MATCHED		
TOM+CP1 NOTICE	TOTAL ++++++++ CK(S) ACTIVE		
TOM+CP1 NOTICE	TOTAL ++++++++ CK(S) DELETED		
TOM+CP1 SCANNED ++++++++ FILE(S)			
TOM+CP1 SEVERE PROGRAM ERROR			
TOM+CP1 WARNING FUNCTION INCOMPLETE			
TOM+CP1 WARNING IKJEHCIR R15(DEC)=++++++++			
TOM+CP1 WARNING CIR LOCATE RC (DEC)=++++++++ R0 (DEC) = ++++++++			
TOM+CP1 WARNING IGNORED NOT SEQ. ////////./////////			
TOM+CP1 WARNING IGNORED/NOT A CK ////////./////////			

Purging Transfer Requests (P1GFIRP1)

During transfer operations, some transfer requests stay in the Request Control Table (RCT) indefinitely. Since the size of the RCT is limited, it may be necessary to purge obsolete requests. The P1GFIRP1 program enables you to get a list of obsolete requests or to purge all requests for one Partner or group of Partners, or for one file or group of files, by age.

This program can be executed in its own address space.

Implementing the P1GFIRP1 Utility

The program gets a list of all transfer requests using the DISPLAY service of the L0B2Z20 interface. Then, requests that match the criteria are selected. If PURGE is requested, each selected request is purged using the MODIFY service of the L0B2Z20 interface.

P1GFIRP1 program is authorized (APF): WTO messages (ROUTCDE=11) are not sent to the OS/390 console. This utility requires that Connect:Express be running.

Initialization Parameters

The utility can be launched in different ways, depending on the criteria, using the following EXEC PARM field.

```
SSN= 'Subsystem Name' , Option , Criteria = mask
```

The table below describes the parameters in the EXEC PARM.

Parameter	Description
SSN	Subsystem name in the form of TOM?.
Option	Action to be performed: PURGE for deleting transfer requests or CHECK for listing obsolete requests.
Criteria	One of the keywords from the following list.

The following table describes the criteria that you can use when defining parameters in the EXEC PARM.

Criteria†	Mask
TPA	1 to 8 characters: Criteria for matching transfer partner name – either one partner name or a pattern Example: TRFUSER
SPN	1 to 8 characters: Criteria for matching session partner name – either one partner name or a pattern Examples: PARTNER, PART
SFN	1 to 8 characters: Criteria for matching symbolic file name – either one file name or a pattern Examples: FILE, F
HRS	2 characters: Criteria for matching the age of the request, in hours (from 00 to 24) Examples: 05, 10
DYS	2 characters: Criteria for matching the age of the request, in days (from 00 to 99) Examples: 05, 10
TRC	4 characters: Criteria for matching return code TRC. All the characters must be entered and you cannot use patterns like F*. Example: 2036
PRC	4 characters: Criteria for matching return code PRC. All the characters must be entered and you cannot use patterns like F*. Example: 2211
RQN	8 characters: Criteria for matching the request number. All the characters must be entered and you cannot use patterns like F*. Example: 00000129

† Positions of Parameters are fixed.

Viewing the Results

The table below lists the possible return codes sent by this utility:

Code	Description
0	Successful
4	'Warning' – Invalid Parameter, monitor not up …
8	Anomaly – state unknown.
12	FATAL ERROR during processing

WTO Messages

The table below describes the conventions for WTO messages sent by this utility.

Character	Description
????????	Status or flag
++++++++	Numeric field, a number, a count, a code
////////	Alphanumeric , partner name for example
CK	'Checkpoint'

The following table lists possible messages for this utility.

P1GFIRP1 Messages

`TOM+RP1 ERROR    PARM        ?              (USER ERROR->STOP, CHECK PARM)`
`                  ? = K - keyword missing or misplaced`
`                  ? = L - PARM field missing or invalid length`
`                  ? = N - invalid numeric field`
`                  ? = M - 'MFY' option expected`
`                  ? = P - parameter missing or misplaced`

`TOM+RP1 ERROR    DISPLAY RCT                    (INTERNAL ERROR->STOP)`
`TOM+RP1 ERROR    LOGIC?      ???                (INTERNAL ERROR->STOP)`
`TOM+RP1 ERROR    LOOP RCT                       (INTERNAL ERROR->STOP)`
`TOM+RP1 ERROR    OPEN    //SNAP      (USER ERROR, CHECK JCL)`
`TOM+RP1 ERROR    OPEN    //WTOPRINT  (USER ERROR, CHECK JCL)`
`TOM+RP1 ERROR    REQUEST UNK. ST2:?    ++++++++ FILE //////// PARTNER ////////`
`TOM+RP1 NOTICE   EMPTY    RCT`
`TOM+RP1 NOTICE   MATCHED  ++++++++ REQUEST(S)`
`TOM+RP1 NOTICE   PROGRAM  EXIT`
`TOM+RP1 NOTICE 'OPTION'  REQUESTS FOR SPN=/////`
`TOM+RP1 NOTICE 'OPTION'  REQUESTS FOR SFN=/////`
`TOM+RP1 NOTICE 'OPTION'  REQUESTS FOR HRS>=++`
`TOM+RP1 NOTICE 'OPTION'  REQUESTS FOR DYS>=++`
`TOM+RP1 NOTICE 'OPTION'  REQUESTS FOR RQN=++++++++`
`TOM+RP1 NOTICE   PURGED   ++++++++ REQUEST(S)`
`TOM+RP1 NOTICE   REQUEST  PURGED        ++++++++`
`TOM+RP1 NOTICE   REQUEST  NOT PURGED    ++++++++  T++++`
`TOM+RP1 NOTICE   REQUEST  REQUEST  ???????? ++++++++++ FILE "////////"` `PARTNER"////////"`
`TOM+RP1          CREATED  ++++++++ ++++++++`
`TOM+RP1 NOTICE   TOTAL    ++++++++ REQUEST(S)`
`TOM+RP1 SCANNED ++++++++  FILE(S)`
`TOM+RP1 SEVERE   PROGRAM  ERROR                 (INTERNAL ERROR->STOP)`
`TOM+RP1 WARNING FUNCTION INCOMPLETE            (IF NOT ALL REQUESTS PURGED)`

Using the IDCAMS Utility (P1GFIIDC and P1GFIID0)

The P1GFIIDC utility provides a way to use IDCAMS in a procedure, and can perform one IDCAMS function at a time. The call is done only with an EXEC PARM field, so no SYSIN file is needed.

The P1GFIID0 program is used in the same way, but it does not return a code if an error occurs. This is used when you don't care if an error occurs and you want to continue with the process.

These programs can be executed in their own address space.

Implementing P1GFIIDC and P1GFIID0

P1GFIIDC can be used either in batch or in the L1GFIUE1 general exit process as the parameter of a $PGMJ$ keyword.

```
SYSUE1 Example---- $PGMJ$  P1GFIIDC 'PARMS'
SYSUE1 Example---- $PGMJ$  P1GFIID0 'PARMS'
```

Both programs use standard parameters of the IDCAMS utility, and IDCAMS functions are called dynamically.

Note: This utility is used internally by Connect:Express for deleting temporary files such as *SYSPRM*[Ryhhmmss] used in the PDS UNLOAD process with member selection. An extra PARM keyword has been added. It is specific and can only be used with P1GFIIDC and P1GFIID0, as shown below.

$DELMEM$ SYSDEL to delete the member file PDS(MEMBER) with **// *SYSDEL DD DSN=PDS (MEMBER).***

Viewing the Results

The table below lists the possible return codes sent by this utility.

Code	Description
0	Successful
4	'WARNING' – return code from IDCAMS is in the SYSPRINT file
8	ERROR detected during processing

The following screen shows examples.

```
BATCH:
  //*
  //GO       EXEC   PGM=P1GFIIDC,
  //  PARM='$DELMEM$  SYSDEL'
  //STEPLIB   DD      DISP=SHR,DSN=INDEX1.TOMV???.LOADLIB
  //SYSPRINT  DD      SYSOUT=*
  //SYSDEL    DD      DSN=USR.PDS(MEMBER),DISP=SHR
  //SYSUDUMP  DD      SYSOUT=*

L1GFIUE1:
  $PGMJ$   P1GFIIDC DEL &API
```

Managing Partners and Files Directories with Batch Processes (P0B2DIR)

This utility provides batch functions equivalent to those found in the TSO/ISPF Files and Partners directories. For each directory, you can:

❖ Create an entry – ADD function

❖ Update an entry – UPDATE function

❖ Delete an entry – DELETE function

❖ Create or update an entry – REPLACE function

❖ List of all entries – LIST function

❖ Update or not internal tables (PCT or FCT) – REFRESH option

❖ Checkpoint function – RESTART option

❖ Trace function –TT option

This program is executed in its own address space.

Implementing P0B2DIR

This program runs in a PL1 language environment. To update correctly, the JCL flow must point to the appropriate system files and the VSAM directory. Examples are provided in the *SAMPOPT* library $P0B2DIP and $P0D2DIF.

```
//GO       DD        PGM=P0B2DIR
//STEPLIB  DD        DISP=SHR,DSN=PROD.TOM410.LOADLIB  <=== TOM
//*                  DSN=PLI.PLILINK.ESA430,DISP=SHR  <== PL/1 TRANSIENT
//                   DSN=CEE.V1R5MO.SCEERUN,DISP=SHR
//*
//SYSPAR   DD        DISP=SHR,DSN=INDEX.TOM4.SYSPAR <=== VSAM
//*
//PLIDUMP  DD        SYSOUT=* <-- OPTIONAL FOR PLI DEBUGGING+TRACE
//SNAP     DD        SYSOUT=* <-- OPTIONAL FOR PLI DEBUGGING+TRACE
//SYSUDUMP DD        SYSOUT=* <-- OPTIONAL FOR ASM DEBUGGING+TRACE
//SNAPDUMP DD        SYSOUT=* <-- OPTIONAL FOR ASM DEBUGGING+TRACE
//SYSPRINT DD        SYSOUT=* <=== FOR LISTING
//*----------------------------------------------------------------
//SYSIN DD *
```

The SYSIN file has fields similar to those you find with the TSO/ISPF interface.

The REFRESH = YES option requires that Connect:Express be started. If REFRESH = NO, only the VSAM file is updated. The monitor is not required to be running.

Initialization Parameters

The program is executed from its SYSIN file and one entry is processed at a time.

```
//SYSIN DD *
SUBSYS                 = 'TOM2',
FUNCTION               = 'REPLACE' ,
REFRESH                = 'YES',
TT                     = 'TT',

PARTNER.NAME           = 'Z2',
PARTNER.PASSWORD       = 'Z2',
PARTNER.NOTE           = 'NOTE';
```

The list below outlines the rules and guidelines for entries in the SYSIN file.

❖ Partner definition parameters are in the form PARTNER.*

❖ File definition parameters are in the form FILE.*

❖ Each Parameter has one parameter per line. A comma (',') at the end of the line is a continuation character, and a semicolon (';') indicates the end of the list.

❖ No comment line is authorized in the SYSIN file.

❖ If a parameter is duplicated, the second one is used.

❖ If a parameter is too long, it is truncated with no information message.

❖ A dash ('-') in a field can be used for clearing a field in the VSAM directory and in the internal control table (PCT or FCT). A blank in a field is equivalent to omitting this parameter. In this case, no update is done in the VSAM directory or in the internal control table (PCT or FCT).

❖ Unchanged fields can be omitted in with the Add, Update, and Replace functions. You can update one field in the Partner or File definition.

❖ If the option REFRESH=NO is set, no control is performed by Connect:Express. Only syntax controls and field value range controls are performed. During Connect:Express initialization, the control process could detect an error such as an invalid entry with status UNUSABLE. This entry can only be updated before being available for transfer operations.

Note: The FILE.TYPE field must contain a binary value, for example, FILE.TYPE='00000000B' for a sequential file type.

Process Description

The P0B2DIR utility checks parameters from the SYSIN file. If REFRESH=YES, parameters are sent to the monitor control process. If all controls are successful, the VSAM files and internal control tables are updated and a report is written that can include any of the following three message types:

- ❖ Messages from the utility about the syntax.
- ❖ Messages from the monitor interface.
- ❖ Messages from VSAM interface.

The table below lists the possible return codes sent by this utility.

Code	Description
0	Successful
4	'WARNING' – anomaly detected during processing
8	ERROR detected during processing

The next two screens show examples of definitions for a symbolic Partner and a symbolic File.

Definition of a Symbolic Partner

```
//*----------------------------------------------------------------
//* TOM:  //SYSPAR AND PCT UPDATE
//*----------------------------------------------------------------
//GO       EXEC   PGM=P0B2DIR
//STEPLIB  DD     DISP=SHR,DSN=PROD.TOM410.LOADLIB  <=== TOM
//*        DD     DSN=PLI.PLILINK.ESA430,DISP=SHR  <== PL/1 TRANSIENT
//        DD     DSN=CEE.V1R5M0.SCEERUN,DISP=SHR
//*
//SYSPAR   DD     DISP=SHR,DSN=PROD.TOM4.SYSPAR  <=== VSAM
//*
//PLIDUMP  DD     SYSOUT=* <-- OPTIONNAL FOR PLI DEBUGGING+TRACE
//SNAP     DD     SYSOUT=* <-- OPTIONNAL FOR PLI DEBUGGING+TRACE
//SYSUDUMP DD     SYSOUT=* <-- OPTIONNAL FOR ASM DEBUGGING+TRACE
//SNAPDUMP DD     SYSOUT=* <-- OPTIONNAL FOR ASM DEBUGGING+TRACE
//SYSPRINT DD     SYSOUT=* <=== FOR LISTING
//*----------------------------------------------------------------
//SYSIN DD *
SUBSYS                 = 'TOM4',
FUNCTION               = 'REPLACE' ,
REFRESH                = 'YES',
TT                     = 'TT',

PARTNER.NAME           = 'Z2',
PARTNER.PASSWORD       = 'Z2',
PARTNER.STATE          = 'E',
PARTNER.ALIAS_NAME     = '-',
PARTNER.ALIAS_PASSWORD = '-',
PARTNER.RACF_USER      = '-',
PARTNER.RACF_GROUP     = '-',
PARTNER.APM_CLASS      = 'A',
PARTNER.RESTART        = 'Y',
PARTNER.TYPE           = 'T',
PARTNER.PROTOCOL       = '52',
PARTNER.EFF_TOTAL      = '004',
PARTNER.EFF_IN         = '002',
PARTNER.EFF_OUT        = '002',
PARTNER.SECURITY       = '-',
PARTNER.SLD            = '-',
PARTNER.LINK_TYPE      = 'M',
PARTNER.LINK_LIST      = 'XSI',
PARTNER.SNA_LUNAME     = 'LUNAME22',
PARTNER.SNA_LOGMODE    = '-',
PARTNER.SNA_LOGDATA    = '-',
PARTNER.SNA_LU62_DISC  = '-',
PARTNER.X25_MSC        = 'M',
PARTNER.X25_DIAL       = '123456789',
PARTNER.X25_SUBAD      = '98',
PARTNER.X25_USER_DATA  = 'C0',
PARTNER.X25_GFA        = '-',
PARTNER.X25_TAXATION   = '1',
PARTNER.TCP_ADDRESS    = '123.123.123.123',
PARTNER.TCP_HOST       = 'UNUSED',
PARTNER.TCP_PORT       = '7000',
PARTNER.FTP_PROF       = '-',
PARTNER.FTP_PASV       = '-',
PARTNER.FTP_RIGHTS     = '-'
PARTNER.NOTE           = 'NOTE';
//
//*=========================================
```

Definition of a Symbolic File

```
   //*---------------------------------------------------------------
   //* TOM:  //SYSFIL AND FCT UPDATE
   //*---------------------------------------------------------------
   //GO       EXEC    PGM=P0B2DIR
   //STEPLIB  DD      DISP=SHR,DSN=PROD.TOM410.LOADLIB  <=== TOM
   //*        DD      DSN=PLI.PLILINK.ESA430,DISP=SHR  <== PL/1 TRANSIENT
   //         DD      DSN=CEE.V1R5M0.SCEERUN,DISP=SHR
   //*
   //SYSPAR   DD      DISP=SHR,DSN=PROD.TOM4.SYSPAR  <=== VSAM
   //*
   //PLIDUMP  DD      SYSOUT=* <-- OPTIONNAL FOR PLI DEBUGGING+TRACE
   //SNAP     DD      SYSOUT=* <-- OPTIONNAL FOR PLI DEBUGGING+TRACE
   //SYSUDUMP DD      SYSOUT=* <-- OPTIONNAL FOR ASM DEBUGGING+TRACE
   //SNAPDUMP DD      SYSOUT=* <-- OPTIONNAL FOR ASM DEBUGGING+TRACE
   //SYSPRINT DD      SYSOUT=* <=== FOR LISTING
   //*---------------------------------------------------------------
   //SYSIN DD *
   SUBSYS               = 'TOM4',
   FUNCTION             = 'DELETE'
   FUNCTION             = 'ADD',
   FUNCTION             = 'UPDATE',
   REFRESH              = 'YES',

   FILE.NAME            = 'FILETEST',
   FILE.STATE           = 'E',
   FILE.DIRECTION       = '*',
   FILE.PARTNER_SENDER   = '*',
   FILE.PARTNER_RECEIVER = '*',
   FILE.PRIORITY        = '1',
   FILE.DEFINITION      = 'D',
   FILE.ALLOCATION      = '2',
   FILE.TYPE            = '00000000'B, // '10001000' = VSAM
   FILE.PRESENTATION    = '01',
   FILE.MEMBER          = '-',
   FILE.SECURITY        = '-',
   FILE.DSN             = 'INDEX1.&FILENAM.&REQDAT.&REQTIM',
   FILE.DISPOSITION     = 'NEW',
   FILE.GDG             = '-',
   FILE.UNIT            = '-',
   FILE.VOLSER1         = '-',
   FILE.VOLSER2         = '-',
   FILE.VOLSER3         = '-',
   FILE.VOLSER4         = '-',
   FILE.VOLSER5         = '-',
   FILE.SMS_DATA_STOR   = '-',
   FILE.SPACE_TYPE      = 'CYL',
   FILE.SPACE1          = '0001',
   FILE.SPACE2          = '0001',
   FILE.DIRECTORY_COUNT = '-',
   FILE.LRECL           = '00080',
   FILE.BLKSIZE         = '08000',
   FILE.RECFM           = 'FB',
   FILE.RETENTION       = '-',
   FILE.REMOTE_DSN      = '-',
   FILE.FTP_OPT         = '-',
   FILE.EXIT_START_SEND      = 'L1GFIUE1',
   FILE.COMMAND_START_SEND   = 'S W',
   FILE.EXIT_END_SEND        = 'L1EX£AE2',
   FILE.COMMAND_END_SEND     = 'S X',
   FILE.EXIT_START_RECEIVE   = 'L1GFIUE1',
   FILE.COMMAND_START_RECEIVE = 'S Y',
   FILE.EXIT_END_RECEIVE     = 'L1EX£AE2',
   FILE.COMMAND_END_RECEIVE  = 'S Z';
   //*==========================================
```

Appendix A

Supporting Lists, Tables, and Examples

This appendix provides reference lists of action keywords, dynamic variables, and error and severe messages. There are also examples of a SYSUE1 file and a SYSOUT file.

Action Keywords

The table that follows lists all the action keywords provided in alphabetic order.

Keyword	Description	Parameters and Example
300	Calls the standard utility P1B2P300. This utility performs actions on a File, Partner, or Request. Note: An action on the current request can result in an error because the monitor might still process this request.	PARM field of the utility. Refer to the User Guide for more information. Example: S=&SSN/P=&PARTNID/H Disable the current partner
$ABEND$	This keyword results in ABEND 001. (The 'ESTAE' of the exit is definitely disabled.)	No parameter
$CAPS$	All characters in SYSUE1 are set to upper case.	This is the default. No parameters.
CMD	The command input is performed. If an error occurs, the transfer is interrupted with the return codes TRC=4908 or 4008 and PRC =3312.	Command and parameters of the command. Examples: 1. TSO Message with local variables &G1 and &L2 (defined by previous VAR): SEND 'ERROR &G1',USER=(&L2),LOGON 2. OS/390 Command with dynamic variable &DSN (expanded at transfer time): S PROCTEST D=&DSN

Keyword	Description	Parameters and Example
DYA	The standard allocation program L1APMDYA is called. Allocation is performed from information received by the transfer protocol. Received files are blocked in a simple way. SMS files are not supported. For example, a fixed format 80 character record file is automatically allocated with a block size that is a multiple of 80, but less than 32760. LRECL=80, RECFM=FB, and BLKSIZE=32720.	The symbolic file must be defined with allocation rule = 3, the general exit called at beginning and end of transfer. With allocation rule=3, the user exit is responsible for both allocation and deallocation. **Allocation**: The only parameter is the data set name. The appropriate dynamic keyword is &DSN. IF …. Start of transfer. DYA &DSN **Deallocation**: IF ……End of transfer (OK or not). DYA deallocation does not require the dsn.
ERR	This keyword is processed in case of error during the general exit. For example, a syntax error in the SYSUE1 file, an abend in a user program, a getmain problem, or no matching IF conditions.	Command to be executed. Example: SEND '&SSN &AP UE1 ERR.,USER=(USR4) Send a TSO message with the subsystem name and the APM number to the user USR4.
$EXIT$	Calls a user program in the way an exit is called with the parameter structure D1B2RUEX. If an error occurs, forwards the TRC and PRC return codes.	Name of the user exit to be called Example: L1EX£AE2 Calls the standard trace exit. Note: L1GFIUE1 cannot call itself
$FNFY$	Calls the P1B2PNFY program to store a Connect:Express notification.	EXEC PARM field of P1B2PNFY. Example: IND=INDEX1.INDEX2

JCL	Calls program P1B2PJCL which starts a procedure with one parameter input. The following DD cards are required in the APM or EAS (FTP) procedure:
	`//SYSJOB indicates the job to be started`
	`//SYSPRT DD SYSOUT=V,HOLD=YES`
	`//SYSPCH DD SYSOUT=(,INTRDR)`
		Parameter of the procedure. One single parameter.
		Example:
		N=&DSN
		An Example is provided in *SAMPLIB* EX£PJCL

LOG	Calls program L1B2LOG that writes a message in the SYSLOG file of the APM.	User message.
$NC-IEX$	If no beginning of transfer condition is found for the current transfer, no error is issued.	No parameters.
$NC-TEX$	If no end of transfer condition is found for the current transfer, no error is issued.	No parameters.
$NC-SEL$	If no initialization or termination of transfer condition is found for the current transfer, no error is issued.	No parameters.
$NOCAPS$	Lower case characters are retained. This is how you can send lower case characters in protocol fields with $PI37$ and $PI99$ keywords. The option is active until the next $CAPS$ keyword.	No parameters.
$NOPI37$	With PeSIT protocol, the PI37 parameter is set to BLANK. No dsn or user field is sent.	No parameter
$NOSTAE$	The ESTAE of L1GFIUE1 is activated. This results in a dump, if any problem occurs.	No parameter
$NOSWAP$	The APM or EAS is set to non swappable. This option can be useful in debugging.	No parameter

Keyword	Description	Parameters and Example
PGM	Calls a user program using the branch method. Register R1 points to a word with the address of the parameters of the program, according to IBM standards. **Note**: If a program requires a DD card DISP=NEW, it cannot be called simultaneously by two tasks of the same address space, and an allocation error can occur. The allocation must be dynamic, and the DDNAME must be calculated from a root concatenated with the number of the effector in which it is executing.	Name of the program followed by its parameters. The parameters of the program are given with a chain of characters, placed column 9, in the form 'Prog ' Parameters' Example: L1EXSNAP &SSN&AP&EF&FILENAME If the transfer is executed in APM number 02, effector number 08 of TOMP, for symbolic file SYMBF, the parameter line will result in: TOMP0208SYMBF Note: L1GFIUE1 cannot call itself.
$PGMJ$	Calls a user program using the exec method. Register R1 points to a word with the length of the parameter field EXEC PARM, followed by a word with the address of the PARM field, according to IBM standards. Note: If a program requires a DD card DISP=NEW, it cannot be called simultaneously by two tasks of the same address space, and an allocation error can happen. The allocation must be dynamic, and the DDNAME must be calculated from a root concatenated with the number of the effector in which it is executing.	Name of the program followed by its parameters. The program is called in the form 'prog.' 'champ PARM' equivalent to EXEC PGM=pgm,PARM='Parameters' Examples: 1. P1GFIID0 DEL &DSN Calls IDCAMS to delete the current file &DSN. If the file does not exist, no return code is set.) 2. L1GFITS1 MSG=HELLO &FILENAM , ULN=USR9,USr11,UER=USR8 Calls L1GFITS1 utility.
$PI11$	PeSIT protocol: A string is sent in the Pi11 parameter.	String to be sent (8 characters maximum).
$PI12$	PeSIT protocol: A string is sent in the Pi12 parameter.	String to be sent (76 characters maximum).
$PI37$	PeSIT protocol: A string is sent in the Pi37 parameter.	String to be sent (80 characters maximum). Example: If VAR is defined by L3:C:ùTRFù&PARTNIDù&FILENAM.TXT &L3 Pi37 will carry a file name in the PC form. Character 'ù' is for '\'.
$PI61$	PeSIT protocol: A string is sent in the Pi61 parameter.	String to be sent (24 characters maximum).
$PI62$	PeSIT protocol: A string is sent in the Pi62 parameter.	String to be sent (24 characters maximum).
$PI99$	PeSIT protocol: A string is sent in the Pi99 parameters. The Partner must be of type Other.	String to be sent (254 characters maximum). Example: If VAR is defined by L3:C:ùTRFù&PARTNIDù&FILENAM.TXT &L3 Pi99 will carry a file name in the PC form. Character 'ù' is for '\'.
$PI99RD$	This keyword provides the name of a data set that you can use to read the Pi99 to send to the current PeSIT partner. The file must contain one fixed format 254-character record. The Partner must be of type Other.	Data set name Example: PROD.&FILENAM.PI99.D&DATE

Keyword	Description	Parameters and Example
$PI99WD$	This keyword provides the name of a data set that you can use to create and write the Pi99 received from the current PeSIT partner. The file is allocated with a fixed format, a record length of 254 characters and under control of SMS. The partner type must be of type Other. If SMS is not used it is possible to use keywords $W-UNT$, $W-VOL$ and $W-RET$.	Data set name Example: PROD.&FILENAM.PI99.&REQNUMB
PRC	You can force a PRC return code. The TRC return code is then forced to TRC=4001 or 4901.	Value (valid for the protocol) of the return code. Examples: 312 216
REQ	Calls the P1B2PREQ program for making a transfer request. Note: Using the current data set can result in an error if the file is not yet deallocated by Connect:Express.	EXEC PARM field of P1B2PREQ. Example: SSN=&SSN, SFN=FICACK,SPN=&PARTNID Transfer request for the symbolic file FICACK with the current Partner. Other parameters are fixed in the directories.

$REXX$	Calls a user REXX procedure with its parameters. The following DD cards are required in the APM or EAS (FTP) procedure:
	`//SYSEXEC  DD points to your execs`
	`//        DD points to ispclib of Express`
	`//SYSTSPRT DD SYSOUT=V,HOLD=YES`
	`//SYSTSIN  DD DUMMY,DCB=(LRECL=80,RECFM=FB)`
		Name of a procedure followed by its parameters, in the form:
		'procedure' 'Parameters'
		Example:
		UE1MSG UID(&UID) DSN(&DSN)
		Calls the UE1MSG procedure with the requestor name and the current data set name.

SIM	Simulation: The command execution is simulated, and the result is written in the SYSOUT file 'UP….'.	Command to simulate the execution with the parameters: 'Command' 'Parameters' Example: S PROC F=&FILENAM, R=&REQNUMB, D=&DSN Dynamic variables are expanded and the string is written in SYSOUT for you to control.
$SNFY$	Calls the P1B2PNFY program to send a Connect:Express notification.	EXEC PARM field of P1B2PNFY. Example: SRV=1.2.3.4,PRT=1000, RCI=INDEX1.INDEX2
SUB	A JCL is submitted and Dynamic variables are expanded from current information. This process gives you the most flexible way of sending the current information to the application.	Name of a procedure. When the dynamic variables are expanded, the job is submitted to the internal reader. 1. Examples: INDEX.&PARTNID.&FILENAM: Submits the procedure whose name depends on the current Partner and the current symbolic file. 2. &DSN: Submits the received data set. 3. INDEX.SUB.JCLLIB(&FILENAM): Submits the member whose name is the current symbolic file.

Keyword	Description	Parameters and Example
VAR	This keyword enables you to define internal variables. Using this keyword, you can overcome the 80-character limit for each line in the file. The limit of an expanded line is 120 characters. A local variable (&Li) can be placed anywhere in the file and its value can be changed locally. A global variable (&Gi) is defined only once.	Any characters string, including dynamic variables Li:'string or Gi:'string'' Examples: Global variable: G2:P&PRC,T&TRC,S&SRC,&SPN,&TYP,&CT 3,&LNK,&A,RST=&RT,U=&UID

Local variable: L1:USR0004,USR0008,USR0012

SEND 'ERROR &G2' L1GFITS1 MSG=HELLO &FILENAM , ULN=&L1,UER=USR0001 |
$W-VOL$	Used for new file allocation, without SMS. Must be used with $PI99WD$ keyword	Volume name
$W-UNT$	Used for new file allocation, without SMS. Must be used with $PI99WD$ keyword.	Unit name
$W-RET$	Used for new file allocation, without SMS. Must be used with $PI99WD$ keyword.	Number of retention days
$WAIT$	Results in a one minute timer.	No parameter

Dynamic Variables

The table below lists all dynamic variables in alphabetical order. Keywords that are PeSIT specific are marked with an asterisk ('*'), and equivalent symbols are grouped.

During processing, the general exit looks at input from the D1B2RUEX structure provided in the *MACLIB* library to resolve keywords. Keywords are replaced by the corresponding field in the input structure. If a requested field is blank or contains zeroes, it is replaced with 'NONE'.

Keyword	Lg	D1B2RUEX	Description
&A	1	UEXCNTYP	Access type (Incoming / Outgoing).
&AP	2	UEXAPMNB	Number of the APM.
&BDATE &REQDAT	7	UEXDATE	Local date: Dyymmdd.
&BTIME &REQTIM	7	UEXTIME	Local time: Hhhmmss.
&CT3	3	UEXCALTY	Call type IEX / TEX.
&CTY	1	UEXCALTY	Call type Initialization / Termination.
&DATE	8	UEXDATE	Local date time YY/MM/DD-HH:MM:SS.
&DIAGI, &TRCX	5	UEXTRC	TRC + origin Locale (L) or remote (R). TRC=3001L or TRC=3001R.

Keyword	Lg	D1B2RUEX	Description
&DIAGP, &PRC	3	UEXPRC	PRC.
&DIR	1	UEXDRCT	Direction of the transfer (Transmission / Reception).
&DSN &FNAME	44	UEXDSN	Local data set name.
&DST *	8	U03DESTN U05DESTN	Destination of the transfer, truncated to 8 characters. Can be different than the session partner name.
&EF	2	UEXEFFNB	Effector number.
&EXTDSN	44	U03USDF U05USDF	Remote Data set name from the PeSIT Pi99. The 26 character ODETTE file name.
&EXTLAB &LAB	80	U03FLABL U05FLABL	PeSIT: File label, Pi37. ODETTE: The 26 character ODETTE file name.
&FA *		U03FTYP U05FTYP	File type, Pi11.
&FILENAM &IDF	8	UEXDDNM	File symbolic name.
&FNAME &DSN	44	UEXDSN	Local data set name.
&FTY	1	UEXALORG	File organization **S**equential, **V**sam, **P**ds.
&IDF &FILENAM	8	UEXDDNM	File symbolic name.
&IDT &REQNUMB	8	UEXREQNB	Local request number, Axxxxxxx, Bxxxxxxx….
&LAB &EXTLAB	80	U03FLABL U05FLABL	PeSIT: File label, Pi37. ODETTE: The 26 character ODETTE file name.
&LNK	1	UEXLNKTP	Type of network link (**S**na / **X**25 / **IP** / **T**3270 / **C**lu6.2).
&MBR	8	U03USDF U05USDF	PDS member of the local file.
&ORG	8	U03ORIGN U05ORIGN	Origin of the transfer, truncated 8 characters. Can be different than the session partner name.
&PART &PARTNID &SIT,&RID, &SDEST	8	UEXPART	Session partner name.
&PARTNID &PART &SIT,&RID, &SDEST	8	UEXPART	Session partner name.
&PI11	2 8	U03FTYPE U05FTYPE	PeSIT protocol: Pi11 received from a PeSIT D or PeSIT E partner
&PI12	76	U03FNAME U05FNAME	PeSIT protocol: Pi12 received from a PeSIT D or PeSIT E partner

Keyword	Lg	D1B2RUEX	Description
&PI37	80	U03FLABL U05FLABL	PeSIT protocol: Pi37 received from a PeSIT D or PeSIT E partner
&PI37NB	80	U03FLABL U05FLABL	PeSIT protocol: Pi37 received from a PeSIT D or PeSIT E partner. Blank trailer is suppressed
&Pi61	24	U05ORIGN	PeSIT E protocol: Pi61 received from a PeSIT E partner
&Pi62	24	U05ORIGN	PeSIT E protocol: Pi62 received from a PeSIT E partner
&Pi99	44	U03FUSDF U05FUSDF	PeSIT protocol: 44 first characters of the Pi99 received from a PeSIT D or PeSIT E partner. The partner type must be of type Other.
&PRC, &DIAGP	3	UEXPRC	Return code PRC.
&PTY	1	UEXPARTY	Partner type (**T**om / **O**ther / **A**pplication)
&REQ	8	UEXREQN	Local request number – numeric format
&REQDAT &BDATE	7	UEXDATE	Local date: Dyymmdd.
&REQNUMB &IDT	8	UEXREQN	Local request number, Axxxxxxx, Bxxxxxxx….
&REQTIM &BTIME	7	UEXTIME	Local time Hhhmmss.
&RID, &SIT, &PARTNID &PART, &SDEST	8	UEXPART	Session partner name.
&RT	1	UEXALRST	Transfer retry flag **Y**es / **N**o.
&SDEST, &PARTNID &PART &SIT,&RID,	8	UEXPART	Session partner name.
&SID &SSN	4	UEXSSNAM	Local subsystem Name.
&SIT, &RID, &PARTNID &PART, &SDEST	8	UEXPART	Session partner name.
&SPN	1	UEXPROTN	Protocol number 3 or 5 for PeSIT, 2 for Odette, 6 for FTP, 4 for ETEBAC3.
&SRC	4	UEXSRC	System return code.
&SSN &SID	4	UEXSSNAM	Local subsystem name.
&STY	1	UEXCNTYP	Type of connection **I**nput / **O**utput.
&TIME	8	UEXTIME	Local time Hhhmmss.
&TRC	4	UEXTRC	TRC return code.
&TRCX, &DIAGI	5	UEXTRC	TRC + origin: Locale (L) or Remote (R) TRC=3001L or TRC=3001R

Keyword	Lg	D1B2RUEX	Description
&TRFID *	8	UEXTRFID	Transfer identification, pi13 editable hexadecimal (characters 0-9, A-F).
&TYP	1	UEXTYPE	Call type: **A**pplication / **I**nitialization / **E**nd / **F**ail / **S**ystem abend.
&UID	8	U05FUSDF	Name of the transfer requestor, user, job, or remote Ad Hoc user.
&USRVAR1	8	U03USDF U05USDF	Last index of the remote data set name. From PeSIT Pi99. The partner type must be TOM.
&USRVAR2	8	U03USDF U05USDF	Before last index of the remote data set name. From PeSIT Pi99. The partner type must be TOM.

Error and Severe Messages

Message	Description
`'DCBO-UE1'`	DCB overlay. Check parallelism. Contact support.
`'ERROR  ESTAE  INIT'`	ESTAE error. Contact support.
`'ERROR  MODESET NZ'`	MODESET error , non zero. Contact support.
`'ERROR  MODESET Z'`	MODESET error , zero. Check APF libraries. Contact support.
`'ERROR  NO AUTHPR='`	Program not authorized. Check Asset protection. Contact support.
`'ERROR  PARAMETERS'`	Invalid call. Check the message and documentation.
`'ERROR  SYSUE1 $DYA$ ........'`	L1APMDYA allocation module error. Check allocation rule and transfer direction
`'ERROR  SYSUE1 $VAR$ ........'`	Syntax error. Check the VAR card.
`'ERROR  SYSUE1 $VAR$ RESET ........'`	Syntax error. Check the VAR card (for example a global variable cannot be changed).
`'ERROR  SYSUE1 CARD COUNT'`	Syntax error. Check the number of cards.
`'ERROR  SYSUE1 KEYWORD ........'`	Syntax error. Check the keyword.
`'ERROR  SYSUE1 KKK ...'`	Syntax error. Check the condition 'KKK' in the IF card.
`'ERROR  SYSUE1 LINK $DYA$  ........'`	L1APMDYA allocation module error. Check the LOADLIB. Contact support.
`'ERROR  SYSUE1 LINK $JCL$  ........'`	P1B2PJCL module error. Check the LOADLIB. Contact support.
`'ERROR  SYSUE1 LINK $PGM$  ........'`	User program error. Check the program.
`'ERROR  SYSUE1 LINK $REQ$  ........'`	P1B2PREQ module error. Check the LOADLIB. Contact support.
`'ERROR  SYSUE1 LINK $REXX$ ........'`	User REXX procedure. Check the REXX procedure.
`'ERROR  SYSUE1 LOAD $EXIT$ ........'`	User exit error. Check the exit.
`'ERROR  SYSUE1 SEQUENCE'`	Syntax error. Check cards sequence (IF with no conditions, two consecutive IF …)
`'ERROR  SYSUE1 SYMBOLIC'`	Syntax error. Check the symbol.
`'ERROR  SYSUE1 TRUNCATED'`	Syntax error. Check the number of symbol used.
`'ERROR  UEX........'`	D1B2RUEX structure error. Contact support.

Message	Description
`'SEVERE CLOSE INTRDR ERR.'`	CLOSE error on the internal reader. Contact support.
`'SEVERE INTRDR   ERR.DYN-............'`	Allocation error on the internal reader. Contact support.
`'SEVERE OPEN   INTRDR ERR.'`	OPEN error on the internal reader. Contact support.
`'SEVERE OPEN UP? ERR.'`	OPEN error on the SYSPRINT file. Contact support.
`'SEVERE READ SYSUE1 SYNAD'`	READ error on the SYSUE1 file. BROWSE the file. Contact support.
`'SEVERE SYSPRINT ERR.DYN-............'`	Allocation error on the SYSPRINT file. Contact support.
`'SEVERE SYSUE1 CLOSE'`	CLOSE error on the SYSUE1 file. Contact support.
`'SEVERE SYSUE1 OPEN'`	OPEN error on the SYSUE1 file. Contact support.

SYSUE1 File Example

```
         * IF SOME INTERNAL PROCESS ERROR, PERFORM THIS COMMAND :
          $ERR$    SEND '&SSN L1GFIUE1 ERROR,&L,&FILENAME,&PARTNID',USER=(USR0004)
          ***************************************************************************
          * ASSIGN VALUES/VARIABLES TO LIST VARIABLES.
         * GLOBAL LIST (&G) CANNOT BE CHANGED :
          $VAR$    G1 >&FILENAM,&DIR: &PARTNID,&REQNUMB,&SSN,&DATE,&TIME
          $VAR$    G2 P&PRC,T&TRC,S&SRC,&SPN,&TYP,&CT3,&RT,U=&UID
          $VAR$    G3 DSN=&DSN
          $VAR$    G4 EXTDSN=&EXTDSN
         * LOCAL LIST (&L) CAN BE CHANGED ANYWHERE :
          $VAR$    L1 USR0003,USR0004,USR0005,USR0008
          $VAR$    L2 USR0004,USR0008
          $VAR$    L4 USR0004
          ***************************************************************************
          ***************SAMPLES WITH NORMAL DEFINED SYMBOLIC FILE-NAME***********
          ***************************************************************************
          $IF$ $******* SCIPSR** * R E F 3*** *** Y **
          $EXIT$   L1APMCFJ
          $IF$ ******* SCIPSR** * R E F 3*** *** Y **
          $CMD$    SEND 'ERROR &G1',USER=(&L2),LOGON
          $CMD$    SEND './. &G2',USER=(&L2),LOGON
          ***************************************************************************
          *    F='FTYL', P='SCIPSR**', TOM3 RECEPTION, END, FAIL TRC 8***
          $IF$ FTYL    SCIPSR** 3 R E F 8*** *** Y **
          *-------
          *        REAL CMD : 'SEND .....'
          $CMD$    SEND '&G1',USER=(&L4),LOGON
          $CMD$    SEND './. LN=&LNK SP=&SPN PT=&PTY ',USER=(&L4),LOGON
          $CMD$    SEND './. U1=&USRVAR1 U2=&USRVAR2',USER=(&L4),LOGON
          $CMD$    SEND './. &DSN &MBR FT=&FTY',USER=(&L4),LOGON
          $CMD$    SEND './. OR=&ORG DE=&DST',USER=(&L4),LOGON
          $REXX$   TYLREX F=&FILENAM P=&PARTNID RT=&RT TRC=&TRCX PRC=&PRC
          $EXIT$   L1APMCFJ
          *   ANY FILE, ANY PARTNER, ANY FAIL CASE, EVEN AT RESTART
          *-------
          *        SIMULATE: 'S USRPROC'
          $SIM$    S USRPROC F=&FILENAM,P=&PARTNID,R=&REQNUMB,D=&DSN
          *        PERFORM : 'SEND .....'
          $CMD$    SEND 'FAIL &G1',USER=(USR0009),LOGON
          ***************************************************************************
          *    $   FILE, ANY PARTNER, ANY FAIL CASE, EVEN AT RESTART
          $IF$ $******* $$ALL$$  * * * F **** *** Y **
          *-------
          *        SIMULATE: 'S USRPROC'
          *$SIM$     S USRPROC F=&FILENAM,P=&PARTNID,R=&REQNUMB,D=&DSN
          *        PERFORM : 'SEND .....'
          $CMD$    SEND 'RESTARTING &G1',USER=(USR0009),LOGON
          $CMD$    SEND './. &G2',USER=(USR0009),LOGON
          $EXIT$   L1EXSNAP
          *   ANY FILE, ANY PARTNER, ANY FAIL CASE, EVEN AT RESTART
          $IF$ $$ALL$$  $$ALL$$  * * * F **** *** Y **
          *-------
          *        PERFORM : 'SEND .....'
          $CMD$    SEND 'ANOMALY &G1',USER=(&UID),LOGON
          $CMD$    SEND './.ANOMALY &G2',USER=(&UID),LOGON
          $EXIT$   L1EXSNAP
          ***************************************************************************
```

```
        *************************************************************************
        $IF$ $******  SCIPSR*  * R E D **** *** N **
        *-------
        *       EXEC EXIT COMPARE
        $EXIT$   L1APMCFJ
        *************************************************************************
        *************************************************************************
        *    F='L1GFIUE1', P='SCIPSR***' , ANY RECEPTION, END, FAIL TRC 8***, NO
        $IF$ L1GFIUE1 SCIPSR** * R E F 8*** *** N **
        *-------
        *       REAL CMD : 'SEND .....'
        $CMD$    SEND 'ERROR APM REC P=&PARTNID,F=&FILENAM',USER=(USR0004),LOGON
        *************************************************************************
        *************************************************************************
        *    F='FABEND', P='SCIPSR**', TOM4, TRANSMIT, END, DONE
        $IF$ FABEND   SCIPSR** 4 T E D **** *** N **
        *--------
        $CMD$    SEND 'ABEND??? P=&PARTNID,F=&FILENAM',USER=(USR0004),LOGON
        *       ACTION : ABEND IN EFFECTOR, WITH IMPLICIT NOSTAE STARTING HERE
        $ABEND$
        *************************************************************************
        *    F='RC1LOOPX', P='SCIPSR**', TOM4 T/R, START/END, DONE/FAIL
        $IF$ RC1LOOPX SCIPSR** 4 * * * **** *** Y **
        *--------
        *       ACTION : SNAP ALL ZONE (SINCE V218 PTF3: A STANDARD EXIT)
        $EXIT$   L1EXSNAP
        *       ACTION : COMMENTED WAIT 5*1MINUTE IN EFFECTOR
        *$CMD$    SEND 'WAIT &G1',USER=(USR0004),LOGON
        *$WAIT$
        *$WAIT$
        *$WAIT$
        *$WAIT$
        *$WAIT$
        *************************************************************************
        *    F='***LOOP*', P='GFI*****', TOM* R, END, DONE
        $IF$ RC1LOOP* GFI***** * R E D **** *** Y **
        *--------
        *       ACTION : EXIT FOR LOOP
        $EXIT$   L1GFIRC1
        *    F='***LOOP*', P='GFI*****', TOM* T, END, DONE
        $IF$ ***LOOP* GFI***** * T E D **** *** Y **
        *--------
        *       ACTION : DEL FILE
        *$CMD$    SEND 'DEL??? F=&FILENAM',USER=(USR0004),LOGON
        $PGMJ$   P1GFIIDC DEL &DSN
        *************************************************************************
        *************************************************************************
        *    IF SOME DUMP ...
        $IF$ GFID**** $$ALL$$  * R S * **** *** Y **
        $CMD$    SEND '*DUMP*START &G1',USER=(&L1),LOGON
        $CMD$    SEND '../.. &G2',USER=(&L1),LOGON
        $CMD$    SEND '../.. &G3',USER=(&L2),LOGON
        $CMD$    SEND '../.. &G4',USER=(&L2),LOGON
        $IF$ GFID**** $$ALL$$  * R E * **** *** Y **
        $CMD$    SEND '*DUMP*END &G1',USER=(&L1),LOGON
        *************************************************************************
        *************************************************************************
        *    IF SOME MSG  ...
        $IF$ GFIMSG** $$ALL$$  * * S * **** *** Y **
        $CMD$    SEND '*MSG* &G1',USER=(&L1),LOGON
        $CMD$    SEND '../.. &G2',USER=(&L1),LOGON
        $CMD$    SEND '../.. &G3',USER=(&L2),LOGON
        $CMD$    SEND '../.. &G4',USER=(&L2),LOGON
        *************************************************************************
```

```
*************************************************************************
* RESTRICT USER LIST (LOCAL &L2) :
$VAR$    L2 USR0004
*************************************************************************
*************************************************************************
*    IF SOME SPECIAL
$IF$ $******* $$ALL$$  * R * * **** *** N **
$CMD$    SEND '&FILENAM FROM &PARTNID ',USER=(&L2),LOGON
$PGMJ$   L1GFITS1 ULN=(RC0,&L1),MSG=&SSN&CT3&FILENAME&DSN,UER=USR0008
$EXIT$   L1EX#AE2
*************************************************************************
***************SAMPLES WITH $$API$$ DEFINED SYMBOLIC FILE-NAME**********
*************************************************************************
*    IF SOME CMD TO DO WITH &LAB (PI37 U0?FLABL), WITH SNAP
$IF$ PI37**** $$ALL$$  * R S * **** *** N **
$EXIT$   L1EXSNAP
$PGMJ$   L1GFITS1 MSG=HELLO &FILENAM,ULN=USR0009,USR0011,UER=USR0008
$CMD$    &LAB
*************************************************************************
*************************************************************************
*    IF SOME THING TO DO FOR SENDING A &LAB FILE TO &DST PARTNER
$IF$ FSEND*** $$ALL$$  4 R E D **** *** N **
*    SEND THE &LAB DSN GIVEN BY CALLER TO &DST DESTINATION
$REQ$    SSN=&SSN,SPN=&DST,SFN=FSEND,DIR=T,TYP=N,CLS=A,PRT=0,DSN=&LAB
*************************************************************************
*    IF SOME CMD TO DO WITH &LAB AND FOR ACKNOWLEDGMENT WITH &ORG
$IF$ FCMDLAB* $$ALL$$  * R E D **** *** N **
*    CALL PGM L1EXSNAP, R1 POINTS WORD POINTING 1 CHAR AFTER PGM NAME
$PGM$    L1EXSNAP &SSN&AP&EF&CT3&FILENAME&DSN...
*    CMD TO DO WITH &LAB (PI37 U0?FLABL)
$CMD$    &LAB
*    SEND BACK ACKNOWLEDGMENT BY P1B2PREQ TO &ORG SENDER
$REQ$    SSN=&SSN,SPN=&ORG,SFN=ACK,DSN=INDEX1.ACK,DIR=T,TYP=N,CLS=A,PRT
*************************************************************************
*    IF SOME REQ TO DO WITH &LAB
$IF$ FREQLAB* $$ALL$$  * R E D **** *** N **
*    REQ TO DO WITH &LAB (PI37 U0?FLABL)
$REQ$    &LAB
*************************************************************************
*    IF PREVIOUS 'ACK' FILE, PROCESS
$IF$ ACK***** $$ALL$$  * R S D **** *** N **
$CMD$    SEND 'RECEIVED &G1',USER=(USR0004),LOGON
*************************************************************************
```

```
      *************************************************************************
      ***************SAMPLES WITH $$API$$ RULE 3 DEFINED SYMBOLIC FILE-NAME***
      *************************************************************************
      *    STARTING HERE: FORCE DUMP IF ABEND FOR FOLLOWING PROCESS
      *$NOSTAE$
      *    STARTING HERE: FORCE NOSWAP APM DEFINITIVELY
      $NOSWAP$
      ******
      *    F='FROUT**', P='SCIPSR***', TOM3 RECEPTION, START, DONE
      $IF$ FROUTE** SCIPSR** 3 R S D **** *** N **
      *-------
      *       ALLOC FILE OF RULE 3 FOR RECEPTOR ONLY, BY L1APMDYA
      $DYA$    &DSN
      $EXIT$   L1EXSNAP
      ******
      *    F='FROUT**', P='SCIPSR***', TOM3 RECEPTION, END, DONE
      $IF$ FROUTE** SCIPSR** 3 R E D **** *** N **
      *-------
      *       DESALLOC FILE ALLWAYS FILE
      $DYA$
      $EXIT$   L1EXSNAP
      *    SEND THE &DSN JUST RECEIVED TO &DST DESTINATION GIVEN BY CALLER
      $REQ$    SSN=&SSN,SPN=&DST,SFN=F1,DIR=T,TYP=N,CLS=A,PRT=0,DSN=&DSN
      ******
      *    F='FROUT**', P='SCIPSR***', TOM3 RECEPTION, END, FAIL, AFTER RESTAR
      $IF$ FROUTE** SCIPSR** 3 R E F 0000 000 Y **
      *-------
      *       DESALLOC FILE ALLWAYS FILE
      $DYA$
      $EXIT$   L1EXSNAP
      *    SEND THE &DSN JUST RECEIVED TO &DST DESTINATION GIVEN BY CALLER
      $REQ$    SSN=&SSN,SPN=&DST,SFN=F1,DIR=T,TYP=N,CLS=A,PRT=0,DSN=&DSN
      ******
      *    F='FROUT**', P='SCIPSR***', TOM3 RECEPTION, END, FAIL, EVEN RESTART
      $IF$ FROUTE** SCIPSR** 3 R E F **** *** Y **
      *-------
      *       DESALLOC FILE ALLWAYS FILE, EVEN IF FAIL
      $DYA$
      $EXIT$   L1EXSNAP
      *************************************************************************
      ******
      *    F='FROUT**', P='SCIPSR***', RECEPTION, END, FAIL, AFTER RESTAR
      $IF$ FROUTE** SCIPSR** * * * * 0000 000 Y **
      $CMD$    SEND 'TEST1300 &G1',USER=(USR0009)
      $CMD$    SEND 'TEST1300 &G2',USER=(USR0009)
      $CMD$    SEND 'TEST1300 &G3',USER=(USR0009)
      $IF$ FROUTE** SCIPSR** * * * * **** 000 Y **
      $CMD$    SEND 'TEST 300 &G1',USER=(USR0009)
      $CMD$    SEND 'TEST 300 &G2',USER=(USR0009)
      $CMD$    SEND 'TEST 300 &G3',USER=(USR0009)
      *       REAL CMD : DISABLE FILE
      $JCL$    N=&DSN
      *       REAL CMD : ENABLE FILE
      $300$    S=&SSN/F=&FILENAM/E
      /*
      *************************************************************************
      *
      *        1         2         3         4         5
      *2345678901234567890123456789012345678901234567890
```

L1GFIUE1 SYSOUT File Example

Lines are truncated in this document. On the right side of each line information includes:

❖ L1GFIUE1 module version number on the first line

❖ The corresponding SYSUE1 line number

The following example results of a simulation (a transfer done with the symbolic file name L1GFIUE1) with the SYSUE1 file example from the previous page.

```
       *01201 134610 TOM4 L1GFIUE1 PESIT-E ./.                              V?R?
       *01201 134610 TOM4 L1GFIUE1        ./.                              0057
       ERR: (SIMULATE) $ERR$    SEND '&SSN L1GFIUE1 ERROR,&L,&FILENAME,&PARTNID',  0057
       SEND 'TOM4 L1GFIUE1 ERROR,0057,L1GFIUE1,SCIPSR3',USER=(USR0004)    0057
       UNUSED CHARS:  0063                                                 0057
                                                                          0062
       VAR: (SIMULATE) $VAR$    G1 >&FILENAM,&DIR: &PARTNID,&REQNUMB,&SSN,&DATE,&T  0062
                                                                          0063
       VAR: (SIMULATE) $VAR$    G2 P&PRC,T&TRC,S&SRC,&SPN,&TYP,&CT3,&RT,U=&UID      0063
                                                                          0064
       VAR: (SIMULATE) $VAR$    G3 DSN=&DSN                               0064
                                                                          0065
       VAR: (SIMULATE) $VAR$    G4 EXTDSN=&EXTDSN                         0065
                                                                          0067
       VAR: (SIMULATE) $VAR$    L1 USR0003,USR0004,USR0005,USR0008        0067
                                                                          0068
       VAR: (SIMULATE) $VAR$    L2 USR0004,USR0008                        0068
                                                                          0069
       VAR: (SIMULATE) $VAR$    L4 USR0004                                0069
                                                                          0075
       IF:  (SIMULATE) $IF$ $******* SCIPSR** * R E F 3*** *** Y **       0075
                                                                          0076
       CMD: (SIMULATE) $EXIT$   L1APMCFJ                                  0076
                                                                          0077
       IF:  (SIMULATE) $IF$ ******** SCIPSR** * R E F 3*** *** Y **       0077
                                                                          0078
       CMD: (SIMULATE) $CMD$    SEND 'ERROR &G1',USER=(&L2),LOGON         0078
       SEND 'ERROR >&FILENAM,&DIR:&PARTNID,&REQNUMB,&SSN,&DATE,&TIME',USER=(USR000  0078
       4,USR0008),LOGON                                                   0078
       UNUSED CHARS:  0035                                                0078
       SEND 'ERROR >L1GFIUE1,R:SCIPSR3,00000001,TOM4,94/07/20,13:46:10',USER=(USR0  0078
       004,USR0008),LOGON                                                 0078
       UNUSED CHARS:  0033                                                0078
                                                                          0079
       CMD: (SIMULATE) $CMD$    SEND './. &G2',USER=(&L2),LOGON           0079
       SEND './. P&PRC,T&TRC,S&SRC,&SPN,&TYP,&CT3,&RT,U=&UID',USER=(USR0004,USR000  0079
       8),LOGON                                                           0079
       UNUSED CHARS:  0043                                                0079
       SEND './. P000,T0000,S0000,5,E,TEX,N,U=NONE',USER=(USR0004,USR0008),LOGON   0079
       UNUSED CHARS:  0053                                                0079
       IF:  (SIMULATE) $IF$ FTYL    SCIPSR** 3 R E F 8*** *** Y **        0082
                                                                          0085
       CMD: (SIMULATE) $CMD$    SEND '&G1',USER=(&L4),LOGON               0085
       SEND '>&FILENAM,&DIR:&PARTNID,&REQNUMB,&SSN,&DATE,&TIME',USER=(USR0004),LOG  0085
       ON                                                                 0085
       UNUSED CHARS:  0049                                                0085
       SEND '>L1GFIUE1,R:SCIPSR3,00000001,TOM4,94/07/20,13:46:10',USER=(USR0004),L  0085
       OGON                                                               0085
       UNUSED CHARS:  0047                                                0085
```

```
        CMD: (SIMULATE) $CMD$    SEND './. LN=&LNK SP=&SPN PT=&PTY ',USER=(&L4),LO   0086
        SEND './. LN=S SP=5 PT=T ',USER=(USR0004),LOGON                            0086
        UNUSED CHARS:  0079                                                        0086
        SEND './. LN=S SP=5 PT=T ',USER=(USR0004),LOGON                            0086
        UNUSED CHARS:  0079                                                        0086
                                                                                  0087
        CMD: (SIMULATE) $CMD$    SEND './. U1=&USRVAR1 U2=&USRVAR2',USER=(&L4),LOG  0087
        SEND './. U1=A0000001 U2=H134608',USER=(USR0004),LOGON                     0087
        UNUSED CHARS:  0072                                                        0087
        SEND './. U1=A0000001 U2=H134608',USER=(USR0004),LOGON                     0087
        UNUSED CHARS:  0072                                                        0087
                                                                                  0088
        CMD: (SIMULATE) $CMD$    SEND './. &DSN &MBR FT=&FTY',USER=(&L4),LOGON      0088
        SEND './. INDEX1.L1GFIUE1.D940720.H134608.A0000001 NONE FT=S',USER=(USR000 0088
        4),LOGON                                                                   0088
        UNUSED CHARS:  0043                                                        0088
        SEND './. INDEX1.L1GFIUE1.D940720.H134608.A0000001 NONE FT=S',USER=(USR000 0088
        4),LOGON                                                                   0088
        UNUSED CHARS:  0043                                                        0088
                                                                                  0089
        CMD: (SIMULATE) $CMD$    SEND './. OR=&ORG DE=&DST',USER=(&L4),LOGON        0089
        SEND './. OR=SCIPSR3 DE=SCIPSR4',USER=(USR0004),LOGON                      0089
        UNUSED CHARS:  0073                                                        0089
        SEND './. OR=SCIPSR3 DE=SCIPSR4',USER=(USR0004),LOGON                      0089
        UNUSED CHARS:  0073                                                        0089
                                                                                  0090
        CMD: (SIMULATE) $REXX$   TYLREX F=&FILENAM P=&PARTNID RT=&RT TRC=&TRCX PRC  0090
        TYLREX F=L1GFIUE1 P=SCIPSR3 RT=N TRC=0000L PRC=000                         0090
        UNUSED CHARS:  0076                                                        0090
                                                                                  0091
        CMD: (SIMULATE) $EXIT$   L1APMCFJ                                          0091
                                                                                  0095
        CMD: (SIMULATE) $SIM$    S USRPROC F=&FILENAM,P=&PARTNID,R=&REQNUMB,D=&DSN  0095
        S USRPROC F=L1GFIUE1,P=SCIPSR3,R=00000001,D=INDEX1.L1GFIUE1.D940720.H13460 0095
        8.A0000001                                                                0095
        UNUSED CHARS:  0041                                                        0095
                                                                                  0097
        CMD: (SIMULATE) $CMD$    SEND 'FAIL &G1',USER=(USR0009),LOGON              0097
        SEND 'FAIL >&FILENAM,&DIR:&PARTNID,&REQNUMB,&SSN,&DATE,&TIME',USER=(USR0009 0097
        ),LOGON                                                                    0097
        UNUSED CHARS:  0044                                                        0097
        SEND 'FAIL >L1GFIUE1,R:SCIPSR3,00000001,TOM4,94/07/20,13:46:10',USER=(USR00 0097
        09),LOGON                                                                  0097
        UNUSED CHARS:  0042                                                        0097
                                                                                  0102
        IF:  (SIMULATE) $IF$ $******* $$ALL$$  * * * F **** *** Y **                0102
                                                                                  0107
        CMD: (SIMULATE) $CMD$    SEND 'RESTARTING &G1',USER=(USR0009),LOGON        0107
        SEND 'RESTARTING >&FILENAM,&DIR:&PARTNID,&REQNUMB,&SSN,&DATE,&TIME',USER=(P 0107
        SR0009),LOGON                                                              0107
        UNUSED CHARS:  0038                                                        0107
        SEND 'RESTARTING >L1GFIUE1,R:SCIPSR3,00000001,TOM4,94/07/20,13:46:10',USER= 0107
        (USR0009),LOGON                                                           0107
        UNUSED CHARS:  0036                                                        0107
                                                                                  0108
        CMD: (SIMULATE) $CMD$    SEND './. &G2',USER=(USR0009),LOGON               0108
        SEND './. P&PRC,T&TRC,S&SRC,&SPN,&TYP,&CT3,&RT,U=&UID',USER=(USR0009),LOGON 0108
        UNUSED CHARS:  0051                                                        0108
        SEND './. P000,T0000,S0000,5,E,TEX,N,U=NONE',USER=(USR0009),LOGON          0108
        UNUSED CHARS:  0061                                                        0108
                                                                                  0109
        CMD: (SIMULATE) $EXIT$   L1EXSNAP                                          0109
```

```
      IF:  (SIMULATE) $IF$ $$ALL$$  $$ALL$$  * * * F **** *** Y **            0111
                                                                              0114
      CMD: (SIMULATE) $CMD$    SEND 'ANOMALY &G1',USER=(&UID),LOGON           0114
      SEND 'ANOMALY >&FILENAM,&DIR:&PARTNID,&REQNUMB,&SSN,&DATE,&TIME',USER=(NONE 0114
      ),LOGON                                                                 0114
      UNUSED CHARS:  0044                                                     0114
      SEND 'ANOMALY >L1GFIUE1,R:SCIPSR3,00000001,TOM4,94/07/20,13:46:10',USER=(NO 0114
      NE),LOGON                                                               0114
      UNUSED CHARS:  0042                                                     0114
                                                                              0115
      CMD: (SIMULATE) $CMD$    SEND './.ANOMALY &G2',USER=(&UID),LOGON        0115
      SEND './.ANOMALY P&PRC,T&TRC,S&SRC,&SPN,&TYP,&CT3,&RT,U=&UID',USER=(NONE),L 0115
      OGON                                                                    0115
      UNUSED CHARS:  0047                                                     0115
      SEND './.ANOMALY P000,T0000,S0000,5,E,TEX,N,U=NONE',USER=(NONE),LOGON   0115
      UNUSED CHARS:  0057                                                     0115
                                                                              0116
      CMD: (SIMULATE) $EXIT$   L1EXSNAP                                       0116
                                                                              0119
      IF:  (SIMULATE) $IF$ $******  SCIPSR*  * R E D **** *** N **            0119
                                                                              0122
      CMD: (SIMULATE) $EXIT$   L1APMCFJ                                       0122
                                                                              0126
      IF:  (SIMULATE) $IF$ L1GFIUE1 SCIPSR** * R E F 8*** *** N **            0126
                                                                              0129
      CMD: (SIMULATE) $CMD$    SEND 'ERROR APM REC P=&PARTNID,F=&FILENAM',USER=( 0129
      SEND 'ERROR APM REC P=SCIPSR3,F=L1GFIUE1',USER=(USR0004),LOGON          0129
      UNUSED CHARS:  0064                                                     0129
                                                                              0133
      IF:  (SIMULATE) $IF$ FABEND   SCIPSR** 4 T E D **** *** N **            0133
                                                                              0135
      CMD: (SIMULATE) $CMD$    SEND 'ABEND??? P=&PARTNID,F=&FILENAM',USER=(USR00 0135
      SEND 'ABEND??? P=SCIPSR3,F=L1GFIUE1',USER=(USR0004),LOGON               0135
      UNUSED CHARS:  0069                                                     0135
                                                                              0137
      CMD: (SIMULATE) $ABEND$                                                 0137
                                                                              0141
      IF:  (SIMULATE) $IF$ RC1LOOPX SCIPSR** 4 * * * **** *** Y **            0141
                                                                              0144
      CMD: (SIMULATE) $EXIT$   L1EXSNAP                                       0144
                                                                              0154
      IF:  (SIMULATE) $IF$ RC1LOOP* GFI***** * R E D **** *** Y **            0154
                                                                              0157
      CMD: (SIMULATE) $EXIT$   L1GFIRC1                                       0157
                                                                              0159
      IF:  (SIMULATE) $IF$ ***LOOP* GFI***** * T E D **** *** Y **            0159
                                                                              0163
      CMD: (SIMULATE) $PGMJ$   P1GFIIDC DEL &DSN                             0163
      P1GFIIDC DEL INDEX1.L1GFIUE1.D940720.H134608.A0000001                  0163
      UNUSED CHARS:  0072                                                     0163
                                                                              0167
      IF:  (SIMULATE) $IF$ GFID**** $$ALL$$  * R S * **** *** Y **            0167
                                                                              0168
      CMD: (SIMULATE) $CMD$    SEND '*DUMP*START &G1',USER=(&L1),LOGON        0168
      SEND '*DUMP*START >&FILENAM,&DIR:&PARTNID,&REQNUMB,&SSN,&DATE,&TIME',USER=( 0168
      USR0003,USR0004,USR0005,USR0008),LOGON                                 0168
      UNUSED CHARS:  0013                                                     0168
      SEND '*DUMP*START >L1GFIUE1,R:SCIPSR3,00000001,TOM4,94/07/20,13:46:10',USER 0168
      =(USR0003,USR0004,USR0005,USR0008),LOGON                               0168
      UNUSED CHARS:  0011                                                     0168
```

```
        CMD: (SIMULATE) $CMD$    SEND '../..  &G2',USER=(&L1),LOGON        0169
        SEND '../..  P&PRC,T&TRC,S&SRC,&SPN,&TYP,&CT3,&RT,U=&UID',USER=(USR0003,USR 0169
        0004,USR0005,USR0008),LOGON                                       0169
        UNUSED CHARS:  0024                                               0169
        SEND '../..  P000,T0000,S0000,5,E,TEX,N,U=NONE',USER=(USR0003,USR0004,USR00 0169
        05,USR0008),LOGON                                                 0169
        UNUSED CHARS:  0034                                               0169
                                                                          0170
        CMD: (SIMULATE) $CMD$    SEND '../..  &G3',USER=(&L2),LOGON        0170
        SEND '../..  DSN=&DSN',USER=(USR0004,USR0008),LOGON               0170
        UNUSED CHARS:  0075                                               0170
        SEND '../..  DSN=INDEX1.L1GFIUE1.D940720.H134608.A0000001',USER=(USR0004,P  0170
        SR0008),LOGON                                                     0170
        UNUSED CHARS:  0038                                               0170
                                                                          0171
        CMD: (SIMULATE) $CMD$    SEND '../..  &G4',USER=(&L2),LOGON        0171
        SEND '../..  EXTDSN=&EXTDSN',USER=(USR0004,USR0008),LOGON          0171
        UNUSED CHARS:  0069                                               0171
        SEND '../..  EXTDSN=NONE',USER=(USR0004,USR0008),LOGON             0171
        UNUSED CHARS:  0072                                               0171
                                                                          0172
        IF:  (SIMULATE) $IF$ GFID**** $$ALL$$  * R E * **** *** Y **       0172
                                                                          0173
        CMD: (SIMULATE) $CMD$     SEND '*DUMP*END &G1',USER=(&L1),LOGON    0173
        SEND '*DUMP*END >&FILENAM,&DIR:&PARTNID,&REQNUMB,&SSN,&DATE,&TIME',USER=(US 0173
        R0003,USR0004,USR0005,USR0008),LOGON                              0173
        UNUSED CHARS:  0015                                               0173
        SEND '*DUMP*END >L1GFIUE1,R:SCIPSR3,00000001,TOM4,94/07/20,13:46:10',USER=( 0173
        USR0003,USR0004,USR0005,USR0008),LOGON                            0173
        UNUSED CHARS:  0013                                               0173
                                                                          0177
        IF:  (SIMULATE) $IF$ GFIMSG** $$ALL$$  * * S * **** *** Y **       0177
                                                                          0178
        CMD: (SIMULATE) $CMD$     SEND '*MSG* &G1',USER=(&L1),LOGON        0178
        SEND '*MSG* >&FILENAM,&DIR:&PARTNID,&REQNUMB,&SSN,&DATE,&TIME',USER=(USR000 0178
        3,USR0004,USR0005,USR0008),LOGON                                  0178
        UNUSED CHARS:  0019                                               0178
        SEND '*MSG* >L1GFIUE1,R:SCIPSR3,00000001,TOM4,94/07/20,13:46:10',USER=(USR0 0178
        003,USR0004,USR0005,USR0008),LOGON                                0178
        UNUSED CHARS:  0017                                               0178
                                                                          0179
        CMD: (SIMULATE) $CMD$     SEND '../..  &G2',USER=(&L1),LOGON       0179
        SEND '../..  P&PRC,T&TRC,S&SRC,&SPN,&TYP,&CT3,&RT,U=&UID',USER=(USR0003,USR0 0179
        004,USR0005,USR0008),LOGON                                        0179
        UNUSED CHARS:  0025                                               0179
        SEND '../..  P000,T0000,S0000,5,E,TEX,N,U=NONE',USER=(USR0003,USR0004,USR000 0179
        5,USR0008),LOGON                                                  0179
        UNUSED CHARS:  0035                                               0179
                                                                          0180
        CMD: (SIMULATE) $CMD$     SEND '../..  &G3',USER=(&L2),LOGON       0180
        SEND '../..  DSN=&DSN',USER=(USR0004,USR0008),LOGON               0180
        UNUSED CHARS:  0076                                               0180
        SEND '../..  DSN=INDEX1.L1GFIUE1.D940720.H134608.A0000001',USER=(USR0004,USR 0180
        0008),LOGON                                                       0180
        UNUSED CHARS:  0039                                               0180
                                                                          0181
        CMD: (SIMULATE) $CMD$     SEND '../..  &G4',USER=(&L2),LOGON       0181
        SEND '../..  EXTDSN=&EXTDSN',USER=(USR0004,USR0008),LOGON          0181
        UNUSED CHARS:  0070                                               0181
        SEND '../..  EXTDSN=NONE',USER=(USR0004,USR0008),LOGON             0181
        UNUSED CHARS:  0073                                               0181
```

```
       VAR: (SIMULATE) $VAR$     L2 USR0004                                0185
                                                                          0189
       IF:  (SIMULATE) $IF$ $******* $$ALL$$  * R * * **** *** N **        0189
                                                                          0190
       CMD: (SIMULATE) $CMD$    SEND '&FILENAM FROM &PARTNID ',USER=(&L2),LOGON  0190
       SEND 'L1GFIUE1 FROM SCIPSR3 ',USER=(USR0004),LOGON                 0190
       UNUSED CHARS:  0076                                                0190
       SEND 'L1GFIUE1 FROM SCIPSR3 ',USER=(USR0004),LOGON                 0190
       UNUSED CHARS:  0076                                                0190
                                                                          0191
       CMD: (SIMULATE) $PGMJ$   L1GFITS1 ULN=(RC0,&L1),MSG=&SSN&CT3&FILENAME&DSN,  0191
       L1GFITS1 ULN=(RC0,USR0003,USR0004,USR0005,USR0008),MSG=TOM4TEXL1GFIUE1INDEX 0191
       1.L1GFIUE1.D940720.H134608.A0000001,UER=USR0008                    0191
       UNUSED CHARS:  0003                                                0191
       L1GFITS1 ULN=(RC0,USR0003,USR0004,USR0005,USR0008),MSG=TOM4TEXL1GFIUE1INDEX 0191
       1.L1GFIUE1.D940720.H134608.A0000001,UER=USR0008                    0191
       UNUSED CHARS:  0003                                                0191
                                                                          0192
       CMD: (SIMULATE) $EXIT$   L1EX#AE2                                  0192
                                                                          0197
       IF:  (SIMULATE) $IF$ PI37**** $$ALL$$  * R S * **** *** N **        0197
                                                                          0198
       CMD: (SIMULATE) $EXIT$   L1EXSNAP                                  0198
                                                                          0199
       CMD: (SIMULATE) $PGMJ$   L1GFITS1 MSG=HELLO &FILENAM,ULN=USR0009,USR0011,U 0199
       L1GFITS1 MSG=HELLO L1GFIUE1,ULN=USR0009,USR0011,UER=USR0008         0199
       UNUSED CHARS:  0067                                                0199
                                                                          0200
       CMD: (SIMULATE) $CMD$    &LAB                                      0200
       INDEX1.PS.F080.EMPTY                                               0200
       UNUSED CHARS:  0105                                                0200
       INDEX1.PS.F080.EMPTY                                               0200
       UNUSED CHARS:  0105                                                0200
                                                                          0204
       IF:  (SIMULATE) $IF$ FSEND*** $$ALL$$  4 R E D **** *** N **        0204
                                                                          0206
       CMD: (SIMULATE) $REQ$    SSN=&SSN,SPN=&DST,SFN=FSEND,DIR=T,TYP=N,CLS=A,PRT 0206
       SSN=TOM4,SPN=SCIPSR4,SFN=FSEND,DIR=T,TYP=N,CLS=A,PRT=0,DSN=INDEX1.PS.F080.  0206
       EMPTY                                                              0206
       UNUSED CHARS:  0046                                                0206
       SSN=TOM4,SPN=SCIPSR4,SFN=FSEND,DIR=T,TYP=N,CLS=A,PRT=0,DSN=INDEX1.PS.F080.  0206
       EMPTY                                                              0206
       UNUSED CHARS:  0046                                                0206
                                                                          0209
       IF:  (SIMULATE) $IF$ FCMDLAB* $$ALL$$  * R E D **** *** N **        0209
                                                                          0211
       CMD: (SIMULATE) $PGM$    L1EXSNAP &SSN&AP&EF&CT3&FILENAME&DSN...    0211
       L1EXSNAP TOM40301TEXL1GFIUE1INDEX1.L1GFIUE1.D940720.H134608.A0000001...  0211
       UNUSED CHARS:  0054                                                0211
                                                                          0213
       CMD: (SIMULATE) $CMD$    &LAB                                      0213
       INDEX1.PS.F080.EMPTY                                               0213
       UNUSED CHARS:  0105                                                0213
       INDEX1.PS.F080.EMPTY                                               0213
       UNUSED CHARS:  0105                                                0213
                                                                          0215
       CMD: (SIMULATE) $REQ$    SSN=&SSN,SPN=&ORG,SFN=ACK,DSN=INDEX1.ACK,DIR=T,T 0215
       SSN=TOM4,SPN=SCIPSR3,SFN=ACK,DSN=INDEX1.ACK,DIR=T,TYP=N,CLS=A,PRT  0215
       UNUSED CHARS:  0060                                                0215
```

```
     IF:  (SIMULATE) $IF$ FREQLAB* $$ALL$$  * R E D **** *** N **              0218
                                                                               0220
     CMD: (SIMULATE) $REQ$    &LAB                                             0220
     INDEX1.PS.F080.EMPTY                                                      0220
     UNUSED CHARS:  0105                                                       0220
     INDEX1.PS.F080.EMPTY                                                      0220
     UNUSED CHARS:  0105                                                       0220
                                                                               0223
     IF:  (SIMULATE) $IF$ ACK***** $$ALL$$  * R S D **** *** N **              0223
                                                                               0224
     CMD: (SIMULATE) $CMD$    SEND 'RECEIVED &G1',USER=(USR0004),LOGON         0224
     SEND 'RECEIVED >&FILENAM,&DIR:&PARTNID,&REQNUMB,&SSN,&DATE,&TIME',USER=(USR 0224
     0004),LOGON                                                              0224
     UNUSED CHARS:  0040                                                       0224
     SEND 'RECEIVED >L1GFIUE1,R:SCIPSR3,00000001,TOM4,94/07/20,13:46:10',USER=(U 0224
     SR0004),LOGON                                                            0224
     UNUSED CHARS:  0038                                                       0224
                                                                               0232
     NSW: (SIMULATE) $NOSWAP$                                                  0232
                                                                               0235
     IF:  (SIMULATE) $IF$ FROUTE** SCIPSR** 3 R S D **** *** N **              0235
                                                                               0238
     CMD: (SIMULATE) $DYA$    &DSN                                             0238
     INDEX1.L1GFIUE1.D940720.H134608.A0000001                                 0238
     UNUSED CHARS:  0085                                                       0238
                                                                               0239
     CMD: (SIMULATE) $EXIT$   L1EXSNAP                                         0239
                                                                               0242
     IF:  (SIMULATE) $IF$ FROUTE** SCIPSR** 3 R E D **** *** N **              0242
                                                                               0245
     CMD: (SIMULATE) $DYA$                                                     0245
                                                                               0246
     CMD: (SIMULATE) $EXIT$   L1EXSNAP                                         0246
                                                                               0248
     CMD: (SIMULATE) $REQ$    SSN=&SSN,SPN=&DST,SFN=F1,DIR=T,TYP=N,CLS=A,PRT=0, 0248
     SSN=TOM4,SPN=SCIPSR4,SFN=F1,DIR=T,TYP=N,CLS=A,PRT=0,DSN=INDEX1.L1GFIUE1.D9 0248
     40720.H134608.A0000001                                                   0248
     UNUSED CHARS:  0029                                                       0248
                                                                               0251
     IF:  (SIMULATE) $IF$ FROUTE** SCIPSR** 3 R E F 0000 000 Y **              0251
                                                                               0254
     CMD: (SIMULATE) $DYA$                                                     0254
                                                                               0255
     CMD: (SIMULATE) $EXIT$   L1EXSNAP                                         0255
                                                                               0257
     CMD: (SIMULATE) $REQ$    SSN=&SSN,SPN=&DST,SFN=F1,DIR=T,TYP=N,CLS=A,PRT=0, 0257
     SSN=TOM4,SPN=SCIPSR4,SFN=F1,DIR=T,TYP=N,CLS=A,PRT=0,DSN=INDEX1.L1GFIUE1.D9 0257
     40720.H134608.A0000001                                                   0257
     UNUSED CHARS:  0029                                                       0257
                                                                               0260
     IF:  (SIMULATE) $IF$ FROUTE** SCIPSR** 3 R E F **** *** Y **              0260
                                                                               0263
     CMD: (SIMULATE) $DYA$                                                     0263
                                                                               0264
     CMD: (SIMULATE) $EXIT$   L1EXSNAP                                         0264
                                                                               0268
     IF:  (SIMULATE) $IF$ FROUTE** SCIPSR** * * * * 0000 000 Y **              0268
```

```
                                                                          0269
       CMD: (SIMULATE) $CMD$     SEND 'TEST1300 &G1',USER=(USR0009)        0269
       SEND 'TEST1300 >&FILENAM,&DIR:&PARTNID,&REQNUMB,&SSN,&DATE,&TIME',USER=(USR 0269
       0009)                                                              0269
       UNUSED CHARS:  0046                                                0269
       SEND 'TEST1300 >L1GFIUE1,R:SCIPSR3,00000001,TOM4,94/07/20,13:46:10',USER=(U 0269
       SR0009)                                                            0269
       UNUSED CHARS:  0044                                                0269
       CMD: (SIMULATE) $CMD$     SEND 'TEST1300 &G2',USER=(USR0009)        0270
       SEND 'TEST1300 P&PRC,T&TRC,S&SRC,&SPN,&TYP,&CT3,&RT,U=&UID',USER=(USR0009)  0270
       UNUSED CHARS:  0052                                                0270
       SEND 'TEST1300 P000,T0000,S0000,5,E,TEX,N,U=NONE',USER=(USR0009)    0270
       UNUSED CHARS:  0062                                                0270
                                                                          0271
       CMD: (SIMULATE) $CMD$     SEND 'TEST1300 &G3',USER=(USR0009)        0271
       SEND 'TEST1300 DSN=&DSN',USER=(USR0009)                            0271
       UNUSED CHARS:  0087                                                0271
       SEND 'TEST1300 DSN=INDEX1.L1GFIUE1.D940720.H134608.A0000001',USER=(USR0009  0271
       )                                                                  0271
       UNUSED CHARS:  0050                                                0271
                                                                          0272
       IF:  (SIMULATE) $IF$ FROUTE** SCIPSR** * * * * **** 000 Y **        0272
                                                                          0273
       CMD: (SIMULATE) $CMD$     SEND 'TEST 300 &G1',USER=(USR0009)        0273
       SEND 'TEST 300 >&FILENAM,&DIR:&PARTNID,&REQNUMB,&SSN,&DATE,&TIME',USER=(USR 0273
       0009)                                                              0273
       UNUSED CHARS:  0046                                                0273
       SEND 'TEST 300 >L1GFIUE1,R:SCIPSR3,00000001,TOM4,94/07/20,13:46:10',USER=(U 0273
       SR0009)                                                            0273
       UNUSED CHARS:  0044                                                0273
                                                                          0274
       CMD: (SIMULATE) $CMD$     SEND 'TEST 300 &G2',USER=(USR0009)        0274
       SEND 'TEST 300 P&PRC,T&TRC,S&SRC,&SPN,&TYP,&CT3,&RT,U=&UID',USER=(USR0009)  0274
       UNUSED CHARS:  0052                                                0274
       SEND 'TEST 300 P000,T0000,S0000,5,E,TEX,N,U=NONE',USER=(USR0009)    0274
       UNUSED CHARS:  0062                                                0274
                                                                          0275
       CMD: (SIMULATE) $CMD$     SEND 'TEST 300 &G3',USER=(USR0009)        0275
       SEND 'TEST 300 DSN=&DSN',USER=(USR0009)                            0275
       UNUSED CHARS:  0087                                                0275
       SEND 'TEST 300 DSN=INDEX1.L1GFIUE1.D940720.H134608.A0000001',USER=(USR0009  0275
       )                                                                  0275
       UNUSED CHARS:  0050                                                0275
                                                                          0277
       CMD: (SIMULATE) $JCL$     N=&DSN                                   0277
       N=INDEX1.L1GFIUE1.D940720.H134608.A0000001                        0277
       UNUSED CHARS:  0083                                                0277
                                                                          0279
       CMD: (SIMULATE) $300$     S=&SSN/F=&FILENAM/E                      0279
       S=TOM4/F=L1GFIUE1/E                                                0279
       UNUSED CHARS:  0107                                                0279
                                                                          0280
          (SIMULATE) EOF                                                  0280
                                                                          0289
       RET0 RC=??  SIMULATE ALL          00000000                         V.R.
                                                                          V.R.
       FLG:                                                               
       TOM40301TEXI94/07/2013:46:10L1GFIUE1SCIPSR3 INDEX1.L1GFIUE1.D940720.H13460  V.R.
       RE0S
       INT/EXT ID  00000001  01801636  0000/2116 0000/2199                V.R.
```

Index