Connect:Express® OS/390

Administration Guide

Version 4.2.0

Sterling Commerce
An IBM Company

Connect:Express OS/390 Adminstration Guide
Version 4.2.0
First Edition

Contents

Appendix B Error Codes and Messages

Appendix C Monitor Commands

Appendix D Initialization Parameters

Appendix E Definition of VTAM Resources

Appendix F JCL Files for the Connect:Express Monitor

Appendix G Using Protocol Traces

Index

Preface

The *Connect:Express OS/390 Administration Guide* is for programmers and network operations staff who install and maintain Connect:Express OS/390 4.2.0.

This guide assumes knowledge of the OS/390 operating system, including its applications, network, and environment. If you are not familiar with the OS/390 operating system, refer to the OS/390 library of manuals.

Chapter Overview

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

Chapter/Appendix	Description
Chapter 1 Sysplex Environment	This chapter describes Connect:Express support for the OS/390 sysplex environment and includes procedures to implement a Connect:Express/Plex environment and Extended Recovery.
Chapter 2 User Exits	This chapter describes the User Exit interface which enables you to start actions throughout the transfer process.
Chapter 3 Controlling Transfer Operations	This chapter describes how Connect:Express processes transfer requests and how you can control transfer operations using automation tools, scheduling features, resources, and security options.
Chapter 4 Incoming and Outgoing Transfers	This chapter describes the file transfer operations of Connect:Express. It covers PeSIT, Odette, and ETEBAC transfers that are processed in the TOM, ANM, and APM address spaces. This chapter outlines the steps of an incoming and an outgoing file transfer and identifies the parameters, the transfer process, and reporting capabilities.
Chapter 5 PeSIT and ODETTE-FTP Protocols	This chapter discusses the PeSIT and ODETTE-FTP protocols. It describes how each parameter is used, how files are exchanged with Partners, and how information is verified. The relationship between internal events and external events is also discussed.
Chapter 6 Troubleshooting	This chapter describes common problems that you may encounter, tools that you can use to identify the problem, and corrective action that you can take. It also includes information about enhancing system performance and getting help from Technical Support.
Appendix A User Libraries	This appendix describes the Connect:Express user libraries which provide examples to help you customize and automate Connect:Express.
Appendix B Error Codes and Messages	This appendix describes error codes and messages from the Connect:Express application.

Chapter/Appendix	Description
Appendix C Monitor Commands	This appendix lists and describes monitor commands that you can use to activate or deactivate resources.
Appendix D Initialization Parameters	This appendix identifies the SYSIN parameters for Connect:Express and describes the parameters that are required for Connect:Express to work as a stand-alone, a Plex manager, or a Plex server, and the optional parameters that you can use to increase resource productivity and to take advantage of advanced features.
Appendix E Definition of VTAM Resources	This appendix provides definitions of the VTAM resources for Connect:Express. It includes definitions of an application major node, a logmode, a switched major node for X.25, an interpret table for transparent PAD, and an X25NPSI definition for X.25 links.
Appendix F JCL Files for the Monitor	This appendix describes the DD Names used by Connect:Express in different address spaces.

Connect:Express Documentation

Connect:Express documentation consists of the following manuals:

- The *Connect:Express OS/390 4.2.0 Release Notes* lists maintenance updates and any important notes.
- The *Connect:Express OS/390 4.2.0 Installation Guide* describes the planning and installation of Connect:Express.
- The *Connect:Express OS/390 4.2.0 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 4.2.0 Utilities Guide* describes the optional Utilities package that you can integrate with Connect:Express.
- The *Connect:Express OS/390 4.2.0 FTP Guide* provides you with the information that you need to use Connect:Express with the FTP protocol.
- The *Connect:Express OS/390 4.2.0 Administration Guide* provides detailed information about transfer operations for system administrators and other advanced users of Connect:Express.
- The *Connect:Express OS/390 4.2.0 Options Guide* provides information about the CICS, IMS, and RJE interfaces available for Connect:Express.
- The *Connect:Express OS/390 4.2.0 PeSIT User Fields Guide* describes how you can exchange the PeSIT Pi37 and Pi99 fields with any PeSIT software.
- The *Connect:Express OS/390 4.2.0 Etebac3 User Guide* provides you with the information that you need to use Connect:Express with the Etebac3 protocol.
- The *Connect:Express HTTP Option Implementation Guide* provides you with the information that you need to implement HTTP access to Connect:Express OS/390 repository.
- The *Connect:Express version 4.2.0 SSL Guide* includes general information on implementing secured file transfers.
- The *Connect:Express version 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 Administration 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.
Underlined Letters	Underlining indicates default values for parameters and subparameters. For example, PLEX=Y|N 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

Sysplex Environment

This chapter describes Connect:Express support for the OS/390 sysplex environment and includes procedures to implement a Connect:Express/Plex environment and Extended Recovery. Refer to the *Connect:Express OS/390 Sysplex Supervision Guide* for further information.

Overview

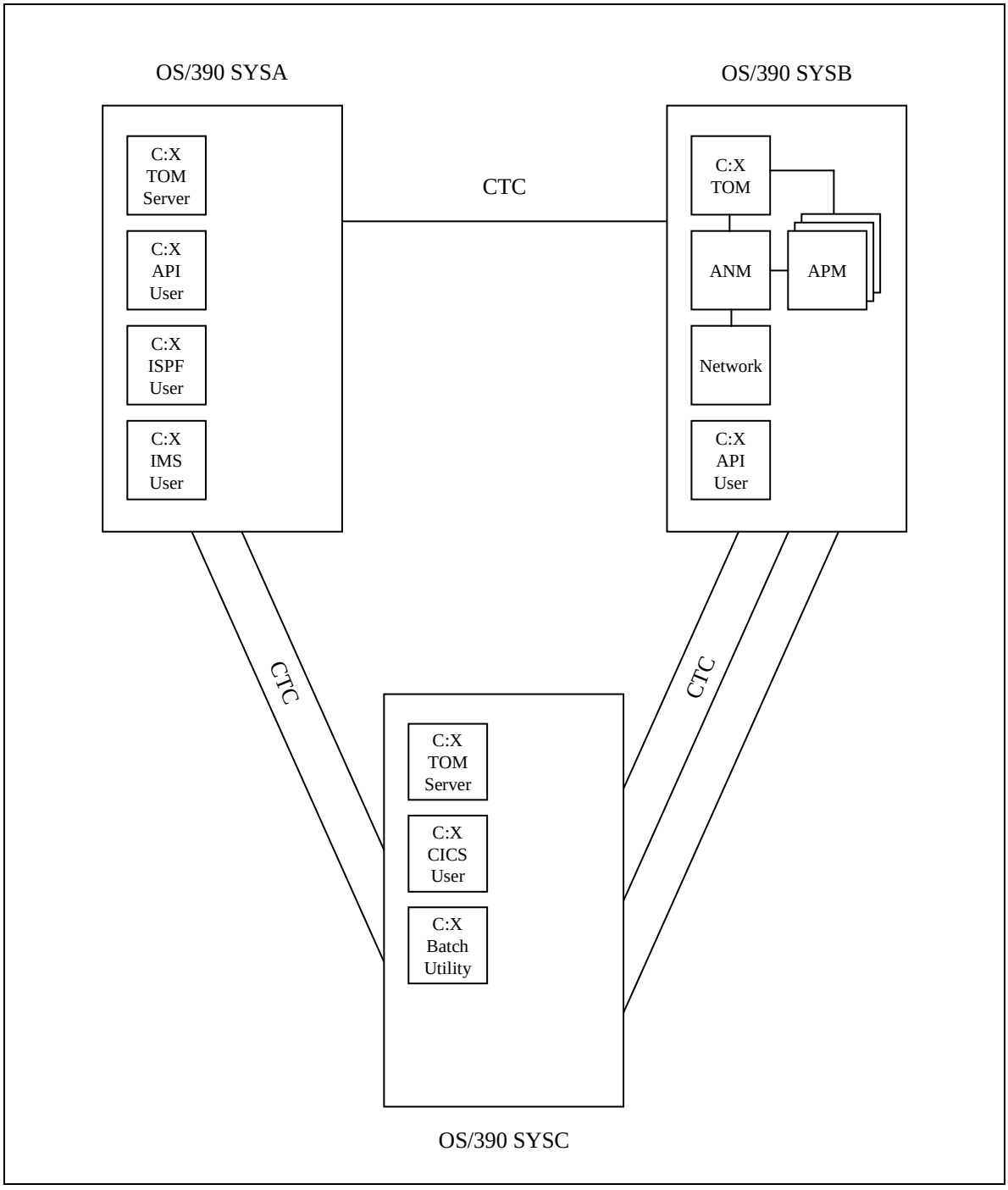
Connect:Express now provides full support for the OS/390 sysplex environment with the release of Connect:Express OS/390. For example, users are no longer required to run all Connect:Express MVS applications, batch utilities, and online connections to Connect:Express within the same OS/390 image that runs Connect:Express. Now, Connect:Express OS/390 initializes a manager/server environment. This environment is called a Connect:Express/Plex. On one OS/390 image, the Connect:Express/Plex Manager and its associated address spaces like the APM, ANM and AFM, are started. The Connect:Express/Plex Manager is the system that actually performs the file transfers and is known as the TOM Manager. During initialization, the TOM Manager issues commands to start the Connect:Express/Plex Servers (TOM Servers) in other OS/390 images of the sysplex. This manager/server environment enables the Connect:Express subsystem to be established on all OS/390 images and lets any user interact transparently with the TOM Manager.

Connect:Express OS/390 also provides support for the OS/390 Extended Recovery Facility, so you do not need to use other automation products or setup manual procedures to take over if Connect:Express fails. When Connect:Express is not configured as a Connect:Express/Plex, it is called a Connect:Express stand-alone.

Note: Before using any of the sample code in this section, you should have a running, stand-alone Connect:Express Version 4.2 system.

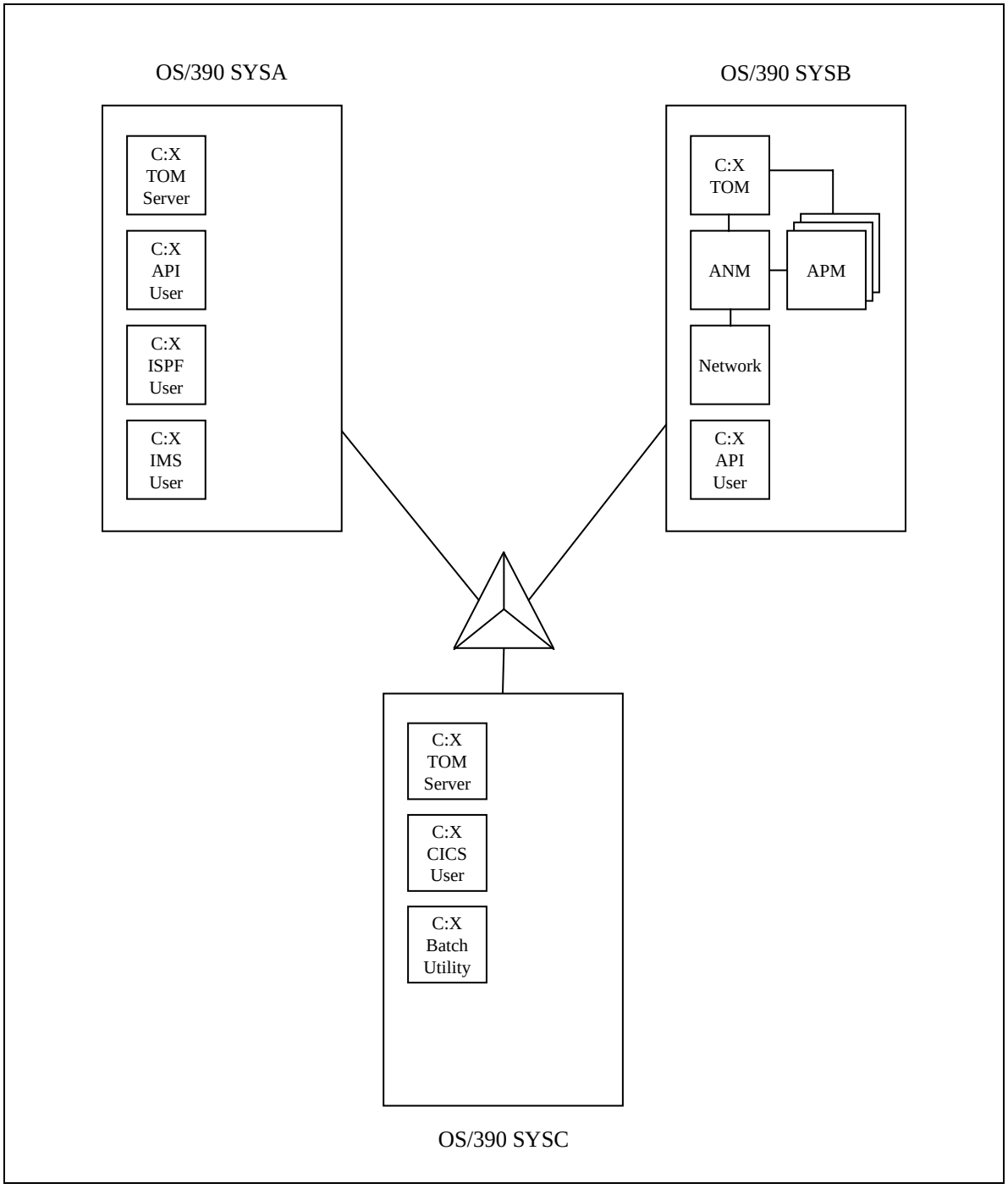
Connect:Express in a Sysplex Environment

The figure below shows how Connect:Express operates in a Sysplex environment. In this example, the system is running on SYSB.



Connect:Express in a Parallel Sysplex Environment

The figure below shows how Connect:Express operates in a parallel Sysplex environment. In this example, the system is running on SYSB.



Note: Both sysplex and parallel sysplex environments are supported. IBM defines a sysplex as a collection of OS/390 images that share work, and a parallel sysplex as a sysplex that includes a Coupling Facility.

Connect:Express OS/390 makes use of the OS/390 facility XCF to communicate between the Connect:Express TOM Manager and its Connect:Express TOM Servers.

Sysplex Initialization Keywords

The Connect:Express OS/390 initialization process includes new parameter keywords that support features like single-system image in a Sysplex or parallel Sysplex environment and Extended Recovery. The new parameter keywords are listed in the table below:

Keyword	Description
PLEX=N|Y	Specifies whether the Connect:Express system will run in a Connect:Express/Plex environment. PLEX=Y indicates that the Connect:Express/Plex environment is used and that the Connect:Express initialization will open and read the CXPLEX data set to retrieve the local initialization parameters that apply to this member of the Connect:Express/Plex. The CXPLEX DD and SYSPRTX DD are required for PLEX=Y. PLEX=N indicates that Connect:Express will run as a stand-alone.
XCFGROUP=ssn| $$ssn$$|name	Indicates the XCF group name for the Connect:Express/Plex or Extended Recovery environment. ssn is the subsystem name for this Connect:Express system. If PLEX=Y, then XCFGROUP=ssn If PLEX=N, then XCFGROUP=$$ssn$$ (used for Extended Recovery) The XCF group name must be unique within the OS/390 sysplex. XCF group names that begin with A through I or SYS are reserved by IBM. Typically you do not need to specify the XCFGROUP= keyword.
XRF=N|Y	Indicates if Extended Recovery is enabled or not. Extended Recovery enables a partially initialized Connect:Express stand-alone, Plex Manager, or Plex Server to watch over the active Connect:Express member. If the active Connect:Express member fails, the partially initialized one takes over. The CXPLEX DD and SYSPRTX DD are required for XRF=Y. This specifies the standby TOM procedure that starts during initialization.

Sysplex Local Initialization Data Set

A new initialization data set called CXPLEX has been added to the Connect:Express initialization process. This parameter is used by the Connect:Express/Plex and Extended Recovery environments. The CXPLEX data set contains information about how the member behaves in a Connect:Express/Plex environment, as well as information about Extended Recovery. This data set is referenced by the CXPLEX DD card in the initialization JCL. The initialization keywords are described below:

Keyword	Description
MGRTYP=NO|YES	Specifies the manager type for the member. A member can be a Connect:Express/Plex Manager or Server. When PLEX=Y, it must be the first keyword in the CXPLEX data set. MGRTYP=YES, the member is a Connect:Express/Plex Manager. MGRTYP=NO, the member is a Connect:Express/Plex Server.
XCFTIM=3|nn	Specifies (in minutes) the time-out value for XCF communications. This is only used when PLEX=Y. Time-outs that occur when a Connect:Express/Plex Manager is attempting to communicate with a Connect:Express/Plex Server causes the Connect:Express/Plex Manager to terminate the Connect:Express/Plex Server. Time-outs that occur when a Connect:Express/Plex Server is attempting to communicate with a Connect:Express/Plex Manager cause a WTOR to be issued. The operator can respond with RETRY or CANCEL to retry the communication or to cancel the Connect:Express/Plex Server. The default is 3.

Keyword	Description
SERVER=(sys-name, proc-name)	This is only used by the Connect:Express/Plex manager (MGRTYP=YES) in a Connect:Express/Plex environment. This field specifies the system name to which a START command is routed by the manager to start the server. For example, if SERVER=(SYSA,CXTOMSA), the following command is issued to OS/390: RO SYSA,S CXTOMSA. Multiple SERVER= keywords can be specified, up to the limit indicated in the Connect:Express Asset Protection file (NUMBER-SERVERS). The upper limit is 32. If MGRTYPE=NO, this keyword is ignored.
XRFPRC=(sys-name, proc-name)	Indicates the Extended Recovery environment system name to which a START command is routed to start its procedure name. This is only used when XRF=Y or Extended Recovery is enabled. The started member will recognize that it is in an Extended Recovery environment and that the active member is already up. It will monitor the health of the active member, and wait to take over its work if the active member fails. For example, if XRFPRC=(SYSC,CXTOMXC), the following command would be issued to OS/390: RO SYSC,S CXTOMXC. Extended Recovery is supported in both a PLEX=Y and PLEX=N environment. If XRF=N, this keyword is ignored.

Note: A SYSOUT file referenced by the SYSPRTX DD card in the initialization JCL contains a summary of CXPLEX cards that were processed.

Implementing a Connect:Express/Plex Environment

To implement a Connect:Express/Plex environment, you must change your stand-alone system into a Connect:Express/Plex by creating CXPLEX files and start procedures for the manager and the servers. The procedures are based on the Connect:Express stand-alone procedure. The manager procedure has not changed, but the server procedures must be adapted because most of the files are not used by the server. You do not need to change the SYSIN file because the server only uses certain parameters of the file that apply to it. Examples are provided in the *PARMLIB* and the *INSTLIB* libraries.

In the procedure below, CXTOM is the name that starts this Connect:Express system. This example assumes that you want to enable applications to use Connect:Express whether they are running in SYSA, SYSB, or SYSC.

Note: The same OS/390 subsystem must be defined in SYSA, SYSB, and SYSC, and will be used by the Connect:Express/Plex.

Complete the following steps to change your stand-alone Connect:Express system into a Connect:Express/Plex:

1. Create a member (MANAGER) in the Connect:Express index1.index2.PARMLIB and include the following statements.

```
MGRTYP=YES
SERVER=(SYSA,CXTOMSA)
SERVER=(SYSC,CXTOMSC)
```

2. Create a member (SERVERA) in the same PDS and include the following statement.

```
MGRTYP=NO
```

3. Create a member (SERVERC) in the same PDS and include the following statement.

```
MGRTYP=NO
```

4. Ensure that the CXTOM procedure includes the following keywords in the PROC prototype and that these new keywords are passed to the PGM=P1B2P000 step.

```
//TOM4200 PROC ...,PLEX=N,XRF=N,XCFGROUP=' '
  :
//TOM4200 EXEC PGM=P1B2P000,
// PARM='RUN=&RUN,SSN=&SSN,XRF=&XRF,PLEX=&PLEX,XCFGROUP=&XCFGROUP',
```

5. Add a DD statement to the CXTOM procedure step that executes PGM=P1B2P000 and points to the MANAGER member, as shown below.

```
//CXPLEX  DD DISP=SHR,DSN=index1.index2.PARMLIB(MANAGER)
//SYSPRTX DD SYSOUT=&OUT
```

6. Copy the CXTOM procedure and name the copy CXTOMSA.
7. In CXTOMSA, remove all steps prior to the step executing PGM=P1B2P000, then change the following DD cards to those listed below.

```
//SYSCHK  DD DUMMY
//SYSCHK2 DD DUMMY
//SYSANM  DD DUMMY
//SYSJCL  DD DUMMY
//SYSJNL  DD DUMMY
//SYSEVT  DD DUMMY
//SYSPRM  DD DUMMY
//SYSFIL  DD DUMMY
//SYSPAR  DD DUMMY
//SYSSNA  DD DUMMY
//SYSX25  DD DUMMY
//SYSTCP  DD DUMMY
//ET5PID  DD DUMMY
//ET5FID  DD DUMMY
//DIFJNL  DD DUMMY
//DIFMBX  DD DUMMY
//RSAPARM DD DUMMY
```

8. Change the CXPLEX DD to point to the SERVERA member.

```
//CXPLEX DD DISP=SHR,DSN=index1.index2.PARMLIB(SERVERA)
```

9. Copy the SYSRCY data set used by CXTOM and name it appropriately for the CXTOMSA procedure. Then change the SYSRCY DD in CXTOMSA to point to this copied SYSRCY data set.

```
//SYSRCY DD …
```

10. If the SYSLOG file is not a SYSOUT data set, format a new log file and change the SYSLOG DD in CXTOMSA to point to this new log data set. You can format the SYSLOG file by referring to step 13 of the installation procedure. The SYSLOG should be a SYSOUT data set.

```
//SYSLOG DD SYSOUT=&OUT
```

11. Copy the CXTOMSA member and name this copy CXTOMSC.
12. Change the CXPLEX DD to point to the SERVERC member.

```
//CXPLEX DD DISP=SHR,DSN=index1.index2.PARMLIB(SERVERC)
```

13. Copy the SYSRCY data set used by CXTOM and name it appropriately for the CXTOMSC procedure. Then change the SYSRCY DD in CXTOMSC to point to this copied SYSRCY data set.

```
//SYSRCY DD …
```

14. If the SYSLOG file is not a SYSOUT data set, format a new log file and change the SYSLOG DD in CXTOMSC to point to this new log data set. You can format the SYSLOG file by referring to step 13 of the installation procedure. The SYSLOG should be a SYSOUT data set.

```
//SYSLOG DD SYSOUT=&OUT
```

Starting a Connect:Express/Plex

After you convert your stand-alone system to a Connect:Express/Plex, start your Connect:Express system on SYSB as usual, but specify PLEX=Y, as shown below.

```
RO SYSB,S CXTOM,SSN=TOM4,PLEX=Y
```

Connect:Express recognizes the PLEX=Y parameter and opens the CXPLEX data set during initialization. The CXPLEX data set indicates that it is a Connect:Express/Plex Manager and then starts its Connect:Express/Plex Servers and related address spaces. To start the Servers, commands are issued to OS/390 that route START commands to SYSA and SYSC. The Connect:Express/Plex Manager sends the PLEX=Y, XRF=, and XCFGROUP= parameters to the Connect:Express/Plex Servers, and then the Connect:Express/Plex Manager initializes XCF communications.

When the Connect:Express/Plex Servers are started, they read the PLEX=Y parameter and open the CXPLEX data set. This data set tells Connect:Express to be a Connect:Express/Plex Server and not to start any other address space. The Connect:Express/Plex Server initializes XCF communications and interfaces with the Connect:Express/Plex Manager. Any application requesting Connect:Express facilities can run in the same Connect:Express/Plex Server OS/390 image. The application has the same capabilities as if it were running on the Connect:Express/Plex Manager’s OS/390 image.

Extended Recovery

Connect:Express OS/390 version 4.2 supports Extended Recovery in both stand-alone and Plexed environments. You can implement Extended Recovery with the same OS/390 image or with different OS/390 images. When using the same image, the recovery is processed if Connect:Express stops. When you implement Extended Recovery with different OS/390 images, the recovery is processed if Connect:Express stops, or if the OS/390 image on which Connect:Express is running stops.

Implementing Extended Recovery with the Same OS/390 Image

You can implement Extended Recovery with the same OS/390 image. In the procedure below, the name to start this Connect:Express system is CXTOM. This example assumes that you want your standby TOM to run on SYSB.

Complete the following steps to convert your stand-alone Connect:Express system to a Connect:Express system with Extended Recovery in the same OS/390 image:

1. Create a member (CXXRF1) in index1.index2.PARMLIB and include the following statements.

```
XRFPRC=(SYSB,CXTOM1)
```

2. Create another member (CXXRF2) in the same PDS and include the following statement.

```
XRFPRC=(SYSB,CXTOM1)
```

3. Ensure that the CXTOM procedure includes the following keywords in the PROC prototype and that these new keywords are sent to the step executing PGM=P1B2P000.

```
//TOM4200 PROC ...,PLEX=N,XRF=N,XCFGROUP=' '
  :
//TOM4200 EXEC PGM=P1B2P000,
// PARM='RUN=&RUN,SSN=&SSN,XRF=&XRF,PLEX=&PLEX,XCFGROUP=&XCFGROUP',
```

4. Add a DD statement to the CXTOM procedure step that executes the step running PGM=P1B2P000. PGM=P1B2P000 points to the CXXRF1 member created in step 3.

```
//CXPLEX  DD DISP=SHR,DSN=index1.index2.PARMLIB(CXXRF1)
//SYSPRTX DD SYSOUT=&OUT
```

5. Change the DISP= in the SYSCHK and SYSRCY DDs to DISP=SHR. This change is required to enable the stand-alone member to have the two data sets allocated, but not opened.

```
//SYSCHK DD …,DISP=SHR
//SYSRCY DD …,DISP=SHR
```

6. Copy the CXTOM procedure and name the copy CXTOM1. Delete all steps before the step executing PGM=P1B2P000.
7. Change the CXPLEX DD to reference the member CXXRF2.

```
//CXPLEX DD DISP=SHR,DSN=index1.index2.PARMLIB(CXXRF2)
```

Starting Connect:Express with Extended Recovery in the Same OS/390 Image

After you implement Extended Recovery with the same OS/390 image, start your Connect:Express system on SYSB as usual, but specify XRF=Y, as shown below.

```
RO SYSB,S CXTOM,SSN=TOM4,XRF=Y
```

Connect:Express recognizes the XRF=Y parameter and opens the CXPLEX data set during initialization. Connect:Express also checks to see if this member is the active or standby member by looking in the XCF group.

- If there already is an active member, then this is the standby TOM for the active member. If the active member fails, this standby TOM completes its initialization and then takes control.
- If there is no active member, then this Connect:Express TOM becomes the active member and continues initializing. After initialization, it issues a START command to its standby TOM, which is indicated by the XRFPRC= keyword in the CXPLEX data set.

If the CXTOM address space fails, the CXTOM1 address space recognizes this failure, continues its initialization, and takes over the work. Then, it issues a START command to its standby TOM as specified by the XRFPRC= keyword in the CXPLEX data set. This starts the CXTOM1 procedure, and the standby TOM takes over.

The CXTOM procedure can include preliminary steps like off loading the recovery file and the checkpoint file or controlling the journal file. These steps do not have to be repeated in the extended recovery process once Connect:Express is running. During extended recovery, the CXTOM1 procedure starts with no preliminary steps. If another failure occurs, extended recovery starts the CXTOM1 procedure on the next system, and the process is repeated.

Note: Normal termination of the active member terminates the standby TOM with RC=0.

Implementing Extended Recovery with Different OS/390 Images

Connect:Express OS/390 version 4.2 supports Extended Recovery in both stand-alone and Plexed environments, and you can implement Extended Recovery with different OS/390 images. In the procedure below, the name that starts this Connect:Express system is CXTOM. This example assumes that you want your standby TOM for SYSB to run on SYSA. If SYSA takes over for SYSB, your standby TOM for SYSA runs on SYSC, and the standby TOM for SYSC runs on SYSB.

Note: If using SNA connections, define the VTAM APPLs in all OS/390 images as DYNAMIC. If using TCP/IP, define the IP address as a VIPA address so it can be moved with the Connect:Express application.

Complete the following steps to convert your stand-alone Connect:Express system to a Connect:Express system with Extended Recovery on different OS/390 images:

1. Create a member (CXXRFA) in index1.index2.PARMLIB and include the following statements.

```
XRFPRC=(SYSA,CXTOMA)
```

2. Create another member (CXXRFB) in the same PDS and include the following statement.

```
XRFPRC=(SYSB,CXTOMB)
```

3. Create another member (CXXRFC) in the same PDS and include the following statement.

```
XRFPRC=(SYSC,CXTOMC)
```

4. Ensure that the CXTOM procedure includes the following keywords in the PROC prototype and that these new keywords are sent to the step executing PGM=P1B2P000.

```
//TOM4200 PROC ...,PLEX=N,XRF=N,XCFGROUP=' '
  :
//TOM4200 EXEC PGM=P1B2P000,
// PARM='RUN=&RUN,SSN=&SSN,XRF=&XRF,PLEX=&PLEX,XCFGROUP=&XCFGROUP',
```

5. Add a DD statement to the CXTOM procedure that runs the step executing PGM=P1B2P000. PGM=P1B2P000 points to the CXXRFA member created in step 4.

```
//CXPLEX DD DISP=SHR,DSN=index1.index2.PARMLIB(CXXRFA)
//SYSPRTX DD SYSOUT=&OUT
```

6. Change the DISP= in the SYSCHK and SYSRCY DDs to DISP=SHR. This change is required to enable the stand-alone member to have two data sets allocated, but not opened.

```
//SYSCHK DD …,DISP=SHR
//SYSRCY DD …,DISP=SHR
```

7. Copy the CXTOM procedure and name the copy CXTOMA. Delete all steps before the step executing PGM=P1B2P000.
8. Change the CXPLEX DD to reference the member CXXRFC.

```
//CXPLEX DD DISP=SHR,DSN=index1.index2.PARMLIB(CXXRFC)
```

9. Copy the CXTOMA procedure and name the copy CXTOMC.
10. Change the CXPLEX DD to reference the member CXXRFB.

```
//CXPLEX DD DISP=SHR,DSN=index1.index2.PARMLIB(CXXRFB)
```

11. Copy the CXTOMC procedure and name the copy CXTOMB.
12. Change the CXPLEX DD to reference the member CXXRFA.

```
//CXPLEX DD DISP=SHR,DSN=index1.index2.PARMLIB(CXXRFA)
```

Starting Connect:Express with Extended Recovery in Different OS/390 Images

After you implement Extended Recovery with different OS/390 images, you should start your Connect:Express system on SYSB as usual, but specify XRF=Y.

```
RO SYSB,S CXTOM,SSN=TOM4,XRF=Y
```

Connect:Express recognizes the XRF=Y parameter and opens the CXPLEX data set during initialization. Connect:Express also checks if this member is the active or standby member by looking in the XCF group.

- If there already is an active member, then this is the standby TOM for the active member. If the active member fails, this standby TOM completes its initialization and then takes control.
- If there is no active member, then this Connect:Express TOM becomes the active member and continues initializing. After initialization, it issues a START command to its standby TOM, which is specified by the XRFPRC= keyword in the CXPLEX data set.

If the CXTOM address space fails, the CXTOMA address space on SYSA recognizes the failure, continues its initialization, and takes over the work. Then, it issues a START command to its standby TOM as specified by the XRFPRC= keyword in the CXPLEX data set.

The execution of CXTOM (on SYSB), CXTOMA (on SYSA), CXTOMC (on SYSC), CXTOMB (on SYSB), and CXTOMA (on SYSA), lets the CXTOM procedure include preliminary steps. This is not repeated once Connect:Express is running. During extend recovery, the CXTOMA (on SYSA) procedure starts. If another failure occurs, Extended Recovery starts the CXTOMC (on SYSC) procedure, and the process continues.

Note: Normal termination of the active member terminates the standby TOM with a RC=0.

Implementing both Connect:Express/Plex and Extended Recovery

The Connect:Express/Plex environment presents a single system image to Connect:Express applications/APIs from all OS/390 images within the sysplex. It is critical that each member of the Connect:Express/Plex is available to present this single system image. You can use a combination of Extended Recovery and the Connect:Express/Plex Manager/Server environment to ensure this availability.

Note: If using SNA connections, define the VTAM APPLs in all OS/390 images as DYNAMIC. If using TCP/IP, define the IP address as a VIPA address so it can be moved with the Connect:Express application.

The same OS/390 subsystem must be defined on SYSA, SYSB, and SYSC, and is used by the Connect:Express/Plex. Complete the following steps to convert your stand-alone Connect:Express system to a Connect:Express/Plex with Extended Recovery:

1. Create a member (MANAGERB) in the Connect:Express index1.index2.PARMLIB and include the following statements.

```
MGRTYP=YES
SERVER=(SYSA,CXTOMSA)
SERVER=(SYSC,CXTOMSC)
XRFPRC=(SYSA,CXTOMA)
```

2. Create a member (MANAGERA) in the Connect:Express index1.index2.PARMLIB and include the following statements.

```
MGRTYP=YES
SERVER=(SYSB,CXTOMSB)
SERVER=(SYSC,CXTOMSC)
XRFPRC=(SYSC,CXTOMC)
```

3. Create a member (MANAGERC) in the Connect:Express index1.index2.PARMLIB and include the following statements.

```
MGRTYP=YES
SERVER=(SYSA,CXTOMSA)
SERVER=(SYSB,CXTOMSB)
XRFPRC=(SYSB,CXTOMB)
```

4. Create a member (SERVERA) in the same PDS and include the following statements.

```
MGRTYP=NO
XRFPRC=(,CXTOMSA)
```

5. Create a member (SERVERB) in the same PDS and include the following statements.

```
MGRTYP=NO
XRFPRC=(,CXTOMSB)
```

6. Create a member (SERVERC) in the same PDS and include the following statements.

```
MGRTYP=NO
XRFPRC=(,CXTOMSC)
```

7. Ensure that the CXTOM procedure includes the following keywords in the PROC prototype and that these new keywords are sent to the PGM=P1B2P000 step.

```
//TOM4200 PROC ...,PLEX=N,XRF=N,XCFGROUP=' '
  :
//TOM4200 EXEC PGM=P1B2P000,
// PARM='RUN=&RUN,SSN=&SSN,XRF=&XRF,PLEX=&PLEX,XCFGROUP=&XCFGROUP',
```

8. Add a DD statement to the CXTOM procedure step that executes PGM=P1B2P000 and points to the MANAGERB member.

```
//CXPLEX  DD DISP=SHR,DSN=index1.index2.PARMLIB(MANAGERB)
//SYSPRTX DD SYSOUT=&OUT
```

9. Copy the CXTOM procedure and name the copy CXTOMA.
10. Change the CXPLEX DD to point to the MANAGERA member.

```
//CXPLEX DD DISP=SHR,DSN=index1.index2.PARMLIB(MANAGERA)
```

11. Copy the CXTOM procedure and name the copy CXTOMB.
12. Change the CXPLEX DD to point to the MANAGERB member.

```
//CXPLEX DD DISP=SHR,DSN=index1.index2.PARMLIB(MANAGERB)
```

13. Copy the CXTOM procedure and name the copy CXTOMC.

14. Change the CXPLEX DD to point to the MANAGERA member.

```
//CXPLEX DD DISP=SHR,DSN=index1.index2.PARMLIB(MANAGERC)
```

15. Copy the CXTOM procedure and name the copy CXTOMSA.
16. In CXTOMSA, delete all steps before the step executing PGM=P1B2P000. Then change the following DD cards to those shown.

```
//SYSCHK  DD DUMMY
//SYSCHK2 DD DUMMY
//SYSANM  DD DUMMY
//SYSJCL  DD DUMMY
//SYSJNL  DD DUMMY
//SYSEVT  DD DUMMY
//SYSPRM  DD DUMMY
//SYSFIL  DD DUMMY
//SYSPAR  DD DUMMY
//SYSSNA  DD DUMMY
//SYSX25  DD DUMMY
//SYSTCP  DD DUMMY
//ET5PID  DD DUMMY
//ET5FID  DD DUMMY
//DIFJNL  DD DUMMY
//DIFMBX  DD DUMMY
//RSAPARM DD DUMMY
```

17. Change the CXPLEX DD to point to the SERVERA member.

```
//CXPLEX DD DISP=SHR,DSN=index1.index2.PARMLIB(SERVERA)
```

18. Copy the SYSRCY data set used by CXTOM and name it appropriately for the CXTOMSA procedure. Change the SYSRCY DD in CXTOMSA to point to this copied SYSRCY data set.

```
//SYSRCY DD …
```

19. If the SYSLOG file is not a SYSOUT data set, format a new log file and change the SYSLOG DD in CXTOMSA to point to this new log data set. Ensure that SYSLOG is a SYSOUT data set.

```
//SYSLOG DD SYSOUT=&OUT
```

20. Copy the CXTOMSA member and name this copy CXTOMSB.
21. Change the CXPLEX DD to point to the SERVERB member.

```
//CXPLEX DD DISP=SHR,DSN=index1.index2.PARMLIB(SERVERB)
```

22. Copy the SYSRCY data set used by CXTOM and name it appropriately for the CXTOMSB procedure. Change the SYSRCY DD in CXTOMSB to point to this copied SYSRCY data set.

```
//SYSRCY DD …
```

23. If the SYSLOG file is not a SYSOUT data set, format a new log file and change the SYSLOG DD in CXTOMSA to point to this new log data set. Ensure that SYSLOG is a SYSOUT data set.

```
//SYSLOG DD SYSOUT=&OUT
```

24. Copy the CXTOMSB member and name this copy CXTOMSC.
25. Change the CXPLEX DD to point to the SERVERC member.

```
//CXPLEX DD DISP=SHR,DSN=index1.index2.PARMLIB(SERVERC)
```

26. Copy the SYSRCY data set used by CXTOM and name it appropriately for the CXTOMSC procedure. Change the SYSRCY DD in CXTOMSC to point to this copied SYSRCY data set.

```
//SYSRCY DD …
```

27. If the SYSLOG file is not a SYSOUT data set, format a new log file and change the SYSLOG DD in CXTOMSA to point to this new log data set. Ensure that SYSLOG is a SYSOUT data set.

```
//SYSLOG DD SYSOUT=&OUT
```

Starting Connect:Express/Plex with Extended Recovery

After you implement both Extended Recovery and Connect:Express/Plex, start your Connect:Express system on SYSB as usual, but specify PLEX=Y and XRF=Y, as shown below.

```
RO SYSB,S CXTOM,SSN=TOM4,PLEX=Y,XRF=Y
```

Connect:Express recognizes the PLEX=Y and XRF=Y parameters and opens the CXPLEX data set during initialization. The CXPLEX data set indicates that it is a Connect:Express/Plex Manager and starts its Connect:Express/Plex Servers and related address spaces. To start the servers, commands are issued to OS/390 that send START commands to SYSA and SYSC. Then, the Connect:Express/Plex Manager sends the PLEX=Y, XRF=Y, and XCFGROUP= parameters to the Connect:Express/Plex Servers. The Connect:Express standby Server is started on SYSA, and then the Connect:Express/Plex Manager initializes XCF communications.

When the Connect:Express/Plex Servers are started, they read the PLEX=Y and XRF=Y parameters and open the CXPLEX data set, which tells Connect:Express to be a Connect:Express/Plex Server and not to start any other address space. The Connect:Express/Plex Server initializes XCF communications and interfaces with the Connect:Express/Plex Manager. Any application requesting Connect:Express facilities runs in the same Connect:Express/Plex Server OS/390 image. The application has the same capabilities as if it were running on the Connect:Express/Plex Manager's OS/390 image. The Connect:Express/Plex Server issues the command to start its standby Server which is also running on the Plex Manager's OS/390 image.

If a Connect:Express/Server fails, the standby server takes over immediately. The standby Server then joins the XCF group with the Connect:Express/Manager and starts its own standby Server on the same OS/390 image.

If the Connect:Express/Plex Manager fails, the Connect:Express/Plex Servers and their standby Servers terminate automatically along with other address spaces started by the Connect:Express/Plex Manager. Then, the standby Connect:Express/Plex Manager on SYSA takes over immediately and starts its address spaces which are the Connect:Express/Plex Servers on SYSB and SYSC. Then, the manager starts its standby Connect:Express Server on SYSC.

Note: Normal shutdown of the Connect:Express/Plex Manager will cause the Connect:Express/Plex Servers and all standby Connect:Express Servers in the same XCF group to terminate with RC=0.

Chapter 2

User Exits

This chapter describes the User Exit interface which enables you to start actions throughout the transfer process.

Overview

Connect:Express provides all the functionality to execute transfers beginning with requesting a transfer to reporting a successful end of transfer. The transfer operation process is outlined in the table below:

Operation Type	Description
Outgoing request	Local request of transfer that you send to Connect:Express
Connection	Session establishment between computers and applications
Selection	Transfer injection between applications
Security	Identification and authorization controls
Transfer initialization	Allocation of storage resources
Record processing	Presentation of data
Transfer termination	Deallocation of storage resources
Journal	End of transfer reporting

User exits result in actions and can be started at any time during the transfer process. For example, you can start an action during the monitor initialization or termination.

Some standard exits are provided as load modules or source examples. The following table identifies standard user exits and the library where you can find them.

Name	Library	Description
EX£*	SAMPLIB	Examples of user exits with a description header. Assembly or Cobol with compile – link edit examples.

Name	Library	Description
EX£APM050 EX£APM051 EX£APM052	SAMPLIB	The data translation facility uses the record processing interface. Source modules are provided for you to customize the translation tables if a specific translation table is to be used with one partner.
M1USRCNA	MACLIB	The RACF connection user exit that you can customize.
EX£USRCN	SAMPLIB	An example of how to use the M1USRCNA macro.
L1USRCNA	SAMPLIB	A connection user exit is provided that performs RACROUTE calls.
L1USRLOG	SAMPLIB	An example of how to call the L1B2LOG program.
L1EXSNAP	LOADLIB	A trace module is provided that can be set into normal exit definition fields or link-edited with user exits. It dumps exit communication areas at entry of user exits (connection, selection, transfer initialization, record processing, security and transfer termination), provided that a SNAPDUMP file is allocated to Connect:Express and/or the APM.
L1EX£AE2	LOADLIB	Trace modules are provided that can be set into normal exit definition fields. It prints exit communication areas at the entry of the user exits interface, during transfer initialization and transfer termination. It also dynamically allocates SYSOUT 'EX£????' in the APM.

User Exits

A user exit typically corresponds to one phase of the transfer process such as initialization, termination, or record processing. User exits can be:

- **Used specifically for one transfer**. It executes a function associated with the transfer of one file, like allocation using a transfer initialization exit. This exit is loaded dynamically when needed, then deleted. It is defined in the file attributes.
- **Shared by several transfers to provide a general service**, like data translation using a record processing exit. This exit is loaded during initialization and defined in a presentation table or in the SYSIN file.
- **Called during the different phases of the transfer**. This user exit is called a server exit because a server is loaded during initialization. The connection server exit, defined in the T1B2PCNT table, and the transfer server exit, defined in the T1B2PSRT, are examples of server exits.

Note: Standard calls and server type calls can be mixed for a single transfer. See *Mixing Server Exits and Standard User Exits* on page 2-20.

User exits are loaded dynamically during initialization or during transfer execution depending on whether they are dedicated to one transfer or are common to many. Depending on the exit type, user exits are loaded by the TOM address space or by the APM address space from their SYSLIB. The TOM address space calls initialization, termination, connection, and journal exits. The APM address space calls initialization, termination, selection, security, transfer, and record processing exits. The AFM address space calls initialization, termination and connection exits and the EAS address space calls initialization, termination, selection, transfer, and record processing exits. With FTP transfers, the SYSLIB is not needed.

Server exits **must** be reentrant, while record processing exits and beginning and end of transfer exits **should** be reentrant. Connect:Express supports both reentrant and non-reentrant exits. User exits should be assembly modules, however COBOL and PL1 modules are also supported.

Connect:Express protects against user exits that take too much time to execute using timers. A timer is set when branching to an exit, and when the time expires the task is stopped. The following standard IBM link conventions are used:

- Register 14 is used as a return address for the caller.
- Register 15 is used for the address of the exit (at entry) and contains the return code.
- Register 1 contains the address of the parameter list.
- Each parameter of the list is the address of one field or a communication structure.
- The parameter is a list of addresses of parameter fields.
- The user return code in register 15 must be less than or equal to 90. If greater than 90, then the 91 return code is issued. User return codes are listed in Appendix B.

Typically, an exit receives one communication structure input with a parameter list made up of one parameter that has the address of the communication structure. The following standard Connect:Express communication structures are provided in the *MACLIB*:

Module	Prefix/Field	Description
D1B2PCNX	CNX*	Connection interface, TOM and AFM initialization, TOM and AFM termination
D1B2RUEX	UEX*	Selection, transfer initialization, and termination, APM and EAS initialization, APM and EAS termination.
D1B2RPEX	PEX*	Record processing interface
D1B2PJNL	Z45*	Journal interface
D1B2PRAC	XS*	RACF security interface

Connect:Express executes symmetrical open/close calls to user exits. This means that after a user exit has executed successfully, the symmetrical call is activated. For example, if a user exit is called during Connect:Express initialization and has executed successfully, it will be called again when Connect:Express terminates. A user exit can also be called during initialization, termination, and at each occurrence of the process for which it is responsible. This is important when setting up user exits because it means that a user exit can be called several times. Some exits are disabled when their execution fails, others make the process stop if the code they return is not OK. The following list provides examples:

Exit	Example
Connection exit	Is normally called during Connect:Express initialization, at each connection to or from a Partner, at each end of connection and during Connect:Express termination. The connection exit can reject the connection. The end of connection call is executed only if the connection call was successful. The connection exit is disabled if the initialization call is unsuccessful.
Selection exit	Is normally called during Connect:Express initialization, at each transfer request to or from a partner, at each end of transfer and during Connect:Express termination. The selection exit can reject the transfer request. The end of transfer call is executed only if the transfer request call was successful. The selection exit is disabled if the initialization call is unsuccessful.
Transfer initialization exit	Is called once, but it is related to the transfer termination exit which is called only if the transfer initialization phase was successfully executed. The transfer initialization exit can reject the transfer. NOTE: You can run a transfer initialization exit and no transfer termination exit.
Transfer termination exit	Is called once, but only if the transfer initialization phase was successfully executed. It can be called even if no transfer initialization exit was defined. The transfer termination exit can reject a transfer on the receiver side. NOTE: You can run a transfer termination exit and no transfer initialization exit.

Exit	Example
Record processing exit	Is called during the open file phase, during each record process and during the close file phase. The transfer record processing exit can interrupt the transfer. If the open phase call is unsuccessful, the transfer is interrupted, and no other call is executed. If the open call was successful and the transfer is interrupted, the close call is executed.
Journal exit	Is normally called during Connect:Express initialization, at each end of transfer, and during Connect:Express termination. The journal exit is disabled if the execution is unsuccessful, at any time.

The table below shows the D1B2PCNX, D1B2RUEX, D1B2RPEX and D1B2PJNL call and status field values. The D1B2PRAC structure corresponds to a fixed call type.

Structure	Call Type field Value	Description	Status Field Value
D1B2PJNL	Z45KINTR ”INIT”	Initialization of Connect:Express, TOM Address Space	N/A
D1B2PJNL	Z45KINTR “TERM”	Termination of Connect:Express, TOM Address Space	N/A
D1B2PCNX	CNXCALTY “OPC”	Open (Initialization) TOM Address Space	N/A
D1B2PCNX	CNXCALTY “CLC”	Close (termination) TOM Address Space	N/A
D1B2RUEX	UEXCALTY “OPS”	Open (Initialization) APM Address Space	N/A
D1B2RUEX	UEXCALTY “CLS”	Close (termination) APM Address Space	N/A
D1B2PCNX	CNXCALTY “CNX”	Connection, input or output : During an input call the exit is invoked before the monitor. During an outgoing call the exit is invoked before sending the connection request to the remote	CNXSTATS “C” – Call “E” – End “F” – Failed
D1B2RUEX	UEXCALTY “SEL”	Selection, input or output : During an input call, the exit is invoked after receiving the transfer request from the remote partner and before forwarding it to the monitor. During an outgoing call, the exit is invoked before sending the transfer request to the remote partner.	UEXSTATS “A” – Call “E” – End “F” – Failed “S” – System abend
D1B2RUEX	UEXCALTY “IEX””	Transfer initialization, input or output, read or write: The exit is invoked during the file allocation phase.	UEXSTATS “A”– Call to application “I” – Initialization (allocation)
D1B2RUEX	UEXCALTY “SEC””	Security	N/A
D1B2RPEX	PEXCALTY “PEX”	Record processing, input or output , read or write: The exit is invoked during the file open and close phases. When receiving it is invoked before writing each record. When sending it is invoked after reading each record.	PEXSTATS “B” – Begin (open) “M” – Middle (record) “E” – End (close)

Structure	Call Type field Value	Description	Status Field Value
D1B2RUEX	UEXCALTY “TEX”	Transfer termination, input or output, read or write. The exit is invoked during the file deallocation phase.	UEXSTATS “E” – End “F” – Failed “S” – System Abend

User Exit Conventions

The table below lists each step in the transfer process during which you can call a user exit. The table identifies the parameter where you can declare the exit, the structure that is used as the parmlist, and the address space that runs the exit. It also shows if the exit is loaded during initialization or loaded when it is needed. When an exit is pre-loaded, this means that you will have to stop and start the address space that is in charge of executing this exit each time you update it.

Operation	Parameter Field	Dsect Call Type/Status	Description	ASID
Connect:Express initialization	SYSIN file - UEXJNL= Or SYSINEXT if UEXJNL=L1B2PDIX (journal exits driver)	D1B2PJNL "INIT"	Only header is passed input, R15 output	TOM (pre-loaded)
Connect:Express initialization	T1B2PCNT TSO/ISPF 3.3.CNT	D1B2PCNX "OPC"	Only header is passed input, R15 output	TOM (pre-loaded)
Connect:Express termination	SYSIN file - UEXJNL= or SYSINEXT if UEXJNL=L1B2PDIX	D1B2PJNL "TERM"	Only header is passed input, R15 output	TOM (pre-loaded)
Connect:Express termination	T1B2PCNT TSO/ISPF 3.3.CNT	D1B2PCNX "CLC"	Only header is passed input, R15 output	TOM/AFM (pre-loaded)
APM/EAS initialization	T1APMSRT TSO/ISPF 3.3.SRT	D1B2RUEX "OPS"	Only header is passed input, R15 output	APM/EAS (pre-loaded)
APM/EAS termination	T1APMSRT TSO/ISPF 3.3.SRT	D1B2RUEX "CLS"	Only header is passed input, R15 output	APM/EAS (pre-loaded)
REQUEST outgoing	Module L1B2P006	D1B2PREQ	Local control of user requests to Connect:Express by L1B2P006 module.	TOM
CONNECTION	T1B2PCNT TSO/ISPF 3.3.CNT	D1B2PCNX "CNX"/"C" "E" "F"	Incoming or outgoing connections during open and close session.	TOM/AFM (pre-loaded)
SELECTION	T1APMSRT TSO/ISPF 3.3.SRT	D1B2RUEX "SEL"/"A"		APM/EAS (pre-loaded)
INITIALIZATION of TRANSFER	SYSIN file File directory	D1B2RUEX "IEX"/"I"		APM/EAS (load+delete)
TERMINATION of TRANSFER	SYSIN file File directory	D1B2RUEX "IEX"/"E" "F" "S"		APM/EAS (load+delete)

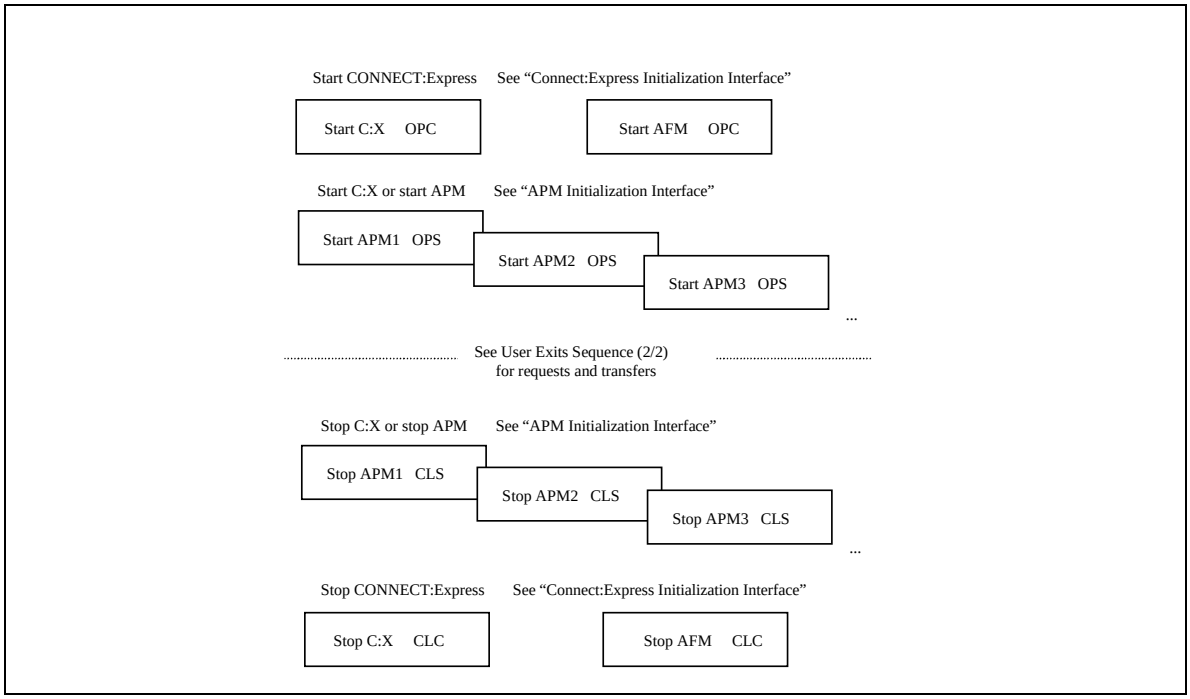
Operation	Parameter Field	Dsect Call Type/Status	Description	ASID
SECURITY RACF	Connect:Express LOADLIB L1B2PTAC	D1B2PRAC	Incoming transfers during selection of transfer.	TOM
RECORD PROCESSING	T1B2PPxx TSO/ISPF 3.3 Pxx	D1B2RPEX "PEX"/"B" "M" "E"		APM/EAS (pre-loaded)
READ-WRITE exit	T1B2PPxx TSO/ISPF 3.3 Pxx	D1B2RUEX	Incoming and outgoing transfers, all I/O functions: Open, Close, Note, Point, Read, Write	APM/EAS (preloaded)
END SELECTION	T1APMSRT TSO/ISPF 3.3.SRT	D1B2RUEX "SEL"/"E/F/S"		APM/EAS (pre-loaded)
JOURNAL	SYSIN file - UEXJNL= Or SYSINEXT if UEXJNL=L1B2PDIX (journal exits driver)	D1B2PJNL	Incoming started transfers and outgoing transfers during termination of transfer, successful or not.	TOM (pre-loaded)

User Exit Sequence

The following diagram illustrates the sequence of user exit execution from initialization to termination of Connect:Express. During Connect:Express initialization and termination, and during APM and EAS initializations and terminations, user exits can be invoked in the TOM ASID, the AFM ASID, and in each APM ASID. For example, you could have any of the following exits:

- Journal exit in TOM ASID
- Connection exit in TOM ASID, if defined in the T1B2PCNT table
- Selection exit in APM ASID, if defined in the T1APMSRT table

Note: Each time an EAS is started or terminated, the selection exit is invoked for initialization or termination. An EAS is started only when needed, as opposed to the APM which is started for performing future transfers.



The table below shows how calls to user exits correspond to the successive protocol steps and to Connect:Express SYSLOG reports. For example, when the CONNECTION OPENED message is issued in the SYSLOG file, the user connection exit has already been invoked with a 'C' call operation. See “Connection Interface” on page 2-16 for more information.

SYSLOG Message	Operation	Reference (see also)
(outgoing request) REQUEST ACCEPTED/REJECTED	REQUEST	"Outgoing Request Control"
(protocol) CONNECTION OPENED	Connection CNX,C	"Connection Interface"
(protocol)	Selection SEL,A	"Selection Interface"
(protocol)	Security	"Security Interface"
(protocol) TRANSFER ACCEPTED	TR.INIT IEX,I	"Transfer Initialization Interface"
(protocol)	Security SEC	"RSA/DES Interface"
(protocol)	I/O UEX, open, point	"Read Write Exit"
(protocol) TRANSFER STARTED	TRANSFER PEX,B	"Record Processing Interface"
(protocol)	TRANSFER PEX,M	"Record Processing Interface"
(protocol)	I/O UEX, read ,write, point, note	"Read Write Exit"
(protocol)	SECURITY SEC	"RSA/DES Interface” protocol
(protocol)	TRANSFER PEX,E	"Record Processing Interface"
(protocol)	I/O UEX, close	"Read Write Exit"

SYSLOG Message	Operation	Reference (see also)
(protocol)	TR.TERM. TEX,E	"Transfer Termination Interface"
(protocol)	SELECTION SEL,E	"Selection Interface"
TRANSFER ENDED/REJECTED/INTERRUPTED	JOURNAL	"Journal Interface"
(protocol) CONNECTON CLOSED/REJECTED	Connection CNX,E	"Connection Interface"

Applications of User Exits

You can create user exits to control transfers with an application, maintain control of file allocation, send sections of a file, and select records for transfer. Examples are provided in the *SAMPLIB*.

Note: A re-compile and re-linkedit is necessary when upgrading from previous versions because exits are an extension of Connect:Express.

Controlling Transfers with an Application

Some environments or transfer protocols require specific identification that Connect:Express is not able to process. This can happen when communications are between hosts from different companies. You can use connection and selection exits to control incoming connections and check for network identification, application partner identification, and application file identification. You can also use these exits to control outgoing connections and check for application partner and file identification after Connect:Express initializes.

All decisions about accepting a connection and a transfer are taken by the application using the D1B2PCNX and D1B2RUEX parameter lists. Using (CNX, SEL, IEX, TEX, PEX) exits enables applications to let Connect:Express only control session and protocol management.

Using File Allocation Rule 3

You can maintain control of file allocation using allocation rule 3 and beginning and end of transfer exits. The exit will get default allocation parameters from the file directory using the D1B2RUEX structure. It then determines the file data set name and the current allocation parameters to process. You can allocate the file and return the data set name or ask Connect:Express to allocate using D1B2RUEX allocation fields.

If you do the file allocation, you must also deallocate the files. Using **allocation rule A** works the same way. IEX and TEX calls are processed by the server exit driver L1APMSRV.

Sending Sections of a File

You can use beginning of transfer and record processing exits to send a section of a file, rather than the entire file. To do this, you will need to store logical files in a single data set.

The user exit tells Connect:Express to point to one block of the data set before the beginning of transfer, using the R15 (1004) return code from the IEX call to the exit and the POINTLIST field of the D1B2RUEX structure. Then, the exit tells Connect:Express that the end of file will be detected by the exit, using the R15 (2002) return code from the record processing user exit during transfer (first call). If Connect:Express detects the end of file before the record processing user exit, it is considered an I/O error. Finally, the exit detects the logical end of file using the R15 (2004) return code from the record processing user exit during transfer.

Selecting Records to Transfer

You can use a record processing user exit during transfer to select records that you do not want to send or write in a file. During transfer, Connect:Express will send or write all of the remaining records in the file.

User Access Control

A connection exit enables you to control user access to the monitor. For example, you can enable TSO users under the control of RACF to access the monitor. The exit authenticates the user and determines the symbolic partner name that the monitor should use to begin the session. The user name is considered the origin of the request and does not need to be defined in the Partners directory.

Connect:Express OS/390 provides a connection exit called L1USRCNA. You can enable this exit by adding the definition to the Connection exits table (TSO/ISPF 3.3.CNT) for the PeSIT and FTP protocols. A source file for the exit is provided in the *SAMPLIB*.

Connect:Express OS/390 also provides a macro called M1USRCNA that enables you to customize the L1USRCNA module without changing the source code. You must compile this macro with the appropriate parameters. The macro M1USRCNA is provided in the *MACLIB*. The example that shows you how to use this macro is called EX£USRCN, and provided in the *SAMPLIB*.

The following section shows how L1USRCNA works if you do not change it. All examples use default symbolic partner names and passwords ($PESITD, $PESITE and $FTP). See *Customizing User Access Control with M1USRCNA* on page 2-11 for a description of the macro parameters.

Messages

The following messages are sent when the monitor starts and stops, if the option INIT. is selected in the activation parameters for the exit. (TSO/ISPF 3.3.CNT)

```
TOM4CNA NOTICE   L1USRCNA INIT
TOM4CNA NOTICE   L1USRCNA TERM
```

Rejected Message

The following message is sent when USER is rejected.

```
TOM4CNA WARNING  $PESITE$ USER   127.000.000.001   CALL     REJECTED 00000008
```

This message is preceded by a security interface message similar to the following:

```
ICH408I USER(USER) GROUP(USR£USR) NAME(USR M.BOBBY      )  162
  LOGON/JOB INITIATION - INVALID PASSWORD
IRR013I  VERIFICATION FAILED. INVALID PASSWORD GIVEN.
```

This message is also followed by the monitor message shown below:

```
TOM4076I L1USRCNA CONNECTION EXIT ERR. CNX RC:0036
```

Accepted Message

The following message is sent when USER is accepted to access the monitor.

```
TOM4CNA NOTICE   $PESITE$ USER  127.000.000.001   CALL     ACCEPTED
```

Using the Exit with the PeSIT Protocol

Connect:Express handles a PeSIT connection based on the preconnection message. A connection exit is only called after the fpdu Connect is received. The connection exit must be accepted before the exit can control the user identification process.

The symbolic partner name, $PESIT$, was added to the Partners directory to enable Connect:Express to accept PeSIT connections before the user identification controls are processed. When the symbolic name $PESIT$ is defined in the directory, all calls from an unknown partner using the PeSIT protocol are accepted with this symbolic Partner name. This feature is necessary to implement user access control, but it can also be used if you only want to accept anonymous sessions.

Note: The keyword $PESIT$ is reserved. All connections from a partner with the symbolic name $PESIT$ are rejected.

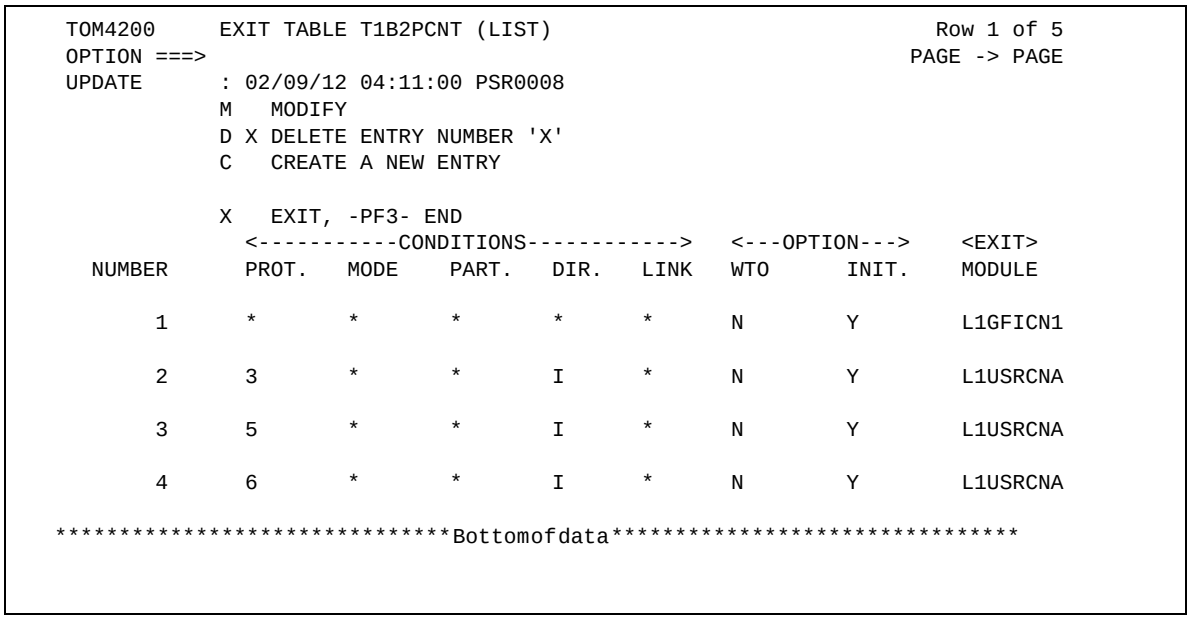
Implementing User Access Control

To implement user access control, you add the exit L1USRCNA to the connection exits table and define the default partners in the Partners directory.

Activating the Exit

The exit can be used with protocols PeSITD, PeSITE and FTP, and is only valid for an incoming request (DIR = I). The following screen shows that the first exit L1GFICN1 is active for all protocols and that the User Access control exit is active for PeSITD (PROT. = 3), PeSITE (PROT. = 5) and FTP (PROT. = 6). The exit is activated for inbound calls (DIR. = I), all session profiles (MODE = *), all partner types (PART. = *), and all link types (LINK = *). No WTO message is issued when the exit is invoked (WTO = N) and the exit is invoked during Connect:Express initialization and termination.

Note: No process is performed during Connect:Express initilalization and termination. Only the INIT and TERM information messages are issued.



```
TOM4200     EXIT TABLE T1B2PCNT (LIST)                          Row 1 of 5
OPTION ===>                                                   PAGE -> PAGE
UPDATE      : 02/09/12 04:11:00 PSR0008
            M   MODIFY
            D X DELETE ENTRY NUMBER 'X'
            C   CREATE A NEW ENTRY

            X   EXIT, -PF3- END
              <-----------CONDITIONS------------>   <---OPTION--->    <EXIT>
  NUMBER      PROT.   MODE    PART.   DIR.   LINK   WTO      INIT.    MODULE

       1      *       *       *       *      *      N        Y        L1GFICN1

       2      3       *       *       I      *      N        Y        L1USRCNA

       3      5       *       *       I      *      N        Y        L1USRCNA

       4      6       *       *       I      *      N        Y        L1USRCNA

*****************************Bottomofdata*********************************
```

Defining Default Partner Profiles

The exit controls user access and, if the access is accepted, tells the monitor which symbolic partner name to use. This name should be defined in the Partners directory so that the connection can occur. The following default profiles are set by the exit:

Symbolic Partner Name	Password	Use
$PESITD$	PESITD	Default profile for a user that calls using the PeSIT D protocol.
$PESITE$	PESITE	Default profile for a user that calls using the PeSIT E protocol.
FTP	FTP	Default profile for a user that calls using the FTP protocol.

The default partner is only used for incoming calls, and enables you to leave the network address field blank. The following screen shows the definition of the PESIT E default partner. The partner type is set to Other so that the PeSIT User fields facility can be used. The session protocol number is set to 5 for PeSIT E. The total number of sesions is equal to the number of IN sesions. No OUT session is allowed. The link type is TCP/IP, and no TCP/IP address control is performed.

```
TOM4200     PARTNER OF TOM4 TO  VIEW       (2/4)
 OPTION ===>                          -ENTER- : GO ON, -PF3- : CANCEL   X : EXIT
 TYPE: COMPAT.,PESIT-E
 MOD: PSR0008  02/09/24 03:40:57        2
 SYMBOLIC NAME         : $PESITE$           DPCSID ALIAS           : -
 TOM PASSWORD          : PESITE             DPCPSW ALIAS           : -
 INITIAL STATE         : E                  APM RECEPTION CLASS    : B
 RACF USER             : RACFUS             RACF GROUP             : -

 PARTNER TYPE          : O
 SESSION PROT.NUM.-T.  : 5    : 1           RSA-DES SECURITY T.    : -
 AUTOMATIC RESTART     : YES

 LINK TYPES            : I    : -           ADJACENT PARTNER       : -
 EFF. TOTAL/IN/OUT     : 064  : 064  : 000  FLOW CONTROL T.   SLD  : -

 SNA: LUNAME  : -     LOGMODE    : -        LOGDATA   : -         DISC  : N
 X25: MCHMSC  : -     REM.ADDR.  : -        LOC.ADDR. : -
      CUG     : -     UDF        : -                CHARGE    : -
      FACILITIES       : -
 IP : ADDR.   : *                 PORT : 4107    FTP: PASV   : Y  RIGHTS : FTP7
      HOST    : MVSB.CSG.STERCOMM.COM                           PROFILE : FTPTEST
 NOTE  : DEFAULT PARTNER FOR L1USRCNA
```

Defining Symbolic Files

Any symbolic files that you want to transfer with these users should be defined with the keyword $$API$$ in the Receiving Partner or Transmitting Partner field. The keyword $$ALL$$ or the individual partner designation for the transfer would require that the partner exists in the directory.

Customizing User Access Control with M1USRCNA

Connect:Express OS/390 provides the macro M1USRCNA to enable you to customize the symbolic partner name that the monitor uses to begin the session. This symbolic partner name and the associated password must be defined in the Partners directory of Connect:Express. You can create a compilation and link job from the

EX£USRCN example provided in the *SAMPLIB* by changing the parameters for your requirements and submitting the job.

The symbolic partner name can be a constant or the keyword $RACGRP$, which means that the symbolic name of the partner is determined dynamically. This is the RACF Group of the user who is making a connection. All possible Groups must be defined in the Partners directory of Connect:Express.

Observe the following rules when you customize M1USRCNA:

- The connection user exit must be re-entrant and AC=1.
- The CSECTNAM field is mandatory.
- If a parameter is omitted, the default value is used.
- A parameter can be set to blank by the following syntax: 'PARM=,' (length = 0).
- The symbolic partner name cannot be blank (length > 0).

The following table describes the parameters of the macro M1USRCNA.

Parameter	Description	Default Value
CSECTNAME= 1 to 8 char.	Required. Specifies the Internal and external name of the load module. This name must be declared in the Connection user exit table associated with the protocol. Example: L1TSTCNA	
FTPPBYP= 0 to 8 char.	Defines the FTP users that are not processed by the exit. The length of this parameter is processed. This parameter can be used during tests. Example: 'FTP' means that all partners which name starts with 'FTP' are not controlled.	FTP
FTPPART= 1 to 8 char.	Specifies the symbolic FTP partner name that is returned when an FTP user is connected. This name must be defined in the partners directory of Connect:Express. The keyword $RACGRP$ is used if the symbolic name is required to be the RACF group. Example: FTPIN, $RACGRP$	FTP
FTPPASW= 0 to 8 char.	Specifies the symbolic password that must be associated to the symbolic FTP partner. Example: FTPPASS	FTP
FTPEPRC= 0 to 8 char.	Specifies the FTP error code that is sent to the user if the connection is rejected. Example: 532	530
PSDPBYP= 0 to 8 char.	Defines the PeSIT D users that are not processed by the exit. The length of this parameter is processed. This parameter can be used during tests. Example: 'PSD' means that all partners whose name starts with 'PSD' are not controlled.	PSD
PSDPART= 1 to 8 char.	Specifies the symbolic PeSIT D partner name that is returned when a PeSIT D user is connected. This name must be defined in the partners directory of Connect:Express. The keyword $RACGRP$ is used if the symbolic name is required to be the RACF group. Example: PESDIN, $RACGRP$	$PESITD$
PSDPASW= 0 to 8 char.	Specifies the symbolic password that must be associated to the symbolic PeSIT D partner. Example: PESDPASS	PESITD
PSDEPRC= 0 to 8 char.	Specifies the PeSIT error code that is sent to the user if the connection is rejected. Example: 300	304

Parameter	Description	Default Value
PSEPBYP= 0 to 8 char.	Defines the PeSIT E users that are not processed by the exit. The length of this parameter is processed. This parameter can be used during tests. Example: 'PSE'means that all partners which name starts with 'PSE' are not controlled.	PSE
PSEPART= 1 to 8 char.	Specifies the symbolic PESIT E partner name that is returned when a PeSIT E user is connected. This name must be defined in the partners directory of Connect:Express. The keyword $RACGRP$ is used if the symbolic name is required to be the RACF group. Example: PESDIN, $RACGRP$	$PESITE$
PSEPASW= 0 to 8 char.	Specifies the symbolic password that must be associated to the symbolic PeSIT E partner. Example: PESEPASS	PESITE
PSEEPRC= 0 to 8 char.	Specifies the PeSIT error code that is sent to the user if the connection is rejected. Example: 300	304

The following example is extracted from the EX£USRCN file in *SAMPLIB*:

```
000046 //TEST     EXEC    ASMHCL,M=L1TSTCNA,      <=== SAME NAME AS CSECT
000047 //            PARMASM='RENT',PARMLKED='RENT,AC=1'
000063 *-------------------------------------------------------------------
000064 * USE WITH RACF GROUP, FTP AND DEFAULT PESIT'S
000065 *-------------------------------------------------------------------
000066       M1USRCNA CSECTNAME=L1TSTCNA,   <=== XXXXXXXX         CSECT       *
000067                FTPPBYP=,             <=== XXXXXXXX/''      BYPASS      *
000068                FTPPART=$RACGRP$,     <=== X..X/$RACGRP$/'' PARTNER     *
000069                FTPPASW=PSR           <=== XXXXXXXX/''      PASSWORD
000070 ***              FTPEPRC=            <=== XXX DEFAULT ERROR-PRC IS KEPT
000071          END
000072 //
```

The connection exit must be re-entrant and you must set the parameter AC=1 for link-edit. The preceding example shows that the internal and external name of the load module is L1TSTCNA. This name must be declared in the user connection exit table (CNT) for FTP partners. In addition, no bypass is used, and the RACF group is used as the symbolic partner name. All possible RACF groups must be defined in the Partners directory of Connect:Express as FTP partners. The symbolic password of these partners is PSR. If the connection is rejected, the default FTP error code 530 is sent in the FTP REPLY.

Outgoing Request Control

You can use the L1B2P006 driver module for request control. The actual user exit is link-edited with L1B2P006 after the 'USEREXIT' field of the driver (V type DC field) is updated.

The user exit sends you the D1B2PREQ parmlist and you can reject or update the transfer request parmlist before it is sent to Connect:Express. You can set the return code to '4,' request updated, or '8,' request rejected.

Connect:Express Initialization Interface

During initialization, Connect:Express initializes the TOM address space, the AFM address space, and all eligible APM address spaces. At this time, the L1B2PCNX and L1APMSRV drivers process the server exits that you defined with the initialization option in the T1B2PCNT and T1APMSRT tables. Then, the journal exit that you defined in the UEXJNL= parameter of the SYSIN file executes. If UEXJNL is set to L1B2PDIX, the journal exit driver processes all the journal exits described in its SYSINEXT file. When the TOM initializes,

the L1B2PCNX driver processes the server exits available for PeSIT and Odette connections. When the AFM initializes, the L1B2PCNX driver processes the server exits available for FTP connections. When the APM initializes, the L1APMSRV driver processes the server exits available for PeSIT and Odette transfers. Any server exit which returned an error at initialization is disabled. Each time an EAS is started to process FTP transfers, the L1AFMSRV driver processes the server exits available for FTP transfers.

During Connect:Express termination, the TOM address space, the AFM address space, and all active APM address spaces terminate as well. The server and journal exits are invoked, if the exit was not disabled during initialization. The APM and AFM can be stopped and restarted by the TOM, and the EAS can be started and stopped by the AFM. Each time they are stopped, the termination process is executed and each time they are started, the initialization process is executed. The journal exits are also invoked in the TOM address space during initialization and termination.

TOM and AFM Initialization Interface

One or more server user exits can take control after Connect:Express is initialized and before it is active. These functions are specified as parameters in the T1B2PCNT exit table. If an error or ABEND occurs during execution of the exit, it will be disabled. These connection exits are activated either in the TOM address space or in the AFM address space, depending on the protocol used.

One or more journal exits can be executed in the TOM address space.

Initialization and Termination Calls to Connection Exits

If the server exit has been invoked during initialization, it will be invoked again during the connections for which it is eligible and during Connect:Express termination. The communication area is set for one way exchange. It is described in the D1B2PCNX *MACLIB* DSECT and addresses the following information:

Action	Description
Connect:Express initialization program → EXIT routine	The exit routine receives control with register R1 pointing to a word containing the address of the communication area.
EXIT routine → Connect:Express initialization program	The exit routine returns control to the protocol program with register R15. Register R15 contains a code less than or equal to 90.

The following screen shows an example of an exit call at the initialization of Connect:Express.

```
D1B2PCNX DSECT
************************************ HEADER
CNXSSNAM DS    CL4                   SUBSYSTEM NAME
/...
CNXCALTY DS    CL3'OPC'              CALL TYPE
```

The screen below shows an example of an exit call at the termination of Connect:Express.

```
D1B2PCNX DSECT
************************************ HEADER
CNXSSNAM DS    CL4                   SUBSYSTEM NAME
/...
CNXCALTY DS    CL3'CLC'              CALL TYPE
/...
```

Initialization and Termination Calls to Journal Exits

One or more journal exits can take control after Connect:Express is initialized and before it is active. The communication area is set for one way exchange. It is described in the D1B2PJNL *MACLIB* DSECT and addresses the following information:

Action	Description
Connect:Express initialization program → EXIT routine	The exit routine receives control with register R1 pointing to a word containing the program address of the communication area. No action is expected from the Exit.

The following screen shows an example of a journal exit call at the initialization of Connect:Express.

```
D1B2PJNL DSECT

Z45ENTRYDS0C
*-------------------------------------------------------------------
*        FILE    FIELD
*-------------------------------------------------------------------
Z45FILEN DS    0CL8                    FILE NAME
Z45KEYWR DS    XL4 OR LAST WRITTEN JNL KEY
Z45KINTR DS    CL4'INIT'            INIT-TERM KEY
Z45DSNAM DS    CL44                    DATA SET NAME
```

The following screen shows an example of a journal exit call at the termination of Connect:Express.

```
D1B2PJNL DSECT

Z45ENTRYDS0C
*-------------------------------------------------------------------
*        FILE    FIELD
*-------------------------------------------------------------------
Z45FILEN DS    0CL8                    FILE NAME
Z45KEYWR DS    XL4 OR LAST WRITTEN JNL KEY
Z45KINTR DS    CL4'TERM'            INIT-TERM KEY
Z45DSNAM DS    CL44                    DATA SET NAME
```

APM and EAS Initialization Interface

One or more server exits can take control after the APM or the EAS is initialized and before it is active. These functions are specified as parameters in the T1B2PCNT and T1APMSRT exit tables. If an error occurs during execution of the server exit, it will be disabled.

If the server exit has been invoked during initialization, it will be invoked again during the transfers for which it is eligible and during APM or EAS termination. The communication area is set for one way exchange and described in the D1B2PRUEX DSECT supplied in the *MACLIB*. It addresses the following items:

Action	Description
APM initialization program → EXIT routine	The exit routine receives control with register R1 pointing to a word containing the address of the communication area.
EXIT routine → APM initialization program	The exit routine returns control to the protocol program with register R15. Register R15 contains a code less than or equal to 90.

The screen below shows an example of communication at the initialization of the APM.

```
D1B2RUEX DSECT
************************************ HEADER
UEXSSNAM DS    CL4                   SUBSYSTEM NAME
/...
UEXCALTY DS    CL3'OPS'              CALL TYPE
/...
```

The following screen shows an example of communication at the termination of the APM.

```
************************************ HEADER
D1B2RUEX DSECT
UEXSSNAM DS    CL4                   SUBSYSTEM NAME
/...
UEXCALTY DS    CL3'CLS'              CALL TYPE
/...
```

Exits Processed in the TOM and AFM Address Spaces

All exits that are executed in the TOM Address space need a hot-start of Connect:Express to be refreshed. These exits are loaded during initialization and disabled if their execution is unsuccessful. Exits that are executed in the AFM address space need a stop/start of the AFM by Connect:Express to be refreshed. The connection interface is executed in both the TOM and AFM address spaces. The Journal interface is only executed in the TOM address space.

Connection Interface

One or more server exits can take control before the session is opened and after the session is closed. These server exits must be reentrant. These functions are entered as parameters in the T1B2PCNT exit table that is processed by the L1B2PCNX driver. A connection exit can be called during initialization and termination of Connect:Express. See *Connect:Express Initialization Interface* on page 2-8.

Each eligible exit is called before the connection attempt if it is an outgoing call, and after the connection is detected if it is an incoming call. If one of the exits involved in the connection returns a non-zero code, the process was unsuccessful. All exits are invoked regardless of the return codes from the previous exits in the list. If the Connection exit process during the open session phase was successful, the connection exit process is invoked again during the close session phase. The following example shows how the driver works. Server1, server2, server3, and server4 were defined in the T1B2PCNT table with their associated session profiles.

When the session opens:

1. Server 1 is invoked and returns **OK.**
2. Server 2 is not invoked because it doesn’t match the session profile.
3. Server 3 is invoked and returns **not OK.**
4. Server 4 is invoked and returns **OK.**
5. Final result is **not OK** because the server 3 return code was not OK.

When the session closes, there are no calls to servers.

The communication area enable two way exchanges. It is described in the D1B2PCNX DSECT, shown below. This is in the *MACLIB* and addresses environment information, Partner and network identification, return codes, and parameters set by the protocol.

Connection OPEN/CLOSE Parmlist Structure D1B2PCNX

```
D1B2PCNX DSECT
************************************ HEADER
CNXSSNAM DS    CL4                   SUBSYSTEM NAME
/...
CNXCALTY DS    CL3'CNX'              CALL TYPE
/...
*---------------------------------- IDENTIFICATION ------
CNXPRC   DS    CL3                    PROTOCOL RETURN CODE
CNXTRC   DS    CL4                     C:X  RETURN CODE
/...
****************************** APPLICATION ***************
CNXAPIXX EQU   *
*-------------------------*
* FTP APPLICATION AREA    *
*-------------------------*
CNXAPIFT ORG   CNXAPIXX  * * * * * * * * * *
*-------------------------*
* ODETTE APPLICATION AREA  *
*-------------------------*
CNXAPI02 ORG   CNXAPIXX  * * * * * * * * * *
*-------------------------*
* PeSIT D APPLICATION AREA *
*-------------------------*
CNXAPI03 ORG   CNXAPIXX  * * * * * * * * * *
*-------------------------*
* ETEBAC1,2,3 APPLI AREA   *
*-------------------------*
CNXAPI04 ORG   CNXAPIXX  * * * * * * * * * *
*-------------------------*
* PESIT-E APPLICATION AREA *
*-------------------------*
CNXAPI05 ORG   CNXAPIXX  * * * * * * * * * *
```

If an error occurs during execution and it is detected by the exit routine, the protocol program stops the connection and returns the appropriate TRC code. Refer to Appendix B for a complete listing of return codes. The following screen shows the values of a TRC issued for a user connection exit error.

```
      TRC=46rr     where rr = 01 to 90 for errors detected by exit
                            = 91 invalid user exit error code
                            = 93 disable exit is requested
                            = 95 exit timer expired
```

If an ABEND occurs during the execution of one server exit, or if the exit requests it, the server exit is disabled. Two connection server exits are provided with Connect:Express. The L1GFICN1 module is a connection server exit from the Utilities option that tracks abnormal end of sessions. L1USRCNA is a connection server exit that can be used for security control . L1USRCNA source code is provided in the *SAMPLIB*.

Outgoing Sessions

The exit routine takes control before establishing a connection with a partner.

Action	Description
Protocol program → EXIT routine	The exit routine receives control with register R1 pointing to a word containing the address of the communication area. Application fields are initialized with default values.
EXIT routine → Protocol program	The exit routine returns control to the protocol program with register R15 and application fields. Register R15 contains a code less than or equal to 90 that indicates if the protocol program can perform the connection. Application fields contain the values to be sent to the Partner in the transfer protocol fields.

If an error occurs during execution of the exit routine, the protocol program does not proceed to establish a session and issues a TRC=46xx code with xx = R15.

Incoming Sessions

The exit routine takes control after the protocol connection command was received and a partner entry was found in the Connect:Express directory.

Action	Description
Protocol program → EXIT routine	The exit routine receives control with register R1 pointing to a word containing the address of the communication area. Application fields are initialized with values found in the protocol fields.
EXIT routine → Protocol program	The exit routine returns control to the protocol program with register R15. Register R15 contains a code that indicates what kind of action the protocol program must perform.

If an error occurs during execution of the exit routine, the protocol program rejects the connection with the PRC returned by the user or the default PRC and a TRC=46xx code with xx = R15.

Journal Interface

Connect:Express can give control to a user exit routine at the end of transfer using the UEXJNL=xxxxxxx parameter. This exit is implemented by a mechanism which performs two functions.

- It protects Connect:Express from malfunctions. If an error occurs, the exit routine is not invoked. A message is then sent to the operator console and the control panel shows the exit as DISABLED.
- It manages the events in a queue.

All journal exits are called during the initialization and termination of Connect:Express. See *Connect:Express Initialization Interface* on page 2-8.

The exit takes control during the following circumstances when Connect:Express is initialized:

- At each normal end-of-transfer.
- At each abnormal end-of-transfer.
- After call attempts to a Partner fail.

The linkage conventions are IBM standards. Register R1 contains the address of a word which has the address of the communication area. This area contains the same information as the record written in the Journal file (SYSJNL). When several exits are used (A-SIT, B-SIT, user EXIT), it is possible to execute the L1B2PDIX module. This exit is entered as a parameter in the Connect:Express SYSIN (UEXJNL). This ensures that programs are loaded during initialization, and it creates the communication area for the monitor at each end-of-transfer. See the *SAMPLIB* for more information about implementation. Refer to the EXIT£DIX example and the £SYSEXT module in the *PARMLIB*. Examples of two end-of-transfer conditions are shown below.

End-of-Reception for a File: Zone Passed to Exit

```
          05 ZON-CR-RECEPT.
            10  Z45FILE.
                15  Z45FILEN          PIC X(8).      < SIT00001
                15  Z45DSNAM          PIC X(44).     < EXP.SIT00001.FILE
                15  Z45FILTY          PIC X.         < S
                15  Z45DIREC          PIC X.         < R
                15  Z45TRPRT          PIC X.         < 1
                15  Z45PPRNB          PIC X.         < 1
                15  Z45FPSIT.
                    20  Z45PSFTY      PIC 9(5).      < 00001
                    20  Z45PSNAM      PIC X(5).      < ?????
                    20  Z45CRDAT      PIC X(6).      < YYMMDD
                    20  Z45CRTIM      PIC X(6).      < HHMMSS
                    20  Z45PSTAM      PIC X(64).     < ?????
                    20  Z45EXDAT      PIC X(6).      < YYMMDD
                    20  Z45EXTIM      PIC X(6).      < HHMMSS
                    20  Z45FILDA      PIC X.         < E, A, B
                    20  Z45FILR1      PIC X(5).      < SPACES
            10  Z45PART.
                15  Z45PARTN          PIC X(8).      < I4X00001
                15  Z45PARTY          PIC X.         < O
                15  Z45PARLK          PIC X.         < N
                15  Z45PARTX          PIC X.         < 0
                15  Z45PARVP          PIC X.         < 1
                15  Z45PARST          PIC X.         < S
                15  Z45PARSP          PIC X.         < 1
                15  Z45SECTB          PIC X(2)       < SECURITY TABLE NUM.
                15  Z45PPSIT.
                    20  Z45LOTYP      PIC 9.         < 2
                    20  Z45LOAPN      PIC 9(5).      < 00022
                    20  Z45PATYP      PIC 9.         < 4
                    20  Z45PAAPN      PIC 9(5)       < 00001
                    20  Z45FILR2      PIC X(38).     < SPACES
            10  Z45REQNB              PIC 9(8).      < REQUEST NB
            10  Z45REQID              PIC X(8).      < I4X00001
            10  Z45REQRS         COMP PIC 9(4).      < RESTART NB
            10  Z45REQCL              PIC X.         < CLASS
            10  Z45SPSTN              PIC X.         < SESS.TABLE NUM.
            10  Z45DATEB       COMP-3 PIC 9(8).      < 0YYMMDDF
            10  Z45HOURB       COMP-3 PIC 9(8).      < 0HHMMSSF
            10  Z45DATEE       COMP-3 PIC 9(8).      < 0YYMMDDF
            10  Z45HOURE       COMP-3 PIC 9(8).      < 0HHMMSSF
            10  Z45TIMET       COMP-3 PIC 9(8).      < 0HHMMSSF
            10  Z45BYTTN         COMP PIC 9(8).      < BYTES COUNT
            10  Z45BYTEN         COMP PIC 9(8).      < BYTES COUNT
            10  Z45RECDN         COMP PIC 9(8).      < RECORDS COUNT
            10  Z45SDRVN         COMP PIC 9(8).      < SEND/RECV COUNT
            10  Z45ABNCD              PIC X(4).      < 0000 OR RET-COD SXXX
            10  Z45PRTRC              PIC X(4).      < 0000 OR SIT-COD PXXX
            10  Z45SYSRC              PIC X(4).      < 0000 OR SYST. CODE X
            10  Z45INTYP              PIC X.         < INTERRUPTION TYPE
            10  Z45FTRID              PIC XXX.       < 0 → 32767
            10  Z45APMNB              PIC XX.        < APM NUMBER
            10  Z45EFFNB              PIC XX.        < EFFECTOR NUMBER
            10  Z45RTDSN              PIC X(44).     < REMOTE DSNAME
            10  Z45TAPID              PIC X(82).     < APPLI IDENTIF.
            10  Z45FILR5              PIC x(66).     < SPACES
            10  Z45CRKEY              PIC XL4        < current key
            10  Z45FILR6              PIC X(212).    < SPACES
```

End-of-Transmission for a File: Zone Passed to Exit

```
05 ZON-CR-RECEPT.
   10  Z45FILE.
       15  Z45FILEN         PIC X(8).    < SIT00004
       15  Z45DSNAM         PIC X(44).   < EXP.SIT00004.FILE
       15  Z45FILTY         PIC X.       < S
       15  Z45DIREC         PIC X.       < T
       15  Z45TRPRT         PIC X.       < 1
       15  Z45PPRNB         PIC X.       < 1
       15  Z45FPSIT.
           20  Z45PSFTY     PIC 9(5).    < 00004
           20  Z45PSNAM     PIC X(5).    < ?????
           20  Z45CRDAT     PIC X(6).    < YYMMDD
           20  Z45CRTIM     PIC X(6).    < HHMMSS
           20  Z45PSTAM     PIC X(64).   < ?????
           20  Z45EXDAT      PIC X(6).   < YYMMDD
           20  Z45EXTIM      PIC X(6).   < HHMMSS
           20  Z45FILDA      PIC X.      < E, A, B
           20  Z45FTRID      PIC XXX.    < ???   TRF ID
           20  Z45FILR1      PIC XX.     < SPACES
   10  Z45PART.
       15  Z45PARTN         PIC X(8).    < O4X00001
       15  Z45PARTY         PIC X.       < O
       15  Z45PARLK         PIC X.       < N
       15  Z45PARTX         PIC X.       < 0
       15  Z45PARVP         PIC X.       < 1
       15  Z45PARST         PIC X.       < S
       15  Z45PARSP         PIC X.       < 1
       15  Z45PPSIT.
           20  Z45LOTYP     PIC 9.       < 2
           20  Z45LOAPN     PIC 9(5).    < 00022
           20  Z45PATYP     PIC 9.       < 4
           20  Z45PAAPN     PIC 9(5)     < 00001
           20  Z45FILR2     PIC X(38).   < SPACES
   10  Z45REQU.
   10  Z45REQNB             PIC 9(8).    < REQUEST NB
   10  Z45REQID             PIC X(8).    < JOBNAME
   10  Z45REQRS       COMP  PIC 9(4).    < RESTART NB
   10  Z45REQCL             PIC X.       < CLASS
   10  Z45FILR3             PIC X.       < SPACES
   10  Z45DATEB     COMP-3  PIC 9(8).    < 0YYMMDDF
   10  Z45HOURB     COMP-3  PIC 9(8).    < 0HHMMSSF
   10  Z45DATEE     COMP-3  PIC 9(8).    < 0YYMMDDF
   10  Z45HOURE     COMP-3  PIC 9(8).    < 0HHMMSSF
   10  Z45TIMET     COMP-3  PIC 9(8).    < 0HHMMSSF
   10  Z45BYTTN       COMP  PIC 9(8).    < BYTES COUNT
   10  Z45BYTEN       COMP  PIC 9(8).    < BYTES COUNT
   10  Z45RECDN       COMP  PIC 9(8).    < RECORDS COUNT
   10  Z45SDRVN       COMP  PIC 9(8).    < SEND/RECV COUNT
   10  Z45ABNCD             PIC X(4).    < 0000 OR RET-COD SXXX
   10  Z45PRTRC             PIC X(4).    < 0000 OR SIT-COD PXXX
   10  Z45IDTRF         COMP PIC 9(8).     <
   10  Z45ZITRF         REDEFINES Z45IDTRF.
     15  Z45FILR4            PIC X.       <
     15  Z45FTRID            PIC XXX.     < 0 → 32767
   10  Z45SPSTN             PIC X        < 0 → 7
   10  Z45FILR5             PIC XXX.     < SPACES
   10  Z45CRKEY             PIC XL4      < current key
   10  Z45FILR6             PIC X(212).  < SPACES
```

System Security Interface

Connect:Express provides a general SAF interface for system security.

RACF Security Interface

The RACF logical interface is controlled by RACFCN= in the PARMLIB (SYSIN) file. A standard RACF source module is provided in the *SAMPLIB* called L1USRTAC. You can modify it and replace the L1B2PTAC module that Connect:Express runs.

The AD HOC interface cannot be customized. See ADHOCN= in the PARMLIB (SYSIN) file. It can only be 'in-service','out-service', or 'unsafe'. Any incoming AD HOC request is logged in the Connect:Express SYSLOG file.

Exits Processed by the APM and EAS Address Spaces

Different kinds of exits are executed in the APM and EAS address spaces. Some are loaded and deleted when needed, like transfer initialization and termination exits, and some are loaded during APM or EAS initialization, like selection and record processing exits.

Some differences exist between the APM and the EAS. The APM only stops when Connect:Express requests it to terminate, and the EAS automatically stops when its job is finished. The exits that are loaded in the APM address space during initialization need a stop/start of the APM by Connect:Express to be refreshed. Exits that are loaded during the EAS initialization are refreshed automatically.

Selection Interface

One or more server exits can take control before the transfer is selected and after the transfer is deselected. These server exits must be reentrant. These functions are entered as parameters in the T1APMSRT exit table processed by the L1APMSRV driver. A selection exit can be called during initialization and termination of Connect:Express. See *Connect Express Initialization Interface* on page 2-8.

Each eligible exit is invoked before sending the transfer request to the Partner if it is an outgoing call, and after receiving the request if it is an incoming call. If one of the exits involved in the selection returns a non-zero code, the process was unsuccessful. All exits are invoked regardless of the return codes from the previous exits in the list. If the selection exit process during the selection phase returns an OK, the selection exit process is invoked again during the de-selection phase. The following example shows how the driver works. Server1, server2, server3, and server4 were defined in the T1APMSRT table with the associated transfer profiles.

During selection:

1. Server 1 is invoked and returns **OK**
2. Server 2 is not invoked because it doesn’t match the session profile
3. Server 3 is invoked and returns **OK**
4. Server 4 is invoked and returns **OK**
5. Final result is **OK**

During de-selection:

1. Server 1 is invoked and returns **OK**
2. Server 2 is not invoked because it doesn’t match the session profile
3. Server 3 is invoked and returns **not OK**
4. Server 4 is invoked and returns **OK**
5. Final result is **not OK**

The communication area enables two way exchanges and is described in the D1B2RUEX DSECT, shown below. This is in the *MACLIB* and addresses environment information, return codes, parameters set by the protocol, and Partner, file and transfer identification.

Selection, INIT/TERM, Security Parmlist Structure D1B2RUEX

```
D1B2RUEX DSECT
************************************ HEADER
UEXSSNAM DS    CL4                    SUBSYSTEM NAME
UEXAPMNB DS    CL2                    APM NUMBER
UEXEFFNB DS    CL2                    EFF NUMBER
UEXCALTY DS    CL3'SEL'               CALL TYPE
UEXDATE  DS    0C                     CURRENT TRANSFER DATE
UEXTIME  DS    0C                     CURRENT TRANSFER TIME
*---------------------------------- IDENTIFICATION ------
*------------------ FILE
UEXDDNM  DS    CL8                    DDNAME
UEXPART  DS    CL8                    PARTNER ID
UEXDSNM  DS    CL44                   DSNAME
UEXDRCT  DS    CL1                    TRANSFER DIRECTION
UEXTYPE  DS    CL1                    INITIALIZA. status (A, I)
*                                     TERMINATION status (E, F, S)
*------------------ TRANSFER
UEXTRFID DS    XL4                    EXTERNAL TRF IDENT
UEXREQNB DS    XL4                    INTERNAL TRF IDENT
*------------------ RETURN CODES
UEXSRC   DS    CL4                    SYSTEM RETURN CODE
UEXTRC   DS    CL4                     C:X  RETURN CODE
UEXPRC   DS    CL3                    PROTOCOL RETURN CODE
UEXIDT   DS    X                      TRANSFER END STATUS
UEXTRFOK EQU   X'00'                    TRF OK                   PESIT
UEXTRSTR EQU   X'04'                    TRF WILL BE RESTARTED    PESIT
UEXTSUSP EQU   X'08'                    TRF SUSPENDED            PESIT
UEXTWAIT EQU   X'0C'                    STOPPED BY WAITER        PESIT
UEXTINIT EQU   X'10'                    STOPPED BY INITIATOR     PESIT
UEXTERSL EQU   X'14'                    ERROR DURING SELECTION
*******************************  RESULTS  *****************
*************************** USER COMMUNICATION *************
UEXPRMAD DS    XL4                    USER PARM FROM APM JCL ADD(USD=)
UEXUSDAD DS    XL4                    USER EXIT COMMUNICATION AREA ADD
******************************** ALLOCATION *****************
UEXALRST DS    XL1                    RESTART FLAG
******************************** APPLICATION ****************
UEXAPIXX EQU   *
*-------------------------*
* FTP APPLICATION AREA    *
*-------------------------*
UEXAPIFT ORG   UEXAPIXX  * * * * * * * * * *
*-------------------------*
* ODETTE APPLICATION AREA *
*-------------------------*
UEXAPI02 ORG   UEXAPIXX  * * * * * * * * * *
*-------------------------*
* PeSIT D APPLICATION AREA *
*-------------------------*
UEXAPI03 ORG   UEXAPIXX  * * * * * * * * * *
*-------------------------*
* ETEBAC1,2,3 APPLI AREA  *
*-------------------------*
UEXAPI04 ORG   UEXAPIXX  * * * * * * * * * *
*-------------------------*
* PESIT-E APPLICATION AREA *
*-------------------------*
UEXAPI05 ORG   UEXAPIXX  * * * * * * * * * *
```

Continued

```
******************************** SECURITY *********************
************** APPLICATION CONNEXION AREA
*
UEXCNXAP EQU   *
UEXCNXXL EQU   800
*-------------------------*
* ODETTE CONNECTION AREA   *
*-------------------------*
UEXCNX02 ORG   UEXCNXAP  * * * * * * * * * *
*
*-------------------------*
* PESIT-D CONNECTION AREA   *
*-------------------------*
UEXCNX03 ORG   UEXCNXAP  * * * * * * * * * *
*
*-------------------------*
* PESIT-E CONNECTION AREA  * (ETEBAC5)
*-------------------------*
UEXCNX05 ORG   UEXCNXAP  * * * * * * * * * *
*
```

If an error occurs during execution and it is detected by the exit routine, the protocol program stops the connection and returns the appropriate TRC code. Refer to Appendix B for a listing of all return codes. The following screen shows the values of TRC issued for user selection exit errors.

```
      TRC=46rr     where rr = 01 to 90 for errors detected by exit
                              = 91 invalid user exit error code
                              = 95 exit timer expired
```

If an ABEND occurs during the execution of one server exit, it will be disabled. The L1GFICN1 module is a selection server exit from the Utilities option that tracks abnormal end of transfers.

Outgoing Selection

The exit routine takes control before establishing selection with a Partner.

Action	Description
Protocol program → EXIT routine	The exit routine receives control with register R1 pointing to a word containing the address of the communication area. Application fields are initialized with default values.
EXIT routine → Protocol program	The exit routine returns control to the protocol program with register R15 and application fields. Register R15 contains a code less than or equal to 90 that indicates if the protocol program can perform the selection. Application fields contain the values to be sent to the Partner in the transfer protocol fields.

If an error occurs during execution of the exit routine, the protocol program does not demand selection and issues a TRC=46xx code with xx = R15.

Incoming Selection

The exit routine takes control after the protocol selection command has been received.

Action	Description
Protocol program → EXIT routine	The exit routine receives control with register R1 pointing to a word containing the address of the communication area. Application fields are initialized with values found in the protocol fields.
EXIT routine → Protocol program	The exit routine returns control to the protocol program with register R15 which contains a code that indicates the kind of action the protocol program must perform.

If an error occurs during execution of the exit routine, the protocol program rejects the selection with the PRC returned by the user, or the default PRC and a TRC=46xx code with xx = R15.

Mixing Server Exits and Standard User Exits

Server exits are defined in the connection exits and selection exits tables. Server exits are called by two drivers, the L1B2PCNX and the L1APMSRV. The L1B2PCNX is called in the TOM address space during Connect:Express initialization and termination, and during the current connection process. The L1APMSRV driver is called in the APM address space during the APM initialization and termination, and during the current transfer selection phases of file selection and allocation.

Some user exits perform common tasks, like data translation, file allocation or end of transfer process. These are called standard user exits. Standard user exits can be called at any time from the transfer initialization phase to the transfer termination phase. You can have different exits for each task, but each task must be in the right place. For example, file allocation might be performed by the allocation exit during initialization, and data translation by the data translation exit during record processing. Standard exits are not involved in the selection phase. These exits are activated by entering the name of the exit, or the presentation number, in a field in the file attributes definition.

When entering L1APMSRV in a user exit field, standard calls are directed to the L1APMSRV driver which processes the T1APMSRT table and the server exits that are defined in it. This way the same exit could process the different phases and it would be defined in one place. You can use both standard user exits and server exits with a single transfer.

Linking Information between User Exits

Some fields can be sent to the different exits and enable you to setup an end to end process from transfer initialization to transfer termination.

For example, the UEXUSDAD field, from the D1B2RUEX structure, is initialized to zero and passed through SEL → IEX → PEX (PEXUSDAD from D1B2RPEX structure) → SEC → I/O → TEX → SEL calls to user exits. So are UEXPRMAD from PARM of the APM EXEC and UEXAPIXX fields, except those of I/O calls. The UEXCNXAP field is passed from the D1B2PCNX structure in the TOM address space to the D1B2RUEX structure in the APM address space only for the PeSIT and Odette protocols.

Note: If a GETMAINED area (sub-pool zero) address is passed through these fields, be aware that if an abend of the protocol task with UEXTYPE=S (SEL and TEX) calls following a protocol task ABEND, this address is no longer available.
Using the BRX=3 or 4 option in the APM PARM of EXEC, which corresponds to ATTACH of user exits, makes the GETMAINED area unsharable unless a special SUBPOOL is used.

Transfer Initialization Interface

A user exit can take control before a file is opened. This function is entered as parameters in the Files directory. If the name of the exit is L1APMSRV, the server exit driver is used and all eligible server exits are executed.

Action	Description
Protocol program → EXIT routine	The exit routine takes control with register R1 pointing to a word containing the address of the communication area.
EXIT routine → Protocol program	The exit routine returns control to the protocol program with register R15. Register R15 contains a code that indicates what kind of action the protocol program must perform.

The communication area enables two way exchanges, and is described in the D1B2RUEX DSECT, shown below. It is in the *MACLIB* and addresses environment information, file and transfer identification, return codes, allocation parameters, and parameters set by the protocol.

Selection, Initialization/Termination, Security Parmlist Structure D1B2RUEX

```
D1B2RUEX DSECT
************************************ HEADER
UEXSSNAM DS    CL4                   SUBSYSTEM NAME
UEXAPMNB DS    CL2                   APM NUMBER
UEXEFFNB DS    CL2                   EFF NUMBER
UEXCALTY DS    CL3'IEX'              CALL TYPE
UEXDATE  DS    0C                    CURRENT TRANSFER DATE
UEXTIME  DS    0C                    CURRENT TRANSFER TIME
*---------------------------------- IDENTIFICATION ------
*------------------ FILE
UEXDDNM  DS    CL8                   DDNAME
UEXPART  DS    CL8                   PARTNER ID
UEXDSNM  DS    CL44                  DSNAME
UEXDRCT  DS    CL1                   TRANSFER DIRECTION
UEXTYPE  DS    CL1                   INITIALIZA. status (A, I)
*                                    TERMINATION status (E, F, S)
*------------------ TRANSFER
UEXTRFID DS    XL4                   EXTERNAL TRF IDENT
UEXREQNB DS    XL4                   INTERNAL TRF IDENT
*------------------ RETURN CODES
UEXSRC   DS    CL4                   SYSTEM RETURN CODE
UEXTRC   DS    CL4                    C:X  RETURN CODE
UEXPRC   DS    CL3                   PROTOCOL RETURN CODE
UEXIDT   DS    X                     TRANSFER END STATUS
UEXTRFOK EQU   X'00'                   TRF OK                  PESIT
UEXTRSTR EQU   X'04'                   TRF WILL BE RESTARTED   PESIT
UEXTSUSP EQU   X'08'                   TRF SUSPENDED           PESIT
UEXTWAIT EQU   X'0C'                   STOPPED BY WAITER       PESIT
UEXTINIT EQU   X'10'                   STOPPED BY INITIATOR    PESIT
UEXTERSL EQU   X'14'                   ERROR DURING SELECTION
```

Continued

```
*
*******************************  RESULTS  *****************
*
*************************** USER COMMUNICATION *************
UEXPRMAD DS    XL4                    USER PARM FROM APM JCL ADD(USD=)
UEXUSDAD DS    XL4                    USER EXIT COMMUNICATION AREA ADD
*
******************************* ALLOCATION *****************
*
UEXALRST DS    XL1                    RESTART FLAG
*
******************************* APPLICATION ****************
UEXAPIXX EQU   *
*-------------------------*
*  C:X  APPLICATION AREA *
*-------------------------*
UEXAPI03 ORG   UEXAPIXX  * * * * * * * * * *
*-------------------------*
* ODETTE APPLICATION AREA  *
*-------------------------*
UEXAPI02 ORG   UEXAPIXX  * * * * * * * * * *
*-------------------------*
* ETEBAC1,2,3 APPLI AREA   *
*-------------------------*
UEXAPI04 ORG   UEXAPIXX  * * * * * * * * * *
*-------------------------*
* PESIT-E APPLICATION AREA *
*-------------------------*
UEXAPI05 ORG   UEXAPIXX  * * * * * * * * * *

******************************* SECURITY *********************
************** APPLICATION CONNEXION AREA
*
UEXCNXAP EQU   *
UEXCNXXL EQU   800
*-------------------------*
* ODETTE CONNECTION AREA   *
*-------------------------*
UEXCNX02 ORG   UEXCNXAP  * * * * * * * * * *
*
*-------------------------*
* PESIT-D CONNECTION AREA   *
*-------------------------*
UEXCNX03 ORG   UEXCNXAP  * * * * * * * * * *
*
*-------------------------*
* PESIT-E CONNECTION AREA  * (ETEBAC5)
*-------------------------*
UEXCNX05 ORG   UEXCNXAP  * * * * * * * * * *
*
```

Options for R15 Code at Beginning-of-Transfer

The return code from the exit can indicate an error or an action that Connect:Express must perform. For example, the exit can prepare allocation without performing it. If the file is to be partially transferred, only Connect:Express can point to the part of the file that you will transfer.

```
R15 = binary zero    → transfer continues

R15 = 1 to 90        → errors detected by the exit (stop)

R15 = 1000           → allocate file with the returned DSNAME

R15 = 1004           → recover DSNAME if Allocation rule is 3
                       The user exit performed allocation
                     → point with returned key

R15 = 1008           → recover DSNAME if Allocation rule is 3
                     → allocate file if Allocation rule is 3
                     → point with returned key
```

If an error occurs during execution and it is detected by the exit routine, the protocol program stops the transfer and returns the appropriate TRC code. The following screen shows values of TRC issued for a user initialization exit error.

```
  TRC=4err    where e = 0 exit routine at the beginning of transfer
                      = 9 exit routine at the end of transfer
                  rr = 01 to 90 for errors detected by exit
                     = 91 invalid user exit error code
                     = 92 invalid key
                     = 95 exit timer expired
```

Note: When the user performs the file allocation, some protocol parameters, such as record length UEXALREC, record format (U03/ U05FRFMT), and file space (U03/U05SPTYP,FKBYT) must be returned through the D1B2RUEX fields.

Transfer Termination Interface

A user exit can take control after a file is closed. These functions are entered as parameters in the Files directory. If a problem occurs during transfer, the end exit will take control, and you will get a problem notification. The communication area enables two way exchanges, as shown in the following table.

Action	Description
Protocol program → EXIT routine	The exit routine receives control with register R1 pointing to a word containing the address of the communication area.
EXIT routine → Protocol program	The exit routine returns control to the protocol program with register R15. R15 contains a code which indicates what kind of action the protocol program must execute.

The D1B2RUEX DSECT, shown below, is in the *MACLIB*, and addresses the environment information, file and transfer identification, return codes, allocation parameters, and parameters set by the protocol.

Selection, Initialization/Termination, Security Parmlist Structure D1B2RUEX

```
D1B2RUEX DSECT
************************************ HEADER
UEXSSNAM DS    CL4                    SUBSYSTEM NAME
UEXAPMNB DS    CL2                    APM NUMBER
UEXEFFNB DS    CL2                    EFF NUMBER
UEXCALTY DS    CL3'TEX'               CALL TYPE
UEXDATE  DS    0C                     CURRENT TRANSFER DATE
UEXTIME  DS    0C                     CURRENT TRANSFER TIME
*---------------------------------- IDENTIFICATION ------
*------------------ FILE
UEXDDNM  DS    CL8                    DDNAME
UEXPART  DS    CL8                    PARTNER ID
UEXDSNM  DS    CL44                   DSNAME
UEXDRCT  DS    CL1                    TRANSFER DIRECTION
UEXTYPE  DS    CL1                    INITIALIZA. status (A, I)
*                                     TERMINATION status (E, F, S)
*------------------ TRANSFER
UEXTRFID DS    XL4                    EXTERNAL TRF IDENT
UEXREQNB DS    XL4                    INTERNAL TRF IDENT
*------------------ RETURN CODES
UEXSRC   DS    CL4                    SYSTEM RETURN CODE
UEXTRC   DS    CL4                     C:X  RETURN CODE
UEXPRC   DS    CL3                    PROTOCOL RETURN CODE
UEXIDT   DS    X                      TRANSFER END STATUS
UEXTRFOK EQU   X'00'                    TRF OK                  PESIT
UEXTRSTR EQU   X'04'                    TRF WILL BE RESTARTED   PESIT
UEXTSUSP EQU   X'08'                    TRF SUSPENDED           PESIT
UEXTWAIT EQU   X'0C'                    STOPPED BY WAITER       PESIT
UEXTINIT EQU   X'10'                    STOPPED BY INITIATOR    PESIT
UEXTERSL EQU   X'14'                    ERROR DURING SELECTION
*
*******************************  RESULTS  ****************
*
*************************** USER COMMUNICATION *************
UEXPRMAD DS    XL4                    USER PARM FROM APM JCL ADD(USD=)
UEXUSDAD DS    XL4                    USER EXIT COMMUNICATION AREA ADD
****************************** ALLOCATION ****************
UEXALRST DS    XL1                    RESTART FLAG
****************************** APPLICATION ***************
UEXAPIXX EQU   *
*--------------------------*
*  FTP APPLICATION AREA    *
*--------------------------*
UEXAPIFT ORG   UEXAPIXX  * * * * * * * * * *
*--------------------------*
* ODETTE APPLICATION AREA  *
*--------------------------*
UEXAPI02 ORG   UEXAPIXX  * * * * * * * * * *
*--------------------------*
*  C:X  APPLICATION AREA   *
*--------------------------*
UEXAPI03 ORG   UEXAPIXX  * * * * * * * * * *
*--------------------------*
* ETEBAC1,2,3 APPLI AREA   *
*--------------------------*
UEXAPI04 ORG   UEXAPIXX  * * * * * * * * * *
*--------------------------*
* PESIT-E APPLICATION AREA *
*--------------------------*
UEXAPI05 ORG   UEXAPIXX  * * * * * * * * * *
```

Continued

```
******************************** SECURITY *********************
************** APPLICATION CONNEXION AREA
*
UEXCNXAP EQU   *
UEXCNXXL EQU   800
*-------------------------*
* ODETTE CONNECTION AREA   *
*-------------------------*
UEXCNX02 ORG   UEXCNXAP  * * * * * * * * * *
*
*-------------------------*
* PESIT-D CONNECTION AREA   *
*-------------------------*
UEXCNX03 ORG   UEXCNXAP  * * * * * * * * * *
*
*-------------------------*
* PESIT-E CONNECTION AREA  * (ETEBAC5)
*-------------------------*
UEXCNX05 ORG   UEXCNXAP  * * * * * * * * * *
*
```

Options for R15 Code at End of Transfer

The return code from the exit can indicate an error or an action that Connect:Express must perform. For example, the exit can ask Connect:Express to deallocate the file or to interrupt or delete the file and transfer request.

```
R15 =binary zero → nothing to do
R15 =2000 → file to be deallocated
R15 =2004 → transfer context to be deleted (file and request)
R15 =2008 → file to be deallocated + transfer context deleted
```

If an error occurs during execution and it is detected by the exit routine, the protocol program stops the transfer and returns the appropriate TRC code. Refer to Appendix B for a list of TRC codes. The following screen shows values of TRC issued for a user termination exit error.

```
TRC =4err where
e    = 0 exit routine at the beginning of transfer
     = 9 exit routine at the end of transfer
rr   = 01 to 90 for errors detected by exit
     = 91 invalid user exit error code
     = 92 invalid key
     = 95 exit timer expired
```

Record Processing Interface

The record processing interface enables users to manage the data during the transfer phase. You can call an exit during processes like reading or writing the data, or selecting, changing, or checking the data.

User Exit During Transfer

Up to three user exit routines can take control during a file transfer. For this to happen, the user exits must be defined in the protocol presentation table (T1B2PPxx, where xx=09 to xx=24 [TSO/ISPF 3.3.PPxx). For each table, the user can indicate one to three exit names which receive control from the protocol program. The

protocol program uses IBM standard linkage conventions and provides a communication area which contains the items of the current transfer. If the name of the exit is L1APMSRV, the server exit is used.

The user exit is called for the first time during the open file phase. The protocol program provides a communication area. In this area, the PEXSTATS field is set to the value of EXASTART (B), but there is nothing in the file record. The user exit sets up the environment.

For each record of the file, the user exit routine takes control from the protocol program which provides a communication area. In this case, the PEXSTATS field is set to the value of EXAMIDDL (M). It contains the contents of the file record.

After all the records of the file have been processed, the protocol program gives control to the user exit during the close file phase and provides a communication area. The PEXSTATS field is set to the value EXAENDED (E), and the user exit routine must then reset the environment.

The following table summarizes the process.

Action	Description
Protocol program → EXIT routine	When the exit routine receives control for CALL with PEXSTATS = M, register R1 points to a word containing the address of the communication area. The address of the current record and its length are indicated by the communication area.
EXIT routine → Protocol program	At the end of processing, the exit routine returns control to the protocol program. R15 now contains a code which indicates what action must be done by the protocol. The length of the record can be changed by the exit according to the record format. If the record length is fixed, the length cannot be changed. If it is variable, it can be less than the original length.

The screen below shows values for the PEXSTATS field during transfer.

```
          PEXSTATS DS    CL1                    TRANSFER STATE
          EXASTART EQU   C'B'                   BEGIN
          EXAMIDDL EQU   C'M'                   MIDDLE
          EXAENDED EQU   C'E'                   END
```

Implementation

The communication area is designed to allow two-way communications. This area is described by D1B2RPEX DSECT, shown below. It is in the *MACLIB* and gives information about the environment, file and transfer identification, return codes, data area (length of the record and the current record), the position of data in the list, and the position in the file. It also includes applicable parameters conveyed by the protocol. Adding the PEXDOFFS value to the communication area address gives the address of the data area. The first two bytes of the data area are set to the current record length, and the current record is behind this field. The exit can change the length of the record and set the data area record length field with the new value. The exit can change the data in the current record.

Record Processing Parmlist D1B2RPEX

```
D1B2RPEX DSECT
************************************* HEADER (COMPATIBILITY)
PEXENTRY EQU   *
PEXSSNAM DS    CL4                    SUBSYSTEM NAME
PEXAPMNB DS    CL2                    APM NUMBER
PEXEFFNB DS    CL2                    EFF NUMBER
PEXZDATE DS    0C                     DATE
PEXZTIME DS    0C                     TIME
******************************** IDENTIFICATION ***********
*--------------------- FILE
PEXNDDN  DS    CL8                    DDNAME
PEXNLUID DS    CL8                    SESSION PARTNER ID
PEXDIREC DS    CL1                    DIRECTION
PEXSTATS DS    CL1                    TRANSFER STATE
EXASTART EQU   C'B'                    BEGIN
EXAMIDDL EQU   C'M'                    MIDDLE
EXAENDED EQU   C'E'                    END
PEXDOFFS DS    H                      DATA AREA OFFSET FROM PEXENTRY
*--------------------- TRANSFER       ---------------------
PEXTRFID DS    XL4                    EXTERNAL TRF IDENT
PEXREQNB DS    XL4                    INTERNAL TRF IDENT
PEXDSNAM DS    XL44                   DATA SET NAME
*--------------------- RETURN CODES    ---------------------
PEXSRC   DS    CL4                    SYSTEM RETURN CODE
PEXTRC   DS    CL4                     C:X  RETURN CODE
PEXPRC   DS    CL3                    PROTOCOL RETURN CODE
PEXIDT   DS    CL1                    IDT CODE
*
*************************** USER COMMUNICATION *************
PEXPRMAD DS    XL4                    USER PARM FROM APM JCL ADD(USD=)
PEXUSDAD DS    XL4                    USER EXIT COMMUNICATION AREA ADD
*
********************************* APPLICATION *************
PEXAPIXX EQU   *
*-----------------------------*
*  FTP  APPLICATION AREA      *
*-----------------------------*
PEXAPIFT ORG   PEXAPIXX                * * * * * * * *
********************************* SECURITY ****************
*
********************************* POINT FACILITY **********
PEXPOINT DS    0CL140                    PARMLIST USER
PEXHLNGT EQU   *-PEXSSNAM                PARMLIST HEADER LENGTH
PEXLDATA DS    XL2                       DATA LENGTH (BINARY)
PEXRDATA DS    0C                        FIRST BYTE OF RECORD
```

R15 Code During Transfer

The return code from the exit can indicate an error or an action that Connect:Express must perform. For example, the exit can ask Connect:Express to skip the current record, to end the transfer, or to stop calling the exit.

```
  R15 = zero           → transfer continues
  R15 = 1 to 90        → errors detected by user (stop)

- on reception:

  R15 = 1000 (SAM,VSAM)→ do not write current record

- on transmission:

  R15 = 2002(1st call) → the end of file is indicated by R15 = 2004 (if the normal
                         end of file occurs before the logical end of file this is
                         considered and error.)
  R15 = 2000           → current record not to be transmitted
  R15 = 2004           → end of file
  R15 = 2008 (VSAM)    → point the record with key provided
                         in the communication area

- on transmission/reception:

  R15 = 3000            → transfer to be cancelled
  R15 = 3004            → transfer to interrupted
  R15 = 3008            → transfer to be suspended
  R15 = 4000            → exit disabled (only last END call processed)
```

The file record format determines the options for changing record length. If the file record format is *fixed*, the record length cannot be modified. If the file record format is *variable*, the current length of the record can be changed, however it cannot exceed the maximum value given in the PEXLRECL field.

If the exit routine indicates an error, or if an error occurs when the protocol program executes a specified action, the protocol program stops the transfer and returns the appropriate TRC code. Refer to Appendix B for a complete list of TRC codes. The screen below shows values of TRC issued for user record processing errors.

```
TRC = 4err where
e   = exit number from 1 to 3
rr  = 1 to 90 if error detected by exit
    = 91 invalid user exit error code
    = 92 invalid key
    = 95 exit timer expired
    = 98 fixed length modified
    = 99 variable length > authorized limit
```

User Read/Write Exit

Read/write operations like QSAM, BPAM, VSAM, and HFS are executed by Connect:Express standard routines, however users can create their own routines. A user read/write exit model is shown below.

Model of User Read/Write Exit Routine

```
*-------------------------------------------------------------------
* R0  FUNCTION    ADDRESSES   TYPE/LENG. DIREC   VALUE
*-------------------------------------------------------------------
* (0  OPEN input  A(DDNAME)      E/8     P →E allocation ddname
* (or
* (4  OPEN output A(DDNAME)      E/8     P →E allocation ddname
*                 A(USDAD)       B/4     P →E user workarea
*                 A(RECFM)       B/1     P← E record format
*                 A(LRECL)       B/4     P← E record length
*                 A(BLKSE)       B/4     P← E block size
*                 A(DSORG)       E/1     P →E org. S,V,P,M
*                 A(T1B2Pxx)     ./.     P →E present. table
*
*-------------------------------------------------------------------
* 8  NOTE          A(NOTE-AREA)  X/136   P← E see UEXPNTPL field in D1B2RUEX
*
*-------------------------------------------------------------------
* 12 POINT         A(POINT-AREA) X/136   P →E see UEXPNTPL field in D1B2RUEX
*
*-------------------------------------------------------------------
* 16 READ          A(BUFFER)     E/4     P →E reception buffer
*                  A(BUFLEN)     E/4     P← E cur. phys. buffer length
*                  A(RECORD)     E/4     P← E record count (optional)
*
*-------------------------------------------------------------------
* 20 WRITE         A(BUFFER)     E/4     P →E buffer sent
*                  A(BUFLEN)     E/4     P →E phys. buffer length
*
*-------------------------------------------------------------------
* 24 CLOSE         A(RECORD)     B/4     P← E record count (optional)
*
*-------------------------------------------------------------------
* 28 WRITE         A(RECORD)     B/4     P← E record count (optional)
*    FORCE         A(BUFLEN)     B/4     P← E phys. buffer length
*
*-------------------------------------------------------------------
```

The following abbreviations are used in the model above.

TYPE/LENGTH	Description
E/8	Extended area (length 8)
B/4	Binary area (length 4)
X/136	Mixed area (length 136)

Direction	Description
P → E	Protocol towards exit (field received by exit)
P ← E	Exit towards protocol (field provided by exit)

You must declare the user read/write exit routine in the protocol presentation table as PRIVATE I/O EXIT. The protocol program sends the function code through R0 and provides field addresses. A communication area is given to the user exit by using standard linkage conventions. The user exit must indicate the value of the return code in register R15 when returning control to the protocol. For a normal completion, the value of the return code must be zero. For Connect:Express to know that the end of file was detected, you must enter the return code X'80000004' in the user exit. Otherwise, the return code is interpreted as an error code.

- RC=X'**8**00000rr' is a user return code: TRC=45rr.
- RC=X'**0**0wxyzrr' is a system return code: TRC=3zfn
  where 'fn' is the I/O function failed, 'z' indicates if transfer is restarted or not, and wxyzrr is issued in the APM SYSLOG file. A system return code (SRC) is indicated in the APM log.

Note: The user private I/O exit is responsible for all I/O services in place of Connect:Express. It is expected to return at OPEN time the E → P values:

- Record format RECFM ('00' = fixed, '04' = variable, '08' = undefined for sequential files).
- Record length LRECL
- Blocksize BLKSIZE

Note: The function WRITE FORCE is requested by the protocol program in relation to OS/390 sequential files data management. That function is evoked at every protocol synchronization point, and is intended to ensure that all data acknowledged is physically written onto disk. It is required for a possible restart procedure.

Controls are done and TRC=3092, 3093, 3094 are issued if one of these values is not returned. You have to implement these controls for each function that the process executes. This must be done even if the exit routine does not perform all the functions.

The user exit module must have the attributes RENT, REUS. You can implement a combination of allocation exits and private I/O exits using the UEXUSDAD/PEXUSDAD user communication area addresses.

System Errors

Return codes (R15) are processed according to their types. If the user exit detects a system error, R15 must contain a value of *x'ssrrrrrr'*, where *x'ss'* is less than X'80', and where *x'rrrrrr'* is the system return code.

The Connect:Express log indicates a message with a TRC code of 3iaa where the following values apply:

- i = 0 if normal transfer
- i = 1 if transfer restarted
- aa = the function code

The APM LOG shows a message with an SRC containing the value of register R15.

User Errors

To indicate a user error, the value in register R15 must be in the form of x'ss0000rr,' where *x'ss*' is equal to or greater than X'80' *and* where *X'rr*' is the user error. The Connect:Express log will show a message with a TRC code of 45rr where the following values apply:

- rr = 01 to 90 (user code)
- rr = 91 invalid user exit return code
- rr = 92 error on LRECL (OPEN)
- rr = 93 error on BLKSIZE (OPEN)
- rr = 94 error on RECFM (OPEN)

The APM LOG shows a message with an SRC code containing the value of register R15.

Chapter 3

Controlling Transfer Operations

This chapter describes how Connect:Express processes transfer requests and how you can control transfer operations using automation tools, scheduling features, resources, and security options.

Overview of Transfer Requests

A file transfer is initiated by a request which can come from a utility or a TSO user. A remote Partner can also submit a transfer request using compatible software. The list below identifies the utilities that you can use to initiate a transfer request.

- The utility P1B2P101 for a specific PeSIT environment.
- The utilities P1B2PREQ,P1B2PRQ2 or P1B2PRQ3 for other environments. See *SAMPLIB*: $REQ£SIT, $REQ£SCI, $REQSCI2, $REQSCI3.
- The L0B2Z20 module, when called by an application program.
- The %TOMREQ TSO interactive REXX utility.

The parameters that you send with the transfer request are combined with the configuration parameters that you defined in the SYSIN file, the Directories, and tables. After Connect:Express verifies and accepts the transfer request parameters, the request is queued in the Request Control Table (RCT).

Connect:Express provides several features to help you control transfer operations. For example, requests can be scheduled based on request type, Partner type, File type, and available resources. You can also integrate transfer controls into your business processes using automation tools. For example, you can track information about session interruptions and rejected calls, and you can setup Connect:Express to automatically restart transfers. These features are discussed in the sections that follow.

Request Controls

When Connect:Express receives a request, several controls are performed before the request is accepted. The table below describes these controls.

Control	Description
Syntax control	Verifies that all necessary parameters are present: • The symbolic File name, a mandatory request parameter. • The symbolic Partner name, given by the request or derived from the File definition. • The direction of transfer, given by the request or derived from the File definition. • The execution class, given by the request or derived from the Partner definition. • The dynamic DSNAME, given by the request or derived from the File definition. • The priority, given by the request or derived from the File definition. • The link type, given by the request or derived from the Partner definition.
Logic control	Checks that parameters are coherent. Validates the data flow to ensure that the Partner is authorized for the file and transfer direction.
Security control	Checks for authorization to access functions or data sets. This control is in place if the parameter RACFN=Y has been coded in the SYSIN file. It ensures that the Partner is authorized if the request is transmitted by a remote partner, or that the requestor is authorized, if the request is transmitted locally.
File allocation control	Checks the file allocation parameters. The following controls apply when receiving a file: If the Allocation Rule = 0 and the file exists, it is replaced, if it doesn't exist, it is created. If the Allocation Rule = 1, the file must already exist. If the Allocation Rule = 2, the file cannot exist. These controls also verify that the file must already exist when sending a file. When unloading a file before sending or reloading after receiving, it must be possible to allocate the work file (PDS, VSAM, USER, or SYSOUT).

Request Types

The transfer request type defines how the transfer is performed and when the transfer occurs. For example, a transfer can be executed immediately or scheduled for a later time. The table below describes the different request types.

Request Type	Description
N – Normal	This request is executed as soon as possible with no restriction of Partner and/or direction.
H – Hold	This request type is executed by Connect:Express when an incoming inquiry request is received. This facility depends on the transfer protocol used.
I – Inquiry	This request enables you to inquire if a Partner has a file to be sent. If yes, the file will be received. In this case, Partners submit a request with a HOLD status. This facility depends on the transfer protocol used.
U – Unchain	This normal request prevents the transfer from being chained in an existing session.
Serialized	This is used with the P1B2PRQ2 utility. This request type is used with a list of requests and indicates that each request is scheduled one after the other.

The request mode parameter is also used to determine if the request must be scheduled immediately, Mode=I, or deferred, Mode=D. The deferred mode is not valid with a request type of Hold.

Partner and Request Types

When both Partners use the Connect:Express monitor, all requests can be executed without restrictions of type and/or direction. The table below describes how the protocol of the Partner can affect the transfer process.

Protocol	Restrictions
PeSIT D or PeSIT E	A reception request sent by other compatible software must be preceded by a transmission request to Connect:Express with a HOLD status. A reception request sent to another compatible software must be of type **I** (Inquiry).
ETEBAC3	Only Normal and Unchained requests are available. You must define the file to be received on the allocation screen.
ODETTE	Only Normal and Hold transmission requests are available. Using the connection request feature, you can open a connection with a Partner without sending a file and that Partner can send a file to you.
FTP	Normal, Hold, and Unchained requests are available. To receive a file, you must define it on the allocation screen. See the FTP Guide for more information about FTP transfers.

Request Scheduling

When Connect:Express receives a transfer request, it is entered in the Request Control Table (RCT) and assigned a unique 8-character request number. This number is returned to the requestor. Connect:Express creates a backup of the internal tables modified during this operation in the checkpoint file (SYSCHK). The monitor can then be hot-started without incurring any loss of the data being processed.

The execution and scheduling of a request depends on the following controls:

Control	Description
Global Resource Control	The following conditions must be met: • Connect:Express is active. • The Partners control table is enabled. • The Files control table is enabled. • The Requests control table is enabled.
Specialized Resource Control	When executing a transfer, Connect:Express: • Selects an active APM serving the execution class. • Selects a free server in this APM. • Selects an active network resource serving the Partner link.
Individual Resource Control	Additional conditions that must be met include: • The Partner is enabled. • The File is enabled. • The allowed number of links (TOTAL/IN/OUT) for this Partner are available. • The allowed number of links (TOTAL/IN/OUT) in the T1B2PSLD table are available.

File Types and Scheduling

Once a request is scheduled, Connect:Express determines the transfer process based on the file type and the Partner type. The table below describes how the transfer process is affected by file type.

File Description	File Type	Available Request Types
Sequential file (disk or tape)	S or H	Any request type is valid. An incoming or outgoing transmission or reception request executed in a single transfer operation.
Partitioned file	PU	Outgoing Requests • NORMAL TRANSMISSION request – unloading step (by IEBCOPY) on a dynamically allocated file and transfer of the temporary file. Incoming Requests • RECEPTION request – transfer with a dynamically allocated file and reloading onto the PDS (by IEBCOPY).
	P	Any request type is valid. An incoming or outgoing transmission or reception request executed in a single transfer operation.
VSAM file	VU	Outgoing Requests • NORMAL TRANSMISSION request – unloading step (by IDCAMS) on a dynamically allocated file and transfer of the temporary file. Incoming Requests • RECEPTION request – transfer with a dynamically allocated file and reloading onto the VSAM file (by IDCAMS).
	V	Outgoing Requests • NORMAL TRANSMISSION request. Incoming Requests • RECEPTION request.
USER file	UU	Several data sets of various types can be mixed in one transfer. Outgoing Requests • NORMAL TRANSMISSION request – unloading step (by ADRDSSU single/generic) on dynamically allocated files and transfer of the temporary files. Incoming Requests • RECEPTION request – transfer with a dynamically allocated file and reloading onto any files (by ADRDSSU single/generic).
SYSOUT file	SU	Outgoing Requests • NORMAL TRANSMISSION request – with no unload on a dynamically allocated sequential file, or with unload by ISF unload JOB, and transfer of the temporary file. Incoming Requests • RECEPTION request – transfer with a dynamically allocated file and reloading onto the SPOOL with *SYSPRM* member for output options (by IEBGENER).

Checking the Status of a Request

After an outgoing request is queued in the Connect:Express Request Control Table, it receives a status of *Waiting for Selection* or *Disabled*. Connect:Express sets this status, not the user.

As the system administrator, you may need to investigate the status of a transfer and the reason why it has that status. When troubleshooting this problem, you can check the request parameters, the request type, the resource status, and the Partner activity. If you are using the TSO/ISPF interface, you can place the cursor on the request line and press <*ENTER*> to get the current return code. Typically, this gives you the reason for the status.

The following table identifies the types of parameters that you can check and any special considerations.

Parameters	Details
Request Parameters	The following request parameters can affect the status of a transfer request: • The Partner can be disabled or enabled. • The file can be disabled or enabled. • The direction of transfer • The type of Link among Connect:Express resources • The class of transfer among Connect:Express resources • The priority
Request Type	The request type can affect the status of a transfer, especially if the request is a deferred or held request. Request types are listed below. NORMAL request – available for selection if immediate INQUIRY request – available for selection if immediate IMMEDIAT request – available for selection DEFERRED request – waiting to be enabled HOLD request – waiting for external request SERIALIZED request – available for selection if first in the list EXTERNAL request – being processed
Connect:Express Resource Status	All applicable resources for a request must be enabled. Verify that the following resources are enabled. • Connect:Express must be active. • Global resources such as the Partner Control Table must be enabled. • Single resources such as one Partner must be enabled. • Network resources must be active. • The priority and class of a transfer request can also affect the status. For example, there must be a resource available that serves that class.
Partner Activity	The Partner must be enabled and have enough available outgoing links. The session table can indicate that only transfers in the same direction are chained in the same session.

Resources

Transfer operations require a number of Connect:Express and system resources. System resources are accessed by Connect:Express. Typically, you can enable or disable a resource and display its status and activity through the TSO/ISPF or batch interfaces. As Connect:Express activity increases, so does the need for system resources. For example, you should improve disk storage access before increasing the number of possible simultaneous transfers.

This section identifies and describes Connect:Express resources, network resources, and system resources, and includes information about controlling the flow of transfers.

Connect:Express Resources

There are four classes of Connect:Express resources that you manage during transfer operations. The following table describes these four classes.

Resource	Description
INTERNAL	Describes the size of the monitor’s internal working area.

Resource	Description
GLOBAL	Includes resources like the Partners Control Table (PCT), and the Files Control Table (FCT). Global resources are used by all transfers.
SPECIALIZED	Shared by a group of Partners, files, or requests. For example, Connect:Express selects an APM serving a certain class.
INDIVIDUAL	Attached to one Partner, file, or request. For example, one Partner can be enabled or disabled.

All Connect:Express global resources must be active to initiate transfers, and all specialized resources involved in the transfer must be active. Connect:Express will not initiate transfers for disabled partners or files. Connect:Express resources are defined in the directories, the configuration tables, and the SYSIN file, and enable you to control Connect:Express activity by organizing Connect:Express resources.

Note: Directories and configuration tables are described in the User Guide. The SYSIN file is described in Appendix D.

Internal Resources

Connect:Express has defined internal resources such as the maximum number of requests to store in the Request Control Table (RCT). These resources are defined by parameters in the Connect:Express SYSIN file and determine the internal working area of the monitor. Modifications to any of the following parameters must be followed by a cold-start.

Parameter	Description
RQEMAX	Size of the Request Control Table.
MAXSRQ	Maximum number of internal or external IEFSSREQ supported by Connect:Express.

You can use a hot-start when you modify any of the SYSIN parameters listed below.

Parameter	Description
APLNUM	Maximum number of applications which can be connected to Connect:Express.
PCTADD	Maximum number of Partner entries added dynamically in the Partner directory.
FCTADD	Maximum number of file entries added dynamically in the file directory.
STDMSG	Standard network message size. The Maximum is 64 kilobytes, and the default is 4 kilobytes.
MAXEXT	Number and size in kilobytes of the ANM buffer extensions.

Global Resources

The following table lists the global resources, the commands used to implement them, and a short description.

Resource	User Action	Description
Connect:Express	Stop/start Connect:Express activity	NO outgoing transfer will be initiated by Connect:Express if it is STOPPED. Any new incoming call is rejected with a negative network response, Network Return Code (NRC). Local requests are queued in the request control table.
PCT – Partners Control Table	Enable/disable all partners	When the Partners table is disabled, any new incoming call is rejected with a negative network response (NRC), and no new outgoing transfer is initiated. Local requests are queued in the request control table.
FCT – Files Control Table	Enable/disable all files	When the files table is disabled, any new incoming call is rejected with a negative network response (NRC), and no new outgoing transfer is initiated. Local requests are queued in the request control table.
RCT – Request Control Table	Enable/disable the requests	When the requests table is disabled, any new incoming request is rejected with a negative protocol response (PRC), and no new outgoing transfer is initiated. No other local requests are queued in the request control table.

Specialized Resources

The table below lists special resources, the commands used to implement them, and a short description.

Resource	User Action	Description
Transfer resources	Activate/deactivate one APM	Connect:Express can manage 1 to 8 APMs, and each APM can manage 1 to 16 transfer servers (effectors). The APM is activated during Connect:Express initialization. If no APMs are available, any new incoming request is rejected with a negative protocol response (PRC), and no new transfer is initiated. Local requests are queued in the request control table. See *“APM SYSIN Parameters” on page D-10*.
Network resource	Activate/deactivate any ANM handler or X.25 MCH (TCP/IP can only be activated).	All network resources are managed by the ANM. During initialization, Connect:Express starts the ANM with needed and authorized link handlers, and the X.25 MCHs. Each ANM handler and X.25 handler can be activated during Connect:Express initialization. If the needed resource is inactive, no incoming call can be detected on this link, and no new outgoing transfer is initiated on this link. If the Partner is given a list of possible links, the first available link is selected. If no link from the list is available, the local request is queued in the request control table. See *“SYSIN Parameters for the ANM” on page D-5*.
FTP resource	Activate/deactivate AFM	Connect:Express can activate or deactivate the FTP manager (the Auxiliary FTP Manager). The resource parameters of the AFM are defined in the AFM parameter file. See *“AFM SYSIN Parameters” on page D-11*.
SSL resource	Activate/deactivate SSL	Connect:Express can activate or deactivate the SLL handler. See “SSL Parameters” on page D-7.

Resource	User Action	Description
Class Of Transfer	Change the class of transfer list associated with an APM	Each of the APMs must be dedicated to at least one class of transfer (up to seven). The class of transfer for incoming requests is determined by the Partner directory field class of transfer. The class of transfer for an outgoing request is an optional parameter. If this parameter is not given, Connect:Express defines the class for you. If the class of transfer for an outgoing request was not defined in any of the APMs, it is queued but not scheduled. If the class of transfer for an incoming request was not defined in any of the APMs, it is rejected with PRC. See *"APM SYSIN Parameters" on page D-10*.
RACF	N/A	The RACF security interface can be activated during Connect:Express initialization, and can have different configurations. Note that changing an RACF parameter requires a hot-start. An RACF SECURITY user exit is also available. SYSIN parameters: ADHOCN, RACFUD, RACFCN
Global LINK number	N/A	The maximum number of input, output, and total active sessions can be fixed and defined in the T1B2PSLD table. Several Partner definitions can share one T1B2PSLD table entry. When the maximum is reached, the outgoing requests for any one of the Partners are queued, and incoming requests are rejected with a PRC return code.

Individual Resources

The following table lists the individual resources, the commands used to implement them, and a short description.

Resource	User Action	Description
Partner	Enable/disable one partner	If the partner is disabled, all local requests for this partner are queued in the request control table, and incoming requests for this Partner are rejected with a PRC.
File	Enable/disable one file	If the file is disabled, all local requests for this file are queued in the request control table, and incoming requests for this file are rejected with a PRC.
Request	Enable/disable one request	If an error occurs, the request can be automatically disabled and will not be scheduled until you enable it.
PARTNER LINK number	Define session limits for one partner	The maximum number of input, output, and total active sessions is fixed, and defined in the partner directory for each partner. When the maximum is reached, the outgoing requests for this partner are queued and incoming requests are rejected with PRC.
PARTNER X25 charge rule	Declare charge rule applicable for X.25 connections with a Partner	This is processed like a resource (incoming requests only). If the rules of the partners do not match, incoming requests are rejected with a PRC.

Network Resources

Network resources are required to perform file transfers. Some network resources are needed for internal Connect:Express features such as the TOM LOCAL option. Most network resources of Connect:Express interface with the ANM, except GLOBAL-LOCAL Connect:Express connections and FTP connections managed by the AFM.

The Connect:Express SYSIN file contains all the parameters needed to define the ANM environment. These parameters are sent to the ANM in the SYSANM file created by Connect:Express during initialization. The different link handlers are started based on the ANM EXEC parameters set by Connect:Express. Connect:Express selects the handlers to be started depending on the link types set in the partners directory and on the network options provided by the Connect:Express Asset Protection File.

The Auxiliary Network Manager (ANM) resources include link type handlers and X.25 link accesses. Each X.25 link access, or MCH, must be defined by a set of parameters. Link type handlers can be any of the following:

- SNA application-to-application handler
- 3270 handler
- X.25 GATE/DATE/PCNE handlers
- LU6.2 handler
- TCP/IP handler
- SSL handler

System Resources

Transfer operations require system resources for file management, system security, storage management, and performance management. Using and optimizing these resources is discussed in this chapter and chapter 6. References about OS/390 resources are listed in the following table.

Resource	Reference
ALLOCATION	“Allocating Files” on page 3-20
SECURITY	“System Security” on page 3-33
STORAGE	“Storage Estimates” on page 6-12
PERFORMANCE	“System Performance Parameters” on page 6-11

SYSIN Resource Fields

Some of the SYSIN fields are used to define the internal, global, and some of the specialized resources. Some of these fields are mandatory and some are optional. Using these parameters, you can optimize Connect:Express performance. Note that changing some of the parameters may require a cold-start. The following tables describe the required parameters, X.25, TCP/IP, and optional parameters.

Required Parameters

The following parameters are required. These are set during installation of Connect:Express but can be changed later.

Field	Description
ADHOCN	If ADHOCN is set to **Y**, AD HOC requests are allowed with mandatory RACF security. If ADHOCN is set to **N**, AD HOC requests are not permitted. If ADHOCN is set to **U**, AD HOC requests are allowed with optional RACF security.
ANMPRC	The name of the start procedure for the Auxiliary Network Manager (ANM).
APLNUM	Indicates the number of applications that can be connected to the monitor.
APLPFX	A six-digit prefix of VTAM application names used by the ANM.
APMPRC	The name of the start procedure for the Auxiliary Protocol Manager (APM).
DAPM01 01 to 08	Defines the transfer resources. DAPMxx=(E/nn/xyz) DAPMxx, *xx* can be 01 to 08. The first field *E* can be one of the following values: E – APM must be started during initialization H – APM cannot be started during initialization, but can be submitted later by command. O – APM is disabled and cannot be submitted. The second field, *nn*, is the number of protocol servers in this APM. Valid values are 01 to 16. The third field, *xyz*, defines transfer classes. Valid values are ABCDEFG.
FCTADD	This value represents the maximum number of new entries in the File directory sent to Connect:Express in real-time. If FCTADD=0, updates are prohibited.
MAXSRQ	The maximum number of IEFSSREQ issued simultaneously by TSO users or batch JOBs using the API, (L0B2Z20). If changed, this field requires a cold-start, RUN=C.
PCTADD	This value represents the maximum number of new entries in the Partner directory sent to Connect:Express in real-time. If PCTADD=0, updates are prohibited.
RACFCN	If RACFCN is set to **Y**, Connect:Express controls local or remote user RACF authorization to transmit requests. If RACFCN is set to **N**, Connect:Express does not check the RACF origin of the request. If RACFCN is set to **S**, Connect:Express checks user authorization for RACROUTE-SAF for compatibility with products such as ACF2 and TOP-SECRET.
RMFLOG	If RMFLOG is set to **Y**, Connect:Express sends RMF records. This can be used to measure Connect:Express system utilization. If RMFLOG is set to **N**, Connect:Express does not produce RMF records.
RQEMAX	Represents the maximum number of requests in the request control table. This value includes the possible number of requests pending, those in progress, and those which have been interrupted. The requests which have been successfully executed are deleted from the RCT. The maximum recommended number of authorized requests is 1024. If changed, this field requires a cold-start, RUN=C.
SMFREC	This is the user SMF record number assigned to Connect:Express. Enter 000 if you do not want these records.

X.25 Parameters

X.25 parameters are mandatory if MCHNBR is greater than zero. Each MCHNAM has subparameters which must be defined. The table below identifies these subparameters.

Field	Description
MCHNAM	The line macro name defining the MCH in the NCP.
MCHNBR	The number of multi-channel lines dedicated to the ANM. The maximum is 32.
MCHLUN = luname	The MCH LU name defined in X25NPSI.
MCHWDS	Indicates the size of the X.25 packet window size for this subscription.
MCHVCN	Sets the number of generated switched circuits for this MCH.
MCHPKS	Sets the size of the X.25 packets for this subscription.
MCHTYP	Indicates the type of MCH (D=DATE, G=GATE).
MCHRTR	Indicates the occurrences and time of the MCH reactivation (in 30 second increments) which is done automatically if the MCH is lost.
MCHMSC	Provides the method of grouping MCHs by identities, or separating the MCHs.
MCHXLA	The optional X.25 local address of the MCH.

TCP/IP Parameters

The following TCP/IP parameters are mandatory if TCP/IP is used.

Field	Description
TCPORG	Defines the origin of TCP/IP.
TCPPRT	Defines the TCP/IP port number on which the ANM listens for calls.

Optional Parameters

The following parameters are optional.

Field	Description
DAPM02	The APM is not started during Connect:Express initialization, but can be started later by Connect:Express using an operator command.
DAPM03 to DAPM08	These are keys to use additional APMs and must be licensed.
STDMSG	The standard network message size. The maximum is 64, and the default is 4 Kbytes.
MAXEXT	The number and size in KBytes of ANM buffer extensions.
AFMPRC	Name of the start procedures for the Auxiliary FTP Manager.
AFMCLS	Definition of the FTP Session.

Field	Description
RACFUD=username	The RACF-USER by default. If no RACF USER was set in the Partner directory, the RACFUD value is used for security verification. If the ADHOCN parameter is set to unsafe, the RACFUD value is used for security verification for incoming AD HOC transfer requests.

Controlling the Flow of Transfers

To control the flow of file transfers, you can define Partner activity, conditions for file exchange, or control transfer requests by scheduling the transfer or specifying the transfer class. For example, you can enable or disable Connect:Express resources like the APM or change the class managed by an APM. The table below identifies how you can control activity for Partners, Files, and requests.

Control type	Description
Partner activity	You can limit one Partner's number of input/output sessions, or limit a group of Partners' number of input/output sessions. These are defined globally in the Session Link Definition (SLD) table. You can also direct incoming demands from a partner using the class of transfer, or direct outgoing calls to the Partner using MCHMSC (X.25) and LOGMODE (SNA) parameters.
File activity	You can control file activity by setting the direction for the file transfer, and limiting the Partners who can receive or transmit the file.
Transfer activity	You can defer the time of transfer until it is enabled, or schedule the transfer for a specific date and time such as, Monday at 13:10 hours. You can also direct outgoing calls to the Partner by using the class of transfer.

Automation Tools

Automation facilities enable you to integrate file transfer operations into your business processes. Using these features with a network manager system like NETVIEW can provide the most efficient use of network resources. This section includes the following topics:

- Checking the status of requests and automating transfer requests using modify commands, batch utilities, programs, user procedures, and user exits
- Controlling connections with the L1GFICN1 utility
- Checking the status of requests
- Restarting transfers automatically

There is also an optional Utilities package with tools to help you implement automation. See the Utilities Guide for more information.

Automating Transfer Requests

Automation is used for reliability and includes tools for checking the monitor resources, network resources, and system resources. Automation is also used for file transfer execution and includes scheduling, checking, controlling, and restarting requests. It can also include automatic notifications about each step in the file transfer process. Connect:Express provides the following automation facilities:

Facility	Description
Modify commands	You can send Modify commands to the monitor to activate or deactivate monitor and network resources. See Appendix C for a list of monitor commands.
Batch utilities and API	You can call batch utilities within the JCL to request a transfer. The API can also be called from a user program to request a file transfer and to monitor transfer operations. Monitoring includes enabling/disabling monitor resources and displaying the transfer operation and resource status. See Chapter 4, *Monitor Management* and Chapter 3, *Transfer Requests* in the User Guide for more information.
User procedures	User procedures started by Connect:Express enable you to link transfer operations with other operations or procedures within your business. The monitor can start user procedures at the different times shown below: • Connect:Express initialization (SYSIN UPRCPI= field) • Connect:Express termination (SYSIN UPRABE= and UPREND= fields) • Initialization of transfer (FILE directory) • Termination of transfer (FILE directory) depending on UPRFCT= option User procedures can also be connected to Connect:Express. These procedures enable you to receive notifications in journal records with all the relevant information about a transfer so you can quickly glance at the journal records and determine the status or actions that need to be taken. See Chapter 4, *Monitor Management* in the User Guide for more information about the journal file.
User exits	User exits are programs written by the user that are loaded and activated by Connect:Express. When Connect:Express is running, control can be handed over to the user program and then returned to Connect:Express. This enables you to look at the operations and make decisions about file transfer operations, or halt transfer operations, analyze files, and take corrective action if necessary. See Chapter 2 of this book for more information about user exits.

Controlling Connections with the L1GFICN1 Utility

A connection user exit gives online information about rejected incoming and outgoing calls and session interruptions for the user. L1GFICN1 is the connection user exit that can help you track this information. This module issues WTO messages only if there are errors.

When an incoming call is rejected by Connect:Express, no information is stored in the journal file. Instead, the incoming network parameters (D1B2PCNX data structure in the *MACLIB*) are sent to the connection exit. The screen below shows the WTO message with the incoming call error sent by the L1GFICN1 utility.

```
TOM3CN1 ERROR PARTNER1 INP-FAIL 00000051 TRC2151 PRC0000 NRC000000 S/000000
SNA input connection rejected because network identification is invalid.
```

When an outgoing call is rejected by the remote Partner, information is stored in the journal file unless the retry connection procedure is active. The outgoing network parameters (D1B2PCNX data structure in the *MACLIB*) are sent to the connection exit. The following screen displays the WTO message with the outgoing call error sent by the L1GFICN1 utility.

```
TOM3CN1 ERROR PARTNER1 OUT-FAIL 00000052 TRC0000 PRC0000 NRC010DF1 X/000000
X.25 output connection rejected because network identification is invalid.
```

The connection exit receives the D1B2PCNX data structure at the end of the session. The screen below shows the WTO message with a session interruption error sent by the L1GFICN1 utility.

```
TOM3CN1 ERROR PARTNER1 INP-CLOS 00000060 TRC0000 PRC3317 NRC000000 S/000000
X.25 input session interrupted because timer expired.
```

Note: A session can be interrupted after a successful file transfer, and a file transfer can be interrupted without a session error. A transfer failure can only be detected at the end of the transfer.

Controlling Selection with the L1GFICN1 Utility

The connection user exit described above can also be used for controlling file transfer selections. This exit gives online information about rejected incoming and outgoing transfer requests and transfer interruptions. This module only issues WTO messages if there are errors, and must be defined in the T1APMSRT table (TSO/ISPF 3.3.SRT option).

When an incoming transfer request is rejected by Connect:Express, no information is stored in the journal file. Instead, the incoming transfer request parameters (D1B2RUEX data structure in the *MACLIB*) are sent to the selection exit. The screen below shows the WTO message with the incoming transfer request error sent by the L1GFICN1 utility.

```
TOM3CN1 ERROR PARTNER1 INP-FAIL 00000051 TRC0000L PRC204  SRC000000
File transfer input is rejected because the file already exists.
```

When an outgoing call is rejected by the remote Partner, information is stored in the journal file unless the retry selection procedure is active. The outgoing transfer request parameters (D1B2RUEX data structure in the *MACLIB*) are sent to the selection exit. The following screen displays the WTO message with the outgoing transfer request error sent by the L1GFICN1 utility.

```
TOM3CN1 ERROR PARTNER1 OUT-FAIL 00000052 TRC0000L PRC204  SRC000000
File transfer output is rejected because the file already exists.
```

The selection exit receives the D1B2RUEX data structure at the end of the transfer. The screen below shows the WTO message with a transfer interruption error sent by the L1GFICN1 utility.

```
TOM3CN1 ERROR PARTNER1 INP-FAIL 00000060 TRC0000 PRC216 SRC000000
File transfer input has been interrupted by operator.
```

Restarting Transfers

The current activity of Connect:Express is automatically saved so you can restart Connect:Express if any failure occurs. Connect:Express does this with a monitor checkpoint file that can be mirrored. The current status of all the transfer requests, partners, files, and transfer resources (APM) are recorded in this file. Connect:Express also allocates a Recovery file which registers all transfer requests and their status when the monitor is not up. Each transfer allocates a request checkpoint file that has a data set name built by the APM. One index is the unique transfer request number. The current status and counts of the transfer are recorded in this file. This request checkpoint file is purged when the corresponding request is purged.

A failure can be a monitor stop, a transfer interruption, or a connection error. If the monitor stops, all current transfers are interrupted, and the monitor checkpoint file and recovery file are used. If the transfer stops, a transfer request checkpoint is used. If a connection or transfer negotiation fails, no request checkpoint is used.

A transfer can be interrupted by an operator, a system error (SRC=), a network error (NRC=), a protocol error (PRC=), a monitor decision (TRC=), or the termination of the monitor. This information is found in the following locations:

- The Journal record
- The end of transfer user exit parmlist
- The Request Control Table

Automatically Restarting Transfers

Only the requestor can restart a transfer. If you are the initiator of a transfer, this is called a local request of transfer. A remote transfer request is received from the network and cannot be restarted locally.

You can restart a local transfer request using the API or batch utility, the TSO/ISPF interface, or by implementing the T1B2PCOD table. You can enable Connect:Express to automatically restart transfers based on return codes by writing a program or using a batch utility that looks for a certain return code and then automatically restarts the transfer. Return codes are found in the journal record, the user exit parmlist, and the Request Control Table.

After the request is restarted, the current status of the transfer is restored from the request checkpoint file. The following table lists the Connect:Express parameters that are involved in restarting transfers. The SYSMSG file of Connect:Express is used during initialization to issue warnings and information messages about the loading of the checkpoint and recovery file.

From	Field	Description	Report
Connect:Express JCL	SYSCHK SYSCHK2	The monitor checkpoint file contains the current status of the control tables RCT, FCT, PCT, APM.	SYSMSG
Connect:Express JCL	SYSRCY	The recovery files contain all the transfer requests that have been recorded when Connect:Express was not up.	SYSMSG
APM JCL	CHKMODEL	This file is a model for the transfer request checkpoint allocation. The APM gets the volume number where the allocation must be done, so you can direct the files to the appropriate disk. Keywords are provided to you to group the files: &APMCKI1, &SSN, &APM.	
Installation procedure	&APMCKI1	First index for all transfer request checkpoint files.	
APM JCL	&SSN	Subsystem Name used to identify all checkpoint files for one subsystem (Connect:Express).	

From	Field	Description	Report
APM JCL	&APM	APM number that identifies all checkpoint files for one APM. This index only applies to the model file. It is neutralized by the constant value 'CHKP' so the transfer can be restarted in any APM.	

The Retry Process

The first step in processing a transfer request is connecting to the remote Partner, then you must select and agree upon the file to be transferred. If a failure occurs, the monitor will automatically retry, but this depends on the type of error. When the retry is not automatic, the transfer request is disabled and you must manually enable it to try again. Typically you can ask the monitor to retry every n minutes for x retries. A failure can be any of the following:

Failure type	Description
Network failure	The current link is not available and a session is 'NOT OBTAINED.'
Protocol reject	The Partner rejects the connection and the session is 'POSTPONED' OR the Partner rejects the transfer and the request is 'POSTPONED.'

You can define several link types for one partner, for example SNA + X25. On each link you can enter a list of alternate addresses. Connect:Express processes connection network failures in the following manner:

1. x retries every n minutes on the first link, first address
2. x retries every n minutes on the first link, second address
3. ….
4. Change link
5. x retries every n minutes on the second link, first address
6. x retries every n minutes on the second link, second address
7. ….

You can allow this process to continue until all links and addresses have been tried, and then stop or restart it again indefinitely. If the process is stopped, the Partner and the transfer request are disabled, and you must enable the Partner and the request before it can be retried.

You can limit the automatic retry to a list of Protocol Return Codes (PRC) in the T1B2PCOD table. The link types for the Partner are defined in the Partners directory, and the list of alternate addresses are defined in the SYSX25, SYSSNA, and SYSTCP files. You can also limit the scanning of alternate links and addresses to one turn by setting Restart=No in the Partners directory.

Retry Parameters

The table below identifies the file location of retry parameters including specific fields and logs used.

File	Field	Description	Report	Field
SYSIN	STIMEV	Timer values, first for connection, second for selection		
	STIMOC	Retry occurrences	Journal	Z45REQRS

File	Field	Description	Report	Field
T1B2PCOD	Protocol, PRC	List of return codes for this protocol	Journal D1B2RUEX	Z45PRC UEXPRC
DIRECTORY	LINK	One link or mixed link with a list	Journal D1B2RUEX	Z45PARLK UEXLNKTP
	ADDRESS	The first address processed		
	RESTART	If YES process is continued indefinitely		
SYSX25	PARTNER	Alternate addresses		
SYSSNA	PARTNER	Alternate addresses		
SYSTCP	PARTNER	Alternate addresses		

Using T1B2PCOD

Automatic retry is activated when a network connection fails, a session negotiation with the partner fails, or a file transfer negotiation fails. For network connection failures, the retry process is always activated. For transfer negotiations, the retry process depends on the protocol return code. Session negotiations with a Partner and file transfer negotiations can also be rejected. If it is rejected, the Partner issues a protocol return code (PRC). The T1B2PCOD table is used to control automatic session retry or transfer selection retry according to the transfer protocol return codes. The following table describes what happens during a session failure and a transfer selection failure.

Retry Type	Description
Session	In case an outgoing call is rejected by a Partner, the PRC received is compared to the PRC stored in this table and if a match is found, Connect:Express must activate the connection retry procedure using the SYSIN file parameters STIMOC=y, STIMEV=(n,.).
Transfer Selection	If an outgoing transfer selection error occurs, the PRC received is compared to the PRC stored in this table and if a match is found, Connect:Express must activate the transfer selection retry procedure using the SYSIN file parameters STIMOC=y,STIMEV=(.,m).

If a PRC code is not found in the table, the following message displays in the Connect:Express log file.

```
REQUEST 00000001 FILETEST PARTNER SRC=0000 TRC=0000 PRC=0000
REQUEST 00000001 <- TSO00001 ACCEPTED (N)
COMMUNICATION OPENED (O) WITH: PARTNER REQ: 00000001 (X,MCHA,LUSMN08)
REQUEST 00000001 NO AUTOMATIC RESTART BY T1B2PCOD, PRC=304 PESIT
REQUEST 00000001 FILETEST PARTNER REJECTED SRC=0000 TRC=0000 PRC=3304
REQUEST 00000001 <- TSO00001 REJECTED
COMMUNICATION CLOSED (O)  WITH PARTNER
```

Files

Connect:Express can process PDS, VSAM, SYSOUT, and sequential files from disk or tape media. It can also process user structure files using the ADRDSSU utility, for example. Depending on your file organization, you should consider the remote platform and software, transfer protocol, and transfer direction. For example, you cannot transfer PDS files with the ODETTE protocol.

Since transfers take place between different systems, protocols carry only basic file attributes such as record size and record format. When communicating within homogeneous environments, specific attributes can be exchanged, like the physical data set name and physical structure. With Connect:Express, Partners can exchange such attributes through private protocol fields.

TOM, the APM, and the EAS address spaces manage allocation operations. The APM and EAS manage file input and output commands like file open/close and read/write. You can also use user exits to control file allocation. For example, you can use a user exit at the beginning of the transfer or a user exit that is called during the transfer process to control file allocation. The file allocation process involves:

- Naming the files
- Preparing files for transfer
- Allocating files

This section describes the file allocation process and also includes information about Connect:Express workfiles and checkpoint files.

Naming the Files

The first part of the allocation process is naming the files to transfer. Standard keywords are used that enable you to dynamically compute the data set name from your local environment (&REQNUMB and &REQTIM, for example) or from your Partner's information (&EXTDSN and &EXTLAB, for example). Some keywords are only available for reception, while others are available for both directions. For transmission, the only keywords supported are &I, &PARTNID, &FILENAM, &ORG, and &DST. All keywords are supported for reception.

Note: When the file type includes an unload/reload processing, "name of the file" applies to the file sent or received, not to the workfiles used for transferring the data.

Preparing Files for Transfer

To transfer a file of type VSAM/UNLOAD, the VSAM file is unloaded first, and then sent. The receiver will receive it, and then reload it. In this case, you not only transfer the file, you have to unload and reload it. The table below lists different conditions for a file transfer and any special action needed.

Organization	File Type	Software	System	Action
Sequential DISK	S	All	All	No special process is needed.
Sequential TAPE	M	Connect:Express	OS/390	Tape to tape Tape to disk Disk to tape
PDS	P	Connect:Express	OS/390	Direct transfer, normal dynamic allocation on reception side.
PDS/unload	PU (UNLOAD)	All	OS/390	Reload process on reception side. Only outgoing transmission/incoming reception.
VSAM	V	Connect:Express	OS/390	Direct transfer, PREALLOCATION on reception side. Only outgoing transmission/incoming reception.
VSAM/unload	VU (UNLOAD)	All	OS/390	Reload process on reception side. Only outgoing transmission/incoming reception.
SYSOUT	S	All	OS/390	The SYSOUT is copied in a sequential file.

Organization	File Type	Software	System	Action
SYSOUT/unload	SU (UNLOAD)	All	OS/390	Reload process on reception side. Only outgoing transmission/incoming reception.
USER/unload	UU (UNLOAD)	All	All	Use depends on the process, RELOAD on the receive side and UNLOAD on the send side. Use is only outgoing transmission/incoming reception.

The following table describes how files are transferred based on file type.

File Type	Processing Description
Sequential disk files	Sequential files can be processed in two modes, block mode and record mode. In block mode, all the records in the whole block are processed as one unit, then sent. Block mode performance is better, and is only used in Connect:Express to Connect:Express environments. Block Mode is set by an option in the presentation table. In record mode, each record in the block is processed separately into a network message, then sent. Record mode is supported by any partner and any protocol and is the default transfer method.
Tape files	Generally, tape files can only be exchanged between Connect:Express OS/390 partners. The exception is when the record format is undefined. In this case, the file transfer takes place in “Record Mode” so any Partner and protocol are supported.
PDS files	There are two transfer methods, Unload/Reload and Direct: With **Unload/Reload,** the file is transmitted as a VBS file after automatic UNLOAD, and automatically RELOADED after reception. To use the UNLOAD/RELOAD procedure, the following conditions must be met. • The file must be defined with TYPE=PU. • The $JOBREL1 and $JOBUNL1 JCL files in the SYSJCL library must be customized. • If you want to select members from the PDS file during unload operations, you must create a selection member in the SYSPRM library, then enter its name in the UNLOAD/RELOAD member field in the symbolic file definition. Using this information, Connect:Express will automatically build the UNLOAD job, submit it, and wait for the result. Only outgoing transmission and incoming reception are available. With **Direct transfers**, the file and structure are sent. Transmission and reception can be incoming and outgoing only with Connect:Express OS/390 Partners.
VSAM files	There are two transfer methods, Unload/Reload and Direct: With **Unload/Reload**, the file is transmitted as a VBS file after automatic UNLOAD, and automatically RELOADED after reception. To use the UNLOAD/RELOAD procedure, the following conditions must be met. • The file must be defined with TYPE=VU. • The $JOBREL2 and $JOBUNL2 JCL files in the SYSJCL library must be customized. • To receive the file, you must create a DELETE/DEFINE member in the SYSPRM library, then enter its name in the UNLOAD/RELOAD member field in the symbolic file definition. Using this information, Connect:Express will automatically build the RELOAD job, submit it, and wait for the result. Only outgoing transmission and incoming reception are available. With **Direct transfers**, the file and structure are sent. Only outgoing transmission and incoming reception with Connect:Express OS/390 partners is available and the file must be pre-allocated on the receiver side.

File Type	Processing Description
SYSOUT files	There are two transfer methods, Unload/Reload and Direct: **With Unload/Reload**, the file is transmitted as a VBS file after automatic UNLOAD, and automatically RELOADED after reception. To use the UNLOAD/RELOAD procedure, the following conditions must be met. • The file must be defined with TYPE=SU. • The $JOBREL4 and $JOBUNL4 JCL files in the SYSJCL library must be customized. To send the file, the SYSOUT is unloaded by ISF. To receive the file, you must create a member in the SYSPRM library that will provide output options to IEBGENER. Then, give its name in the UNLOAD/RELOAD member field in the symbolic file definition. Using this information, Connect:Express will automatically build the RELOAD job, submit it, and wait for the result. Only outgoing transmission and incoming reception are available. With **Direct transfers**, the SYSOUT is copied to a file and sent as a Sequential file.
USER files	USER files can have any special structure defined by the user that is processed with a UNLOAD/RELOAD mechanism. For example, it can be a group of files that have to be sent in a bundle or any data base extraction. USER files can only be transferred using the Unload/Reload method as described below. The file is transmitted as a VBS file after automatic UNLOAD, and automatically RELOADED after reception. To use the UNLOAD/RELOAD procedure the following conditions must be met. • The file must be defined with TYPE=UU. • The $JOBREL3 and $JOBUNL3 JCL files in the SYSJCL library must be customized. To send the file, the user procedure is started to build the VBS file to be sent. On receiving the file, the reload procedure is started as defined by the user after the VBS file is transmitted. Connect:Express will automatically start the UNLOAD or RELOAD procedure and wait for the result before sending the file or notification that the file has been received. Only outgoing transmission and incoming reception are available. Zipped files can also be transferred this way.

Allocating Files

The file allocation process is more involved when receiving files than sending them. On transmission, you only need to name the file and have access to it. When receiving files, you must set up parameters for allocation in the symbolic file definition. Some of the parameters are mandatory depending on the file transfer protocol that you are using. With some protocols, Connect:Express receives the parameters for the file before receiving the file. Refer to the User Guide for information about allocating files in an SMS environment.

Connect:Express typically manages file allocation, but it can also be performed by a user exit which is defined in the symbolic file parameters. Connect:Express provides the following file allocation features:

Feature	Description
Allocation of catalogued and multi-volume files	Connect:Express can allocate both catalogued and multi-volume file types.
Rotation on a pool of volumes	You can define a group of disks on which you want Connect:Express to allocate a file.
Retention and expiration date processing	You can allocate files with a retention date or an expiration date. With a retention date, the file is kept for a specified number of days. With an expiration date, the file is kept until a specified date.
Exchange of allocation parameters between Partners	By exchanging allocation parameters between Partners, the sender can send parameters to the receiver so that the file will be exactly the same for the receiver.

Connect:Express typically manages file I/O, but it can also be performed by an I/O user exit which is defined in a presentation table. The Presentation Protocol field points to the presentation table from the File Attributes screen.

Connect:Express Workfiles

Connect:Express uses temporary files for the following processes:

Process	Description
Checkpoint processing	Connect:Express creates temporary checkpoint files during transfer execution. These files are built from the CHKMODEL allocated in the APM procedure.
IEBCOPY procedures	The SYSIN file of IEBCOPY is a temporary file. This is created when you select members for a PDS Unload/Reload file.
Allocating work files during transfers	Allocation is based on the WRKUNT field in the SYSIN file. If WRKUNT=SMS, the unit name and volume-serial are provided by SMS. The temporary data set name prefix is the MAJIND field in the SYSIN file. Connect:Express allocates temporary files for the following: • PDS unload/reload workfile • VSAM unload/reload workfile • SYSOUT unload/reload workfile • Testing the device type before creating a new file (reception with SMSSDB=N): A temporary file is allocated by the monitor if it is asked to create a new file (FCT ALLOCATION RULE is 2 or 0) and if the allocation is not done under SMS control (SMSSDB=N in the SYSIN file). The temporary file's DDNAME has the format Wx, where x is the request number, and its allocation disposition is (NEW,CATLG,CATLG). Then the file is deleted by UNALLOC (OLD,DELETE). The purpose is to get the track size to allocate the space based on the Kbyte count specified. Two situations can occur: • If a unit or volser is specified in this FCT entry, the dsn (up to 35 chars) with '.W*request-number*' as last index and volser/unit are used to allocate this temporary file, according to SMS usage. • If no unit/volser is specified in this FCT entry, the MAJIND of *SYSIN* is used with '.W*request-number*' as the last index and WRKUNT unit specification are used to allocate this temporary file, according to user/SMS defaults. NOTE: The temporary file will not be allocated if the Allocation Rule is 2 or 0 and SMSSDB=Y because SMS will manage block size and space calculation.
Some TSO/ISPF functions	MSG, EDIT, LIST, and CMD involve workfile allocation. The dsn structure is shown below: • If SYSPREF=SYSUID Index1 = SYSUID • If SYSPREF not equal SYSUID Index1 = SYSPREF and Index2 = SYSUID, 'Clist/Program' and 'Function' name. NOTE: Using option 0 from the main menu, you can set the work unit to any unit allowed or leave it blank.

Checkpoint Files

Connect:Express is designed to be in service 24 hours-a-day, 7 days-a-week and to support file transfer restarts. To accomplish this, Connect:Express provides checkpoint and recovery files to ensure data integrity in the event of a transfer interruption, a system failure or other event that requires you to restart Connect:Express. The Monitor checkpoint and recovery files record the general status of Connect:Express activity, and the current status of each transfer request is recorded in its own checkpoint file.

The Monitor Checkpoint File and Recovery File

When Connect:Express hot-starts (RUN=H), the request control table is filled in with all interrupted requests recorded in the Connect:Express checkpoint file. Then, the request control table gets filled in with all requests recorded in the Connect:Express recovery file.

If a Connect:Express cold-start (RUN=C) is needed, you should save the checkpoint file and the recovery file and restore them after Connect:Express is started. To do this, complete the following steps:

1. STOP Connect:Express
2. RUN the job for unload: $CKOFLOD part 1
3. RUN the job for reload: $RCOFLOD part 1
4. START Connect:Express, RUN=COLD
5. RUN the job for unload: $CKOFLOD part 2
6. RUN the job for reload: $RCOFLOD part 2

The checkpoint file can be allocated and formatted from the punch file, stage 2 ($STGSMP2) of the installation procedure. It uses the steps ATOMCHK for allocation, and ITOMCHK for initialization from $STGSMP2. If you format an existing checkpoint file, all requests are lost. You may need to allocate a new checkpoint file to allocate a mirror, duplicate the checkpoint file, change the format, or when a checkpoint file gets lost or corrupted. Examples of the two files, $CKOFLOD/$RCOFLOD, are provided in the Connect:Express *SAMPLIB*.

Note: Using the SYSTEM CANCEL command can cause data to be lost in the checkpoint file.

The Transfer Request Checkpoint File

The transfer request checkpoint file is a temporary sequential file allocated by the APM from a checkpoint model file. In the Data Set Control Block (DSCB) of the model information, the APM is told the format of the file and where to allocate the current checkpoint file. The transfer request file is created before the beginning of transfer and is deleted when the transfer has ended successfully or when an operator purges it. The data set name is made from the request number used for the last index.

The checkpoint model file is allocated by user: DSORG=PS,RECFM=FS,LRECL=8192,BLKSIZE=8192. The data set name of this template file is built from index keywords that make it possible to distribute the different transfer request checkpoints to different devices. You can have each APM allocate these temporary files on a special disk. The following screen shows an example of the transfer request checkpoint data set name.

```
DD CHKMODEL DSN=CHKIND1.&SSN.AP&APM..CHKMODEL is resolved this way:

Monitor Connect:Express TOM1 running with four APMs

DD CHKMODEL DSN=CHKIND1.TOM1AP01.CHKMODEL is pre-allocated during installation
DD CHKMODEL DSN=CHKIND1.TOM1AP02.CHKMODEL
DD CHKMODEL DSN=CHKIND1.TOM1AP03.CHKMODEL
DD CHKMODEL DSN=CHKIND1.TOM1AP04.CHKMODEL

Monitor Connect:Express TOM2 running with two APMs

DD CHKMODEL DSN=CHKIND1.TOM2AP01.CHKMODEL
DD CHKMODEL DSN=CHKIND1.TOM2AP02.CHKMODEL
```

Security

Security in Connect:Express involves managing identification of Partners and files and controlling access to Connect:Express features. This section describes the identification parameters used in Connect:Express. It also outlines the security process and lists the parameters that you can use to setup security in your business environment.

Identification Parameters

Connect:Express must be able to identify who Partners are, what files to transfer, what method or protocol to use, and how to notify you that the transfer has taken place. This is done on the following four levels:

Level	Description
Application	This is the requestor of the transfer. The application processes the data set name and extended identification for the Partner and the file.
Monitor	The monitor executes the transfer and processes the data set name, network address, and symbolic identification for the Partner and file.
Remote Partner	This is the receiving Partner. The remote Partner processes the data set name and the symbolic and extended identification for the Partner and file.
System	The system controls access to the files and processes the data set name and local user identification.

The tables in this section identify the parameters that Connect:Express uses during the identification process. The tables are organized by function. The first two columns identify where Connect:Express gets the transfer information. The third column provides a description, and the last two columns list the reports and fields where Connect:Express logs information about the transfer.

File Identification

The table below shows how file identification is processed and lists the parameters and logs that Connect:Express uses.

From	Field	Description	Report	Field
DIRECTORY	SYMBOLIC NAME	The symbolic name identifying the file and profile of the transfer. It must be negotiated with the Partners involved *before* running the transfer. This name must correspond to an entry in the remote Connect:Express directory.	JOURNAL SMF	Z45FILEN Z07FILEN
P1B2PRQ2 parmlist P1B2PRQ3 P1B2PREQ REQUEST	SFN=	The symbolic name of the file to be transferred.	JOURNAL SMF	Z45FILEN Z07FILEN
DIRECTORY P1B2PRQ2 parmlist P1B2PRQ3 P1B2PREQ REQUEST	DSNAME DSN= DSR=	The local identification of the file. It can be sent to the Partner using protocol fields.	JOURNAL SMF	Z45DSNAM Z07DSNAM
P1B2PRQ2 parmlist REQUEST	API= A48= A34= SYSAPI82	The information that the application uses for identifying the transfer (other than Connect:Express). Most protocols use a structure of ORIGIN+DESTINATION+NAME+DATE. Connect:Express moves this information from user to user and stores it in the journal.	JOURNAL SYSLOG	Z45TAPID API_CREATE
D1B2RUEX	UEXDDNM=	The symbolic file name exchanged between Connect:Express and a user exit.	JOURNAL SMF	Z45FILEN Z07FILEN

From	Field	Description	Report	Field
	UEXDSNM=	The physical data set name exchanged between Connect:Express and a user exit.	JOURNAL	Z45DSNAM
			SMF	Z07DSNAM
	UEXAPIXX=	The application file identification exchanged between Connect:Express and a user exit.	JOURNAL	Z45TAPID

Partner Identification

The following tables show how Partner identification is processed and lists the parameters and logs that Connect:Express uses. There is one table for remote system identification, and one for local system identificaiton.

Remote System Identification

From	Field	Description	Report	Field
DIRECTORY	SYMB. NAME	A symbolic name identifying a remote system or a session profile. It is associated with a network address and identifies the partner. It enables outgoing calls and incoming calls with verification. It must be negotiated with the Partner before running the transfer. If it is not associated with a network address, it identifies a partner profile and assumes the user is in charge of physical identification.	JOURNAL	Z45PARTN
			SMF	Z07PARTN
	ODETTE IDENTIF.	Application name for the remote ODETTE-FTP site. When there is an incoming call, the name is verified. If no API parameter is given at request time, the symbolic identification is used as the default in the origin and destination fields.	D1B2PCNX	
SYSX25	SYMB. PARTNER	The partner name defined in the directory that has an alternate X25 link defined by the parameters associated with the symbolic partner. When attempting to make an outgoing connection, the directory link is used. If there are problems, then the SYSX25 links will be used to attempt the connection. When an incoming connection is attempted, the caller's link is verified first from the directory, then using the directory link, then using the SYSX25 links.	JOURNAL	Z45PART
			SMF	Z07PARTN
	Remote address	One of the network addresses that the partner can be connected with.		
	Local address	Set in the outgoing X.25 call packet.		
	CHARGE	Set in the outgoing X.25 call packet. Enables/disables an incoming call to have the charge reversed.		
	CUG	Call User Group, set in the outgoing X.25 call packet.		
	UDF	User Data Field, set in the outgoing X.25 call packet.		
	MCHSEL	The local identification of the X.25 network through which the partner is linked.		
	X25BYPAS XPAD	The standard key used for general PAD incoming connections. If the calling partner can't be found in the directory, the one associated with that key will be used. The Calling partner is the requestor.	JOURNAL	Z45PARTN
			SMF	Z07PARTN

From	Field	Description	Report	Field
	X25BYPAS ETB3	The standard key used for general ETEBAC3 incoming connections. If the calling partner network address can't be found in the directory, the symbolic partner associated with that key is used. This symbolic partner must be defined with TYPE=ETEBAC3, protocol 4.	JOURNAL SMF	Z45PARTN Z07PARTN
SYSSNA	SYMB. PARTNER	The partner name defined in the directory that has an alternate SNA link defined by the parameters associated with the symbolic partner. When attempting to make an outgoing connection, the directory link is used. If there are problems, then the SYSSNA links will be used to attempt the connection. When an incoming connection is attempted, the caller's link is verified first from the directory, then using the directory link, then using the SYSSNA links.	JOURNAL SMF	Z45PARTN Z07PARTN
	LUNAME	One of the LUNAMES that the partner can use for communications. When making an outgoing call, the list of lunames is scanned.	SYSLOG	
	SNA LOGMOD	The LOGMOD pointing to a COS table entry, used to access an SNA partner. The default is taken from the DIRECTORY field.		
	LU2BYPAS	The standard key used for general TED 3270 incoming connections. If the calling partner can't be found in the directory the one associated with that key is used. The calling Partner is the requestor.	JOURNAL SMF	Z45PARTN Z07PARTN
SYSTCP	SYMB.PARTNER	The partner name defined in the directory that has an alternate TCP/IP Link defined by the parameters associated with the symbolic partner. When attempting to make an outgoing connection, the directory link is used. If any problem occurs, then the SYSTCP link will be used to attempt the connection. When an incoming connection is attempted, the caller's link is verified first from the directory, then using the SYSTCP links.	JOURNAL SMF	Z45PARTN Z07PARTN
	ADDRESS	One of the TCP/IP addresses that the partner can use for communications. When making an outgoing call, the list of TCP/IP addresses is scanned.	SYSLOG	
	PORT	The port number associated with the current TCP/IP address. Used only when making an outgoing call.		
	FTPBYPAS	The standard key used for general FTP incoming connections. If the calling partner is not found in the directory, the one associated with this key is used. The calling Partner is the requester.	JOURNAL SMF	Z45PARTN Z07PARTN
P1B2PRQ2 parmlist P1B2PRQ3 P1B2PREQ REQUEST	SPN=	The symbolic name of the remote partner you want to transfer with.	JOURNAL SMF	Z45PARTN Z07PARTN
P1B2PRQ2 parmlist REQUEST	API= A48= A34= SYSAPI82	Information that the application uses for identifying the transfer. Most protocols use a structure of ORIGIN+DESTINATION+NAME+DATE. Connect:Express moves this information from user to user and stores it in the journal.	JOURNAL SYSLOG	Z45TAPID API_CREATE
D1B2PCNX	CNXAPIXX=	The connection parmlist is sent to the connection user exit.		
D1B2PUEX	UEXAPIXX=	The transfer parmlist is sent to the transfer user exit.	JOURNAL	Z45TAPID

Local Connect:Express System Identification

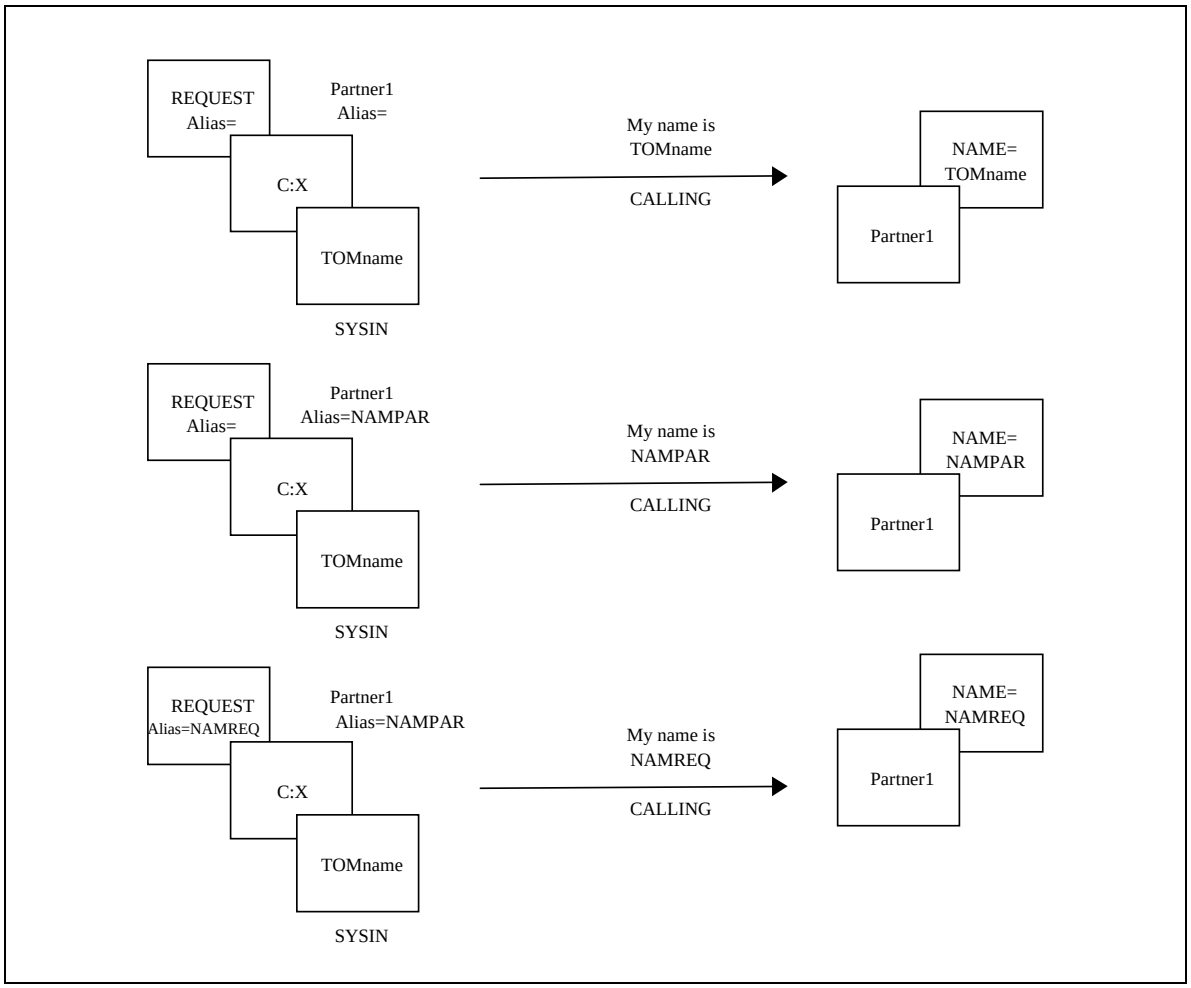
From	Field	Description	Report	Field
SYSIN	DPCSID	Symbolic name identifying the local Connect:Express system. This general identification is also associated with its DPCPSW value. Both the DPCSID and the DPCPSW values can be overridden by the alias for a connection. The DPCSID and the DPCPSW values, or their alias, must be negotiated with the partner before running the transfers. This name must match an entry in the remote Connect:Express directory.	none	
DIRECTORY	ALIAS	The ALIAS is a symbolic name identifying the local Connect:Express system for its associated partner. This name will override the DPCSID connection parameter. The ALIAS must be negotiated with the Partner before running the transfers. This name must match an entry in the remote Connect:Express directory.	none	
	ODETTE IDENTIF.	Application name for local ODETTE-FTP. This name is sent at connection time. It is associated with the local symbolic definition in the Connect:Express directory. If no API parameter is given at request time, the symbolic identification is used as the default in the origin and destination fields.	D1B2PCNX	
P1B2PRQ2 parmlist REQUEST	DPC=	An alias identifying the local Connect:Express system for the current transfer request. This name overrides the DPCSID parameter from SYSIN or the ALIAS field of the directory. It must be negotiated with the Partner before running this transfer. This name must match an entry in the remote Connect:Express directory.	none	
P1B2PRQ2 parmlist REQUEST	API= A48= A34= SYSAPI82	This information is used by the application for identifying the transfer. Most protocols use a structure of ORIGIN+DESTINATION+NAME+DATE. Connect:Express moves this information from user to user and stores it in the journal.	JOURNAL SYSLOG	Z45TAPID API_CREATE
D1B2PCNX	CNXAPIXX=	The connection parmlist is sent to the connection user exit.	none	
D1B2OUEX	UEXAPIXX=	The transfer parmlist is sent to the transfer user exit.	none	

Using ALIAS

An Alias enables you to transmit files using a name other than your Partner name. Sometimes you may need to do this when transferring files with a remote Partner. You can replace the DPCSID from SYSIN by the ALIAS from the directory or request extension for both outgoing and incoming connections. This section provides examples that show what is sent during an outgoing connection and the verification performed during an incoming connection.

Using ALIAS Output

In the figure below, there are three examples of outgoing connections. The first example does not have an alias, and “Partner1” has your DPCSID=TOMNAME field defined in his directory. The second example shows what happens when the partner alias field is defined. The alias NAMPAR must be defined in the partner's directory. The third example shows what happens when the transfer request alias field is defined. The alias NAMREQ must be defined in the Partner's directory.

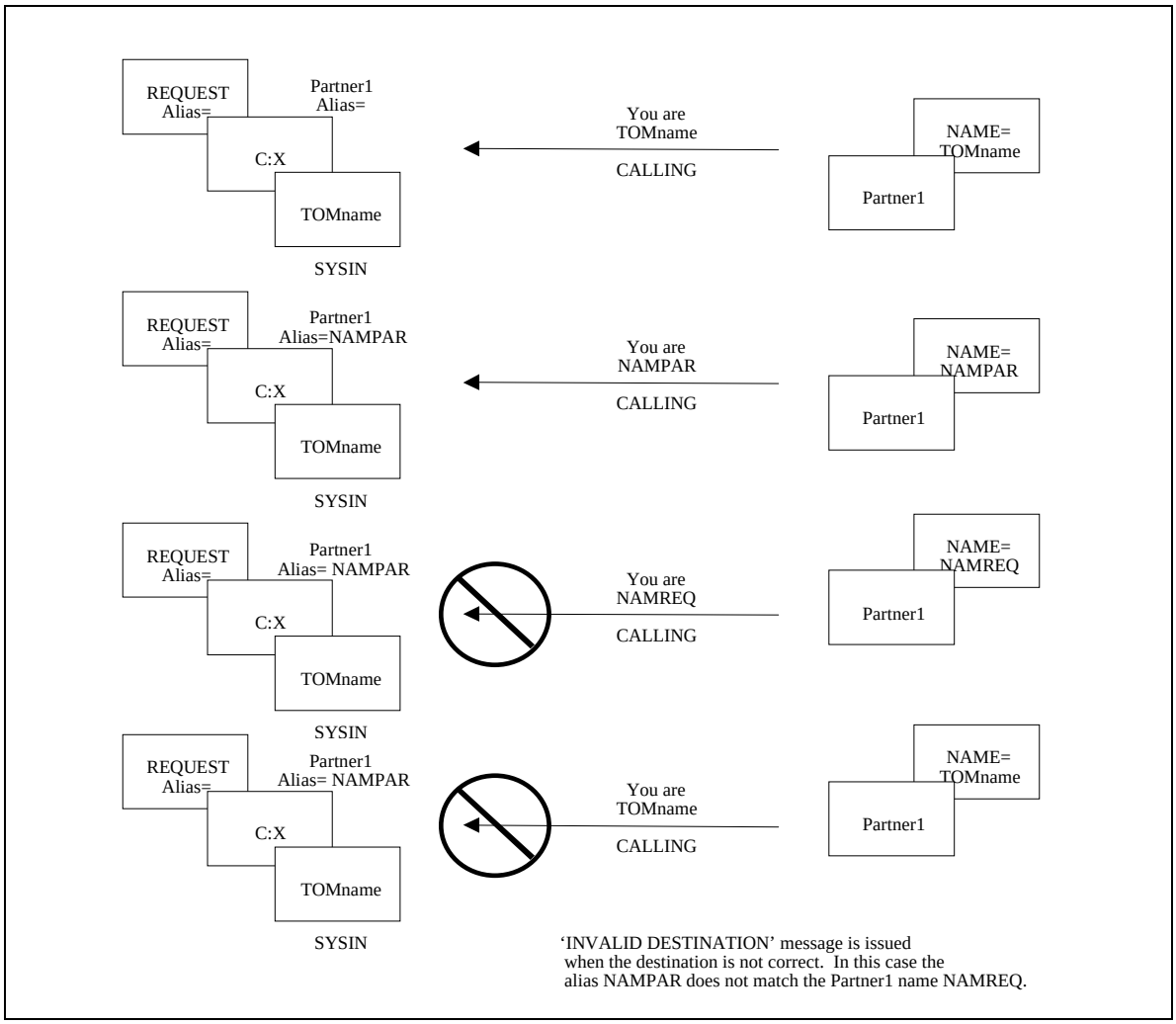


Using ALIAS Input

In the following diagram, there are four examples of incoming connections received from “Partner1.” Connect:Express identifies “Partner1” and checks the destination parameter sent by Partner1. The destination parameter is represented by “You are...” The first example shows that Partner1 is calling “TOMname” found in its SYSIN files as its own name. The second one shows what happens when the partner is calling “NAMPAR.” Connect:Express finds “NAMPAR” in the alias field of the partner1 definition and accepts the connection.

The third example shows that the destination NAMREQ is different from both the SYSIN file field and the alias field, and so Connect:Express rejects the connection. The fourth example shows that the destination parameter is equal to the SYSIN field, but an alias is defined in the partner1 definition, and so Connect:Express rejects the call.

Note: The fourth example is the same as the third one, but NAME is changed to TOMname, and "You are NAMEREQ" is changed to "You are TOMname."



Requestor Identification

The requestor is the user (TSO, BATCH, remote Partner, or origin) that originated the transfer request. For an incoming transfer request, the requestor name is generally the same as the origin. When a bypass is used from SYSX25, SYSSNA or SYSTCP files, the requestor is the name of the origin, and the Partner profile from the bypass is used as partner identification when connecting. The caller name is used as the transfer requestor identification. The requestor identification (the origin name) does not need to be defined in the directory.

From	Field	Description
Local OS/390 SYSTEM	TSO USER	The TSO user symbolic name which issued a request from Connect:Express TSO/ISPF panels (4.1, 4.2, 4.3, 4.4).
	JOBNAME	The name of the JOB that executed as a batch utility: P1B2PREQ P1B2PRQ2 P1B2PRQ3 or called the API subprogram: L1B2P20 L0B2Z20
External user	Origin	The name of the remote system that requested an incoming transfer.

Overview of Security

Transfer operations should be executed in a secured environment. Connect:Express has provisions for the following five levels of security.

1. Symbolic security for flow control.
2. System security for data sets and OS/390 commands access.
3. Network security for incoming call verification.
4. Encryption for integrity, confidentiality, authentication and non-repudiation.
5. User Exit for user specific security.

Symbolic Security

In a Connect:Express environment, file transfers are organized by symbolic names and passwords. Symbolic security is based on the following parameters:

- The Partner symbolic name and Partner type.
- The File symbolic name which identifies the transfer direction for the file, the Partner or list of partners for receiving the file, and the Partner or list of Partners for Transmitting the file.

Connect:Express verifies that the calling Partner exists in the Partners directory and that the password of the calling Partner is correct. The called Partner (or destination) is also verified using the local Connect:Express symbolic name DPCSID or alias.

Connect:Express also verifies that the file exists in the files directory and that the direction of the transfer request is correct. The receiving partner and transmitting partner are verified by checking the following values in the Partner fields:

Value	Description
$$ALL$$ or *	Any partner defined in the directory is allowed.
Partner	Only this partner is allowed.

Value	Description
List	Any partner defined in the list is allowed.
$$API$$	Used for incoming transfers only. No control is done. It is the responsibility of the application (user exit).
££API††	Used for incoming transfers only. Any user defined in the list associated with the file is allowed. Note: A user is not defined in the directory. This is the transfer requestor and it can be different from the Partner.

The available symbolic security facilities depend on the partner type, as shown in the following table.

Partner Type	Symbolic Security Available
T = Connect:Express (TOM) partner	All facilities available: symbolic name, password, flow control.
O = Other	All facilities available: symbolic name, password, flow control.
A = Application	Limited controls: Identification is considered the responsibility of the application. This profile suppose that user exits are implemented.
E = ETEBAC 1/2	All facilities available: symbolic name, password, flow control. The application has limited controls. Refer to Connect:Express /F-ETB410 documentation for more information.
3 = ETEBAC 3	All facilities available: symbolic name, password, flow control. The application has limited controls. Refer to Connect:Express /F-ETB410 documentation for more information.

Security Parameters

The following table lists all parameters in Connect:Express files that are involved in the security processes.

From	Field	Description	Report	Field
SYSIN	DPCSID	The local name of the Connect:Express system with which you are transferring files. It is verified as the destination for an incoming call and sent as the origin of an outgoing call. This value is the default, but can be replaced by an ALIAS for one partner or for one transfer.	Journal user exit TSO/ISPF 2.3	Z45TAPID U03ORIGN U03DESTN U05ORIGN U05DESTN API
	DPCPSW	The password of the Connect:Express system with which you are transferring files. It is sent to a partner for establishing connection. This value is the default. It can be replaced by an ALIAS for one partner or for one transfer.		
	RACFCN	RACF control can be disabled (RACFCN=N)		
	ADHOCN	This parameter enables or disables all or part of the AD HOC feature. You can disable the entire AD HOC facility or leave it with no security.	TSO/ISPF 4.5	ADHOCN=
	RACFUD	Default RACF USER, as opposed to the partner RACF USER field.		

From	Field	Description	Report	Field
	UPRFCT	UPRFCT=N disables the start command activation provided in the file directory definition. If 'N,' a warning message is issued when a beginning or end of transfer command is found during the transfer.	TSO/ISPF 1.2 screen 4/5	UPRFCT=
	LRECLT	LRECL verification: 'Y' enables the verification. The sender's LRECL must match the symbolic file definition or the existing data set format. 'N' disables the verification. The sender's information must be coherent with the symbolic file definition or the existing data set format, but the local and remote definitions are made independent. Combined with the next parameter, RECFMT=N, you can receive a remote fixed file in a local variable file.		
	RECFMT	RECFM verification: 'Y' enables the verification. The sender's RECFM must match the file definition or the existing data set format. 'N' disables the verification. The sender's LRECL must be coherent with the definition, but the local and remote definitions are made independent. Combined with the previous parameter, LRECLT=N, you can receive a remote fixed file in a local variable file.		
PARTNERS DIRECTORY	SYMB. NAME	Remote partner identification. It is verified as the origin for an incoming call, and sent as the destination of an outgoing call.	Journal user exit SYSLOG TSO/ISPF 2.3	Z45TAPID UEXPARTNU 03DESTIN U03ORIGN U05DESTIN U05ORIGN API
	PASSWORD	Remote partner password verified. It is verified for an incoming call.		
	TYPE	Enables/disables symbolic security and some specific Connect:Express facilities like transferring PDS or VSAM files with no UNLOAD -RELOAD processing or AD HOC request.		
	RACF USER NAME	Used for RACF verification on incoming connections when RACFCN=Y. If omitted, RACFUD from the SYSIN is the default.		
	RACF USER GROUP	Used for RACF verification on incoming connections when RACFCN=Y.		
	ALIAS DPCSID	The local name of the Connect:Express system with whom you are transferring files. It is verified as the destination for an incoming call and sent as the origin of an outgoing call. This value replaces the default DPCSID	Journal user exit TSO/ISPF 2.3	Z45TAPID UEXPARTNU 03DESTIN U03ORIGN U05DESTIN U05ORIGN API
	ALIAS DPCPSW	The password of the Connect:Express system with whom you are transferring files. It is sent to this partner for establishing connections. This value replaces the default DPCPSW		

From	Field	Description	Report	Field
	SESSION PROT.	Number of the session table: value of xx in T1B2PSxx. This table provide the CRC option.		
	RSA DES SECURITY	Number of the security table: value of xx in T1SECPxx.		
T1B2P52x T1B2PS3x T1B2PS5x	USE CRC	Yes or No. Available for protocols 2 (OFTP), 3 and 5 (PeSIT).		
T1SECPxx	GENEX	Encryption User exit.		
	MODN	Encryption User exit.		
FILES	SYMB. NAME	File identification.	Journal user exit SYSLOG TSO/ISPF 2.3	Z45TAPID UEXPARTNU 03DESTIN U03ORIGN U05DESTIN U05ORIGN API
DIRECTORY	DIRECTION	Transmission, Reception or Both.		
	SENDING PARTNER	The partner or list of partners allowed for sending files. You can call them to receive a file or they can call you to send a file.	Journal user exit SYSLOG TSO/ISPF 2.3	Z45TAPID UEXPARTNU 03DESTIN U03ORIGN U05DESTIN U05ORIGN API
	RECEIVING PARTNER	The partner or list of partners authorized to receive files that you send. You can call them to send a file or they can call you to receive a file.	Journal user exit SYSLOG TSO/ISPF 2.3	Z45TAPID UEXPARTNU 03DESTIN U03ORIGN U05DESTIN U05ORIGN API
	RSA DES SECURITY	Number of the security table: value of xx in T1SECPxx.		
REQUEST EXTENSION	RSA DES SECURITY	Number of the security table: value of xx in T1SECPxx.		
	RACF USER GROUP	RACF group used for local RACF verification. When you make a transfer request you can change your group.		
	ALIAS DPCSID	The local name of the Connect:Express system with whom you are transferring files. It is sent as the origin of an outgoing call. This value replaces the default DPCSID and the ALIAS DPCSID of the directory.	Journal user exit TSO/ISPF 2.3	Z45TAPID UEXPARTNU 03DESTIN U03ORIGN U05DESTIN U05ORIGN API
	ALIAS DPCPSW	The password of the Connect:Express system with whom you are transferring files. It is sent to this partner for establishing connections. This value replaces the default DPCPSW and the ALIAS DPCPSW of the directory.		

From	Field	Description	Report	Field
AD HOC REQUEST	RACF USERID	Used for RACF verification on the remote site.		
	RACF USER GROUP	Used for RACF verification on the remote site.		
	RACF PASSWORD	Old and new passwords are provided for verification on the remote site.		
P1B2PRQ2	SID=	The local name of the Connect:Express system with whom you are transferring files. It is sent as the origin of an outgoing call This value replaces the default DPCSID and the ALIAS DPCSID of the directory.	Journal user exit TSO/ISPF 2.3	Z45TAPID UEXPARTNU 03DESTIN U03ORIGN U05DESTIN U05ORIGN API
	PSW=	The password of the Connect:Express system with whom you are transferring files. It is sent to this partner for establishing connections. This value replaces the default DPCPSW and the ALIAS DPCPSW of the directory.		
	SEC=	Number of the security table: value of xx in T1SECPxx.		
	RGR=	RACF group used for local RACF verification. When you make a transfer request you can change your group.		

System Security

Connect:Express interfaces with system security facilities like RACF – ACF2 and TOP SECRET through the SAF interface. System security is a combination of RACF type security and options that control file allocation and the issuing of OS/390 commands. System security facilities are invoked to check access to the data sets transferred. You can disable some verifications. The following conditions apply:

- Connect:Express must be allowed to read and create-write all files to be transferred and temporary workfiles used for unloading/reloading. If not, the typical return code is TRC=2098.
- The local requestor or the application that makes the transfer request must be allowed access to the data set. The local requestor can provide an RACF group at request time. The default is its own group.
- The remote requestor or the remote partner who connects to transfer a file must be allowed to access the data set. Optional RACF USER and RACF GROUP fields are provided in the partner definition for incoming request verification.
- The AD HOC requestor or the remote user who connects to transfer a file with the AD HOC facility must be allowed to access the data set on the remote site. With this feature, the user provides a user logon and password, and the symbolic partner definition used for connection (RACF USER and RACF GROUP) is not involved in the security checking. The AD HOC facility can be completely disabled, or the security option of the AD HOC feature can be disabled.
- Connect:Express is enabled to start any OS/390 command, but you may not want users to have access to this feature. You can set a general option in the SYSIN file to disable this feature (UPRFCT).
- You can ensure that the allocation parameters sent are consistent with what you defined in the file directory.
- HFS files are under the control of the BPX server.

Network Security

Network address control is done through the partner definition. Each link type (X.25, SNA, TCP/IP) is given one network address or a list of network addresses. This control can be disabled if one of the addresses is set to '*' in the directory or in the alternate address files.

Encryption

CRC (Cyclic Redundancy Check) is activated by an option provided in the session tables T1B2PS2x, T1B2PS3x and T1B2PS5x. This algorithm follows the ISO DIS 8073 transport layer standard. Refer to the *Connect:Express OS/390 SSL Guide* for securing file transfers.

User Exits

You can manage security through the user exit security interface. Connection, selection, and transfer exits enable you to control access at different levels. Connect:Express OS/390 provides an RACF connection server exit that you can implement called M1USRCNA. The macro M1USRCNA is provided in the *MACLIB* for customization, and an example is provided in the *SAMPLIB*. See Chapter 2, *User Exits* for more information about user access control.

Protocol Management

When you transfer a file, you exchange information with your Partner using a file transfer protocol that defines the rules that Partners use to negotiate with each other and transfer files. File transfer protocols are implemented over network protocols so the administrator must be aware of network protocol parameters used to communicate over physical links.

Transfer Protocols

A transfer protocol consists of commands for negotiating and terminating a connection or transfer, or for sending a file. Protocol return codes (PRC) are a means of controlling operations in association with Connect:Express return codes (TRC).

A transfer is comprised of a series of events that precede or result from protocol commands. Events and results are reported in the Connect:Express SYSLOG file, Request Control Table (RCT) and Journal. Reports can be stored in an SMF file.

The protocol used for transferring a file is associated with the Partner using the PROT.NUM field of the Partner definition in the directory. Each protocol is assigned 15 session tables called T1B2PSxy, where x is the protocol number and y the session table number (y= 1 – 9, A – F). Fields in protocol commands contain information about the connection and the data to be transferred. This includes the identification parameters set in Connect:Express directories or other files, the session and transfer parameters set in tables, and application parameters.

The following table describes the transfer protocols supported by Connect:Express.

Active Protocol	Description
PeSIT	Protocol specifications from the French Banking Organization. PeSIT is an FTAM like protocol enabling all Connect:Express facilities (AD HOC, PDS, VSAM files). The latest version, PeSIT-E, is used for RSA/DES security.

Active Protocol	Description
ODETTE-FTP	Protocol specifications from the Organization for Data Exchange by TeleTransmission in Europe for the European Car Constructors organization.
ETEBAC3	Protocol specifications from the French Banking Organization. Described in the French documentation Connect:Express /F-ETB410.
ETEBAC1/2	Protocol specifications from the French Banking Organization. Described in the French documentation Connect:Express /F-ETB410.
FTP	Protocol Specification from RFC959.

Note: PeSIT and ODETTE-FTP protocols are described in Chapter 5, the FTP protocol is described in the FTP Guide, and the Etebac3 protocol is described in the Etebac3 User Guide.

Network Protocols

You can connect with a Connect:Express Partner using one or more link types depending on its definition in the Directory fields and in the SYSSNA, SYSTCP, or SYSX25 files. Connect:Express and the Transfer Protocol Manager (APM) work with the Network Manager (ANM) to communicate with the Partner. The link drivers of the ANM handle different types of connections and interface with the Network Protocols executed by VTAM, X.25-NPSI, or TCP/IP. FTP partners are accessed over TCP/IP links, and FTP transfers are handled by the AFM and EAS address spaces.

The table below identifies the transfer and network protocols that Connect:Express supports. The codes are used in the table on page 3-37.

Code	Protocol
D	PeSIT D
E	PeSIT E
O	ODETTE-FTP
3	ETEBAC 3
1	ETEBAC 1,2
F	FTP

The following table lists the available environments and is subject to change.

Connect:Express Software and Protocols

Computer Company	System	X25				SNA			LU62		pAD				BSC	TCP			
		D	E	O	3	D	E	3	D	E	D	E	O	3	1	D	E	F	O
IBM	OS/390	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
	AIX	X	X		X						X			X		X	X		
COMPAQ																			
OSF1	UNIX	X	X								X	X		X		X	X	X	
TANDEM	GUAR.D30	X	X						X		X	X				X	X		
HP	HP-UX	X	X		X						X	X		X		X	X	X	
NCR	UNIX	X	X		X						X	X		X		X	X	X	
SUN																			
SOLARIS	UNIX	X	X		X						X	X		X		X	X	X	
VARIOUS	Windows32	X	X		X				X	X	X	X		X					

Connect:Express Protocol Facilities

Transfer protocols are limited to transferring sequential files and identifying virtual files and Partners, however some file transfer protocols provide user data fields, but some restrictions can apply. Some fields are available only if both partners use the PeSIT protocol, others are available only if both partners use Connect:Express, and others require that both platforms are OS/390. Open protocol facilities are those features that are available with any partner on any platform. General protocol facilities are available when both partners are Connect:Express and use any operating platform. Those features that are only available in the OS/390 environment are called OS/390 protocol facilities.

Connect:Express Open Protocol Facilities

Open protocol facilities are available for any partner and depend on the file transfer protocol used. All of these facilities are keywords that you can use to compute a file data set name or a parameter passed to a user command. The table below shows the relationship between the Connect:Express field sent, the protocol field used, and the facility that receives the value. Some parameters are only available with Partners of type Other.

Connect:Express Parameter	Protocol	Field	Partner type	Facility to receive the value
REMOTE DSNAME	PeSIT	Pi99 (44 characters)	Other	&Pi99
API=P:...	PeSIT	Pi99 (80 characters)	Other	&Pi99
API=U:....	PeSIT	Pi80		&EXTLAB
FILE IDENTIFICATION	ODETTE	File Identification		&EXTDSN, &USRVAR
REMOTE DSNAME	ODETTE	File Identification		&EXTDSN, &USRVAR

Note: If a Connect:Express Partner has a type of Other, only open protocol facilities are available with this Partner, even though it uses Connect:Express software.

Connect:Express General Protocol Facilities

General protocol facilities are available with any Connect:Express software running in any environment like UNIX, NT, VMS, or OS/400 that uses the PeSIT protocol. These features are activated if the Partner Type field in the Partner definition is set to T for TOM. This indicates that the Partner is a Connect:Express monitor.

The service transfers the data set name from the sender to the receiver. This enables the receiver to build the data set name of the file using symbolic keywords that apply to the sender data set name such as &EXTDSN, &USRVAR1, and &USRVAR2. The following screen shows an example.

```
Receiving index1.index2.index3 into INDEX1.&EXTDSN results in
INDEX1.index1.index2.index3.
Receiving index1.index2.index3 into INDEX1.&USRVAR1 results in INDEX1.index3.
Receiving index1.index2.index3 into INDEX1.&USRVAR1.&USRVAR2 results in
INDEX1.index3.index2
```

The ADHOC facility is available with Connect:Express for NT. See *Identifying the User* on page 3-41 for more information.

Note: When using PESIT protocol, &PI99, &EXTDSN, and &USRVAR keywords are exclusive. To use &PI99, the Partner type must be other. To use &EXTDSN and &USRVAR, the Partner type must be TOM.

Connect:Express OS/390 Protocol Facilities

OS/390 protocol facilities are available between Connect:Express Partners running in an OS/390 environment. You can only use the PeSIT protocol with these features. They are activated if the Partner Type= T for TOM, and there are two levels of service offered. Level 1 transfers the data set name, and Level 2 identifies the user. This is controlled by the ADHOCN field of the SYSIN file. The following tables show the services that are available with each protocol level, session table, and Partner type.

Transferring Physical Data Set Name

Protocol Level	Session Table	Service Level	Services	Partner Type
PeSIT-D	T1B2PS31	1	&EXTLAB	other
PeSIT-D	T1B2PS30	1	&USRVAR, &EXTLAB	Connect:Express
PeSIT-D	T1B2PS31	1	&USRVAR, &EXTLAB	Connect:Express
PeSIT-D'	T1B2PS32	2	&EXTDSN,&USRVAR &MBR,&EXTLAB, &EXTDAT,&EXTNUMB	Connect:Express
PeSIT-E	T1B2PS5x	2	&EXTLAB	other
PeSIT-E	T1B2PS5x	2	&EXTDSN,&USRVAR &MBR,&EXTLAB, &EXTDAT,&EXTNUMB, &MBR AD HOC: local and remote DSN exchanged	Connect:Express
PeSIT-E	T1B2PS3x	1	&EXTLAB, &P99	other
PeSIT-E	T1B2PS5x	1	&EXTLAB, &P99	other

† or later

Exchanging File Attributes

Protocol Level	Connect:Express Version	Session Table	Service Level	Services	Partner Type
PeSIT-D	TOM222†	T1B2PS31	1	none	any
PeSIT-D	TOM222†	T1B2PS30	1	none	Connect:Express
PeSIT-D'	TOM222†	T1B2PS32	2	Attributes &X (Lrecl, Blksize, Recfm)	Connect:Express
PeSIT-E	TOM222†	T1B2PS5x	2	none	other
PeSIT-E	TOM222†	T1B2PS5x	2	Attributes &X (Lrecl, Blksize, Recfm) AD HOC: local and remote attributes exchanged	Connect:Express

† or later

Transferring PDS, VSAM, or TAPE Files

Protocol Level	Connect:Express Version	Session Table	Service Level	Services	Partner Type
PeSIT-D	TOM222†	T1B2PS31	1	none	any
PeSIT-D	TOM222†	T1B2PS30	1	none	Connect:Express
PeSIT-D'	TOM222†	T1B2PS32	2	PDS send/receive, VSAM send TAPE send/receive	Connect:Express
PeSIT-E	TOM222†	T1B2PS5x	2	none	other

† or later

Protocol Level	Connect:Express Version	Session Table	Service Level	Services	Partner Type
PeSIT-E	TOM222†	T1B2PS5x	2	PDS send/receive, VSAM send TAPE send/receive	Connect:Express

† or later

Identifying the User

When the RACF function is enabled, Connect:Express processes an input transfer request from a remote Partner with RACF control using the RACF user field of the remote partner. Any user of the calling Connect:Express can access all files authorized for this monitor. With the AD HOC facility, the called Connect:Express can identify you so that you can access your own files on the remote system. A user can only access the files for which they are authorized. The files that you can access may be different from those that the monitor has authorized the user of the remote system to access.

Reporting

You can access reports that provide information about the status of the Connect:Express monitor and operations in general. You can view the status online as it is happening, or receive notifications through a user exit, or offline in a journal or log file. You can then use Help to interpret the return codes.

You can use reports to start other processes or take corrective action when required. The Reporting table shows the report options that are available for the Monitor Status and Operations Status.

The Reporting Table

Operations and Resources	SYSLOG	Journal	User Exits	User Commands	API	TSO/ISPF Interface
Monitor Status						
Initialization	X	X	X	X		
Resources	X				X	X
Termination	X	X	X	X		
Operations Status						
Transfer Requests	X	X	X		X	X
Connections	X		X			
Transfer Execution			X	X		

Monitor Status

When Connect:Express is initialized or terminated, you can start a user command which reports the status of Connect:Express. This information comes from the following parameters in the SYSIN file: UPRBEG=, UPREND=, and UPRABE=. These fields tell you if the Monitor is up or down or if any abends have occurred. Some user exits are also invoked during initialization and termination of Connect:Express. These include connection exits, selection exits, and journal exits.

You can check the status and make modifications using the API or TSO/ISPF interface. These can help you enable or disable resources and manage abnormalities by setting alert thresholds and taking corrective actions.

Operations Status

You can display the status of a transfer request, track connection errors, or introduce your own procedures before, during, and after the execution of a transfer.

At any time, you can display the status and results of transfer requests by their local request number using API or TSO/ISPF reports. You can also display active transfer requests for one partner or file, or all transfer requests using the API or TSO/ISPF. Then you can activate, interrupt, or restart these requests. The log file lets you look at the entire history of the transfer request while the journal file gives you the transfer results and statistics.

You can control the local user requests with a transfer request user exit. For example, you can set the priority of a transfer or select the class of the request. In addition, you can track connection errors online with a user exit or offline using the log file messages.

Additional Reporting

Transfer operations events are reported in the Connect:Express log file (SYSLOG) with return codes. There are four types of Return codes:

1. Connect:Express (TOM) Return Codes – TRC
2. Transfer Protocol Return Codes – PRC
3. Network Return Codes – NRC
4. System Return Codes – SRC

Note: See Appendix B, *Error Codes and Messages* for a listing of TRCs, PRCs and NRCs. SRCs can be found in the IBM OS/390 documentation.

APM Messages and Logging

Any events pertaining to the APM are reported in the APM message file (SYSMSG). Transfer operations incidents such as allocation, exit, and network errors are reported in the APM log file (SYSLOG) with return codes. The APM log file supplements the information in the Connect:Express log file. APM messages are listed in Appendix B, *Error Codes and Messages*.

Connect:Express Messages

Any events pertaining to Connect:Express are reported in the Connect:Express message file (SYSMSG). The message structure is shown below.

```
10/04/00  14:04:32  TOM8046I SUBSYSTEM REQUEST CLEAN UP COMPLETED
10/04/00  14:04:32  TOM8047I SUBSYSTEM INTERFACE INHIBITED
10/04/00  14:04:32  TOM8050I SHORT  ABNORMAL TERMINATION END
```

The SYSMSG file shows the time stamp, the subsystem name (TOM8), the message number (046), message type (I), and text. The following table lists the message types:

Code	Description
W	Warning
E	Error
I	Information

Connect:Express Return Codes Structure Descriptions

Connect:Express return codes are structured to help you identify the processes causing errors. Online help lists all the codes.

Return Code	Description
1	Protocol error
2	All errors detected by the Connect:Express Monitor. For example, scheduling, flow control allocation, security.
3	APM file access error
4	User exit error
5	Transfer negotiation error
6	Connect:Express system error (for example, subsystem interface, getmain error)
7	APM checkpoint error
8	APM subtask ABEND
9	Transmission error detected by APM
A	ANM handler error
B	BSC communication error

Sending User Messages to the SYSLOG Files with L1B2LOG

You can send user messages to the SYSLOG files of Connect:Express from a user exit, or from the generalized user exit L1GFIUE1. You can use the connection exit and Journal exit to send messages to the SYSLOG file of the TOM address space. Selection, beginning of transfer, and end of transfer user exits, such as the generalized user exit, can be used to send messages in the SYSLOG file of the APM address space. A program example called L1USRLOG is provided in the *SAMPLIB*.

The program that calls L1B2LOG interface must be re-entrant. The following screen shows how to call L1B2LOG:

```
*--------------------------------------------------------------------*
* PROGRAM INITIALIZATION
*--------------------------------------------------------------------*
*--------------------------------------------------------------------*
* PROGRAM PROCESS
*--------------------------------------------------------------------*
  MVC  USRLOG,MSG1
       LA      R1,USRLOG              POINT TO MESSAGE
       ST      R1,PARMLOG             STORE PARM ADDRESS
       LA      R1,PARMLOG             POINT TO PARM LIST
       CALL    L1B2LOG                PERFORM
       LTR     R15,R15                OK ?
       BNZ     LERRMSG                NO
       B       LEND                   END

*--------------------------------------------------------------------*
* PROGRAM TERMINATION
*--------------------------------------------------------------------*


*--------------------------------------------------------------------*
* PROGRAM CONSTANTS
*--------------------------------------------------------------------*

*       FIRST 19 CHARACTERS ARE RESERVED
MSG1   DC      CL133' AA/MM/JJ HH:MM:SS HELLO, WORLD'


*--------------------------------------------------------------------*
* PROGRAM DYNAMIC WORK AREA
*--------------------------------------------------------------------*

PARMLOG DS      F
USRLOG  DS      CL133
```

Refer to the *Connect:Express OS/390 Utilities Guide* for information about sending messages with L1GFIUE1.

Reliability

Connect:Express is designed to be in service 24 hours-a-day, 7 days-a-week, and reliability is of primary importance. To accomplish this, Connect:Express provides checkpoint and recovery files to ensure data integrity in the event of a system failure or other event that requires you to restart Connect:Express.

If a system failure occurs, you can restart where you left off with a hot-start. In some cases, the monitor must be stopped manually like when you update the software or change the Connect:Express configuration. When the monitor is not up, there is a recovery file that receives and records transfer requests. When the monitor returns to service, it will restart where it left off, then read the recovery file and perform the transfer requests. The hot-start uses a checkpoint file that contains the information and current status needed to restart where the monitor left off.

A cold-start is a fresh start and does not use checkpoint information. In cases when a cold-start is planned, the checkpoint and recovery files should be saved. Then after the cold-start, the checkpoint and recovery files can be reloaded.

Note: The Extended Recovery capability in Connect:Express enables automatic recovery within an OS/390 sysplex environment, even when a complete failure of OS/390 occurs. See Chapter 1 *Sysplex Environment* for more information about Extended Recovery.

To further ensure reliability, parameters are always verified at initialization and whenever they are added, changed, or deleted. The next sections identify when the monitor needs to be stopped and what type of start needs to be performed.

Conditions Requiring an APM Start

Some updates to Connect:Express require you to stop the APM, but not the monitor. The following table identifies the components that need to be reloaded when the APM is stopped.

Component	Description
Tables	Most Connect:Express tables are automatically reloaded when updated using the TSO/ISPF interface. The tables used by the APM, however, are not automatically reloaded. The transfer server exits table, T1APMSRT, and the presentation protocol parameters table, T1B2PP.., must be manually reloaded by issuing stop APM/start APM commands.
User Exits	Some user exits are loaded by the APM during its initialization. Any change to the exits must be followed by a stop APM/start APM command. This is needed in the following cases: • A transfer server exit is declared in the T1APMSRT table and can be inactivated after an ABEND, or if not found during APM initialization. • Middle of transfer user exits are declared in T1B2PP.. tables and loaded during APM initialization.

Conditions Requiring a Hot-start (Run=H)

You can hot-start Connect:Express when changing some parameters. The following table lists the parameters that use a hot-start to make changes effective.

Component	Description
SYSIN	When changing most of the SYSIN parameters, you only need a hot-start to make the changes take effect. One exception is when you update the APLNUM field of the SYSIN file. This must be followed by the '$LOAD$' procedure and then a hot-start.
CXPLEX	When changing the CXPLEX parameter file, you only need to hot-start of Connect:Express.
User Exits	Some user exits are loaded by Connect:Express during initialization. Any change to the exits must be followed by a hot-start of Connect:Express. These include: • A connection user exit declared in the T1B2PCNT table that can be deactivated after an ABEND or if it is not found during initialization. • The request control user exit. • A Journal user exit

Component	Description
Tables	Some tables used by Connect:Express are not automatically refreshed in Connect:Express when updated and require a hot-start. These include: T1B2PCNX–connection user exits (option 3.3.CNT) T1B2PSLD–session links definitions (option 3.3.SLD) T1B2PS..–session protocol parameters (option 3.3.SXX) T1B2PVOL–allocation volumes (option 3.3.VOL) T1B2PACS–access control (option 3.3.ACS) T1B2PCOD–protocol return codes and automatic restart (option 3.3.COD)

Conditions Requiring a Cold-start (RUN=C)

Updating Connect:Express software or changing some configuration parameters in the SYSIN file will require a cold-start. Use the OFFLOAD, RELOAD of CHECKPOINT/RECOVERY procedures before making a cold-start. The following table identifies the components that require a cold-start.

Component	Description
Version change or update	A new version of Connect:Express or a PTF (an update to a version) requires a cold-start.
SYSIN	If any of the following SYSIN parameters are changed, then a cold-start is required. • DPCSID • DPCPSW • MAJIND • CMDPRE • RQEMAX • DAPMxx cards added or deleted

When a cold-start is required, you should plan for it so that you can maintain the current operational status of Connect:Express. To do this, you need to save the checkpoint and recovery files using an unload procedure. Then, perform the cold-start, followed by the reload procedure which restores the previous operational status. Offload and reload procedure examples are provided in the *SAMPLIB* directory in the $CKOFLOD and $RCOFLOD modules.

Note: The offloading of *SYSCHK* and *SYSRCY* can be included as preliminary steps of the standard Connect:Express job.

Timers

To avoid locked situations, Connect:Express address spaces activate timers. The table below describes each timer.

Timer	Description
ANM Timer	During initialization, the ANM activates the network drivers and the TOM waits for the end of ANM initialization. If one driver initialization doesn't succeed and gives no response, the ANM will continue and return to the TOM that continues to initialize the APM and other resources. This timer is fixed and set to 4 minutes 30 seconds. Connection and deconnection timers are activated by the network layers. The ANM provides fixed connection and deconnection timers with values greater than the current network specifications. The Data flow timer during the opened session is controlled by the ANM and is set to a maximum of 7 minutes before aborting a session that is blocked.
TOM Timer	In the TOM address spaces, the timer is set against user exits. The connection user exit is run in the TOM address space and is protected by a 5 minute timer.
APM Timer	In the APM address space, the timer is set against remote network problems and local user exits. The selection, beginning, middle, and end of transfer user exits are run in the APM address space and are protected by two options. One of these options defines how you call the exit and the time it is allowed to be active before returning back. When this timer expires, the user exit is detached if it was previously attached, or the effector is detached.
User Exit Timer	During Connect:Express termination, the APM is notified and waits for effector termination. If the effector doesn't stop, it is detached after 2 minutes.
AFM/EAS Timers	In the FTP address spaces, the timers are set against remote network problems and remote user inactivity. One timer is dedicated to user inactivity during the connection phase, another one is dedicated to user inactivity while connected, and another one controls the data transfer duration.

The following screen shows an example of a user exit timer.

```
// TOM410 EXEC PGM=P1APM001,REGION=4M,TIME=1440,
// PARM=('APM=&APM','SSN=&SSN','MSG=&MSG','EFF=&EFF','ETC=&ETC',
//        USD='uuuuuuuu','SRV=L1APMSRV','TMX=60','BRX=1')
//*--------------------------------------------------------------------
//* APM (AUXILIARY PROTOCOL MANAGER)
//*
//* User exits are called (BRX=1) at BEGINING and END of transfer
//* with a 60 second elapse allowed (TMX=60)
//*
//*--------------------------------------------------------------------Transfer
operations need to be
```

Chapter 4

Incoming and Outgoing Transfers

This chapter describes the file transfer operations of Connect:Express. It covers PeSIT, Odette, and ETEBAC transfers that are processed in the TOM, ANM, and APM address spaces. This chapter outlines the steps of an incoming and an outgoing file transfer and identifies the parameters, the transfer process, and reporting capabilities.

Overview

Each step of transfer operations involves a number of parameters and takes place in one of the address spaces that are running when Connect:Express is started. Each process is reported in a different way. This chapter has a section for outgoing transfers and a section for incoming transfers. The following topics are discussed in each section:

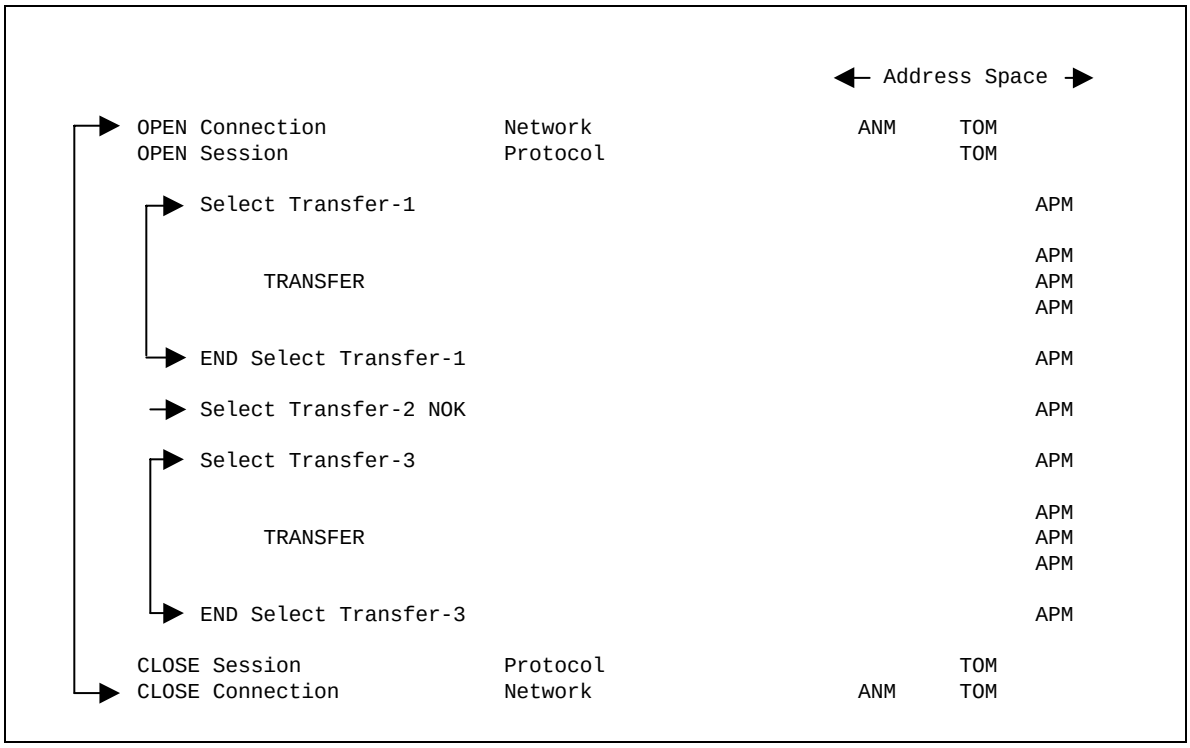
- Connection – establishing the Network link (X25, TCP/IP, SNA..)
- Session – establishing the Protocol link (PeSIT, ODETTE, ETEBAC3…)
- Selection – negotiating the file transfer with the protocol conventions.
- Transmission – sending the file to a remote partner.
- Reception – receiving the file from the remote partner.
- Transfer – exchanging the data.

Connection is executed in the ANM address space. Connection and Session are managed in the TOM address space, and selection and transfers are managed in the APM address space.

Note: The ANM is a network services provider used by both the TOM and APM address spaces. The ANM executes network connections and de-connections, and sends and receives data.

One session is executed during a connection, and more than one transfer selection can be executed during a session. After the transfer negotiation is complete, the file transfer is executed. If the transfer negotiation fails,

the next selection is executed, if any. Every step that was successfully completed is closed or ended, as shown in the following diagram.



The following table lists the situations that are discussed in this chapter and identifies any special considerations. Outgoing transfers are discussed first. These are the transfers that you request locally. Incoming transfers correspond to external requests you receive from remote partners. Data exchange is the same for both outgoing and incoming transfers.

Outgoing transfers		**Incoming transfers**
Outgoing Connection *Outgoing call resources configuration*		Incoming Connection *Input control bypass*
Outgoing Session *Partner identification*		Incoming Session *Partner identification*
Outgoing Selection *File identification*		Incoming Selection *File identification*
Outgoing Transmission *User process, File attributes, Negotiations, Checkpoint*		Incoming Transmission *User process, File attributes, Negotiations, Checkpoint*
Outgoing Reception *User process, File attributes, Negotiations, Checkpoint*		Incoming Reception *User process, File attributes, Negotiations, Checkpoint*
Data exchange		
Transmission	Data Translation	Reception

Outgoing Transfers

A local TSO or batch user sends a transfer request to Connect:Express. The four items listed below are mandatory.

1. Symbolic File name
2. Symbolic Partner name
3. Transfer direction
4. Physical data set name

They must either be provided with the transfer request or stored in the symbolic File name entry in the Files directory. You can store all necessary parameters so that you only have to enter the symbolic File name to execute a transfer.

If Connect:Express is started, the request is identified by a request number, queued into the Request Control Table, and set to enabled or saved in the Connect:Express recovery *SYSRCY* file. If Connect:Express resources are all available and the number of active sessions for the Partner has not reached the maximum allowed, the request is scheduled. If the transfer protocol provides a restart facility, the request is allocated a checkpoint record. The request and checkpoint record will be purged at the successful end of transfer or if you make a purge request command.

Note: If the file space is known and it is less than or equal to one kilobyte, no checkpoint is allocated.

Parameters are stored in the following locations:

```
SYSIN file
PARTNERS   directory + SYS-SNA/TCP/X25 files
FILES      directory
REQUEST    fields
TABLES     session, data presentation
USER EXITS
```

Information is reported in the following files, logs, and structures:

```
TSO/ISPF monitor interface: oper:2.1, log:2.2, journal 2.3
SYSJNL file
SYSLOG file
SMF records, Z07 fields of D1B2PSMF Dsect
REQUEST table RCT, EXR fields of D0B2ZEXR Dsect (Application Programming Interface)
USER EXIT parm list fields:
  connection: D1B2PCNX
  session:    D1B2PCNX
  selection:  D1B2RUEX
  transfer:   D1B2RPEX
  journal:    D1B2PJNL
```

Outgoing Connection

Connect:Express requests and supervises the connection in the ANM address space. This includes both the open and close connection operations. Each connection is processed by a subtask attached when it is opened and when it is closed. Connect:Express can manage several simultaneous connections depending on the available resources. Connection parameters are set in the symbolic Partner definition and the SYSIN/SYS-SNA/TCP/X.25 files. The retry connection and restart transfer options are defined by the STIMOC= and STIMEV= fields in the SYSIN file.

The connection is followed by the protocol session. This includes the open and close session operations that are processed in the Connect:Express address space. The protocol session parameters are set in tables referenced by the symbolic Partner definition. The connection user exit, if any, is given control before a call, and can override information or add applicable parameters that Connect:Express does not implement. The connection user exit is also given control after the close connection.

Parameters used for outgoing connections are defined in the following locations:

Location	Parameter
SYSIN file	RQEMAX, APLPFX, DAPMxx, STIMEV, STIMOC, , MCH definitions, TCPORG, TCPPRT
Request	PARTNER NAME, TRANSFER CLASS, TYPE OF LINK
PARTNERS directory+ SYS-SNA/TCP/X25 files	PARTNER NAME, AUTO.RESTART, TYPE OF LINK, LINK TOT.-IN.-OUT., T-FLOW REGULAT., MCHMSC, Addresses
User Exits	D1B2PCNX structure

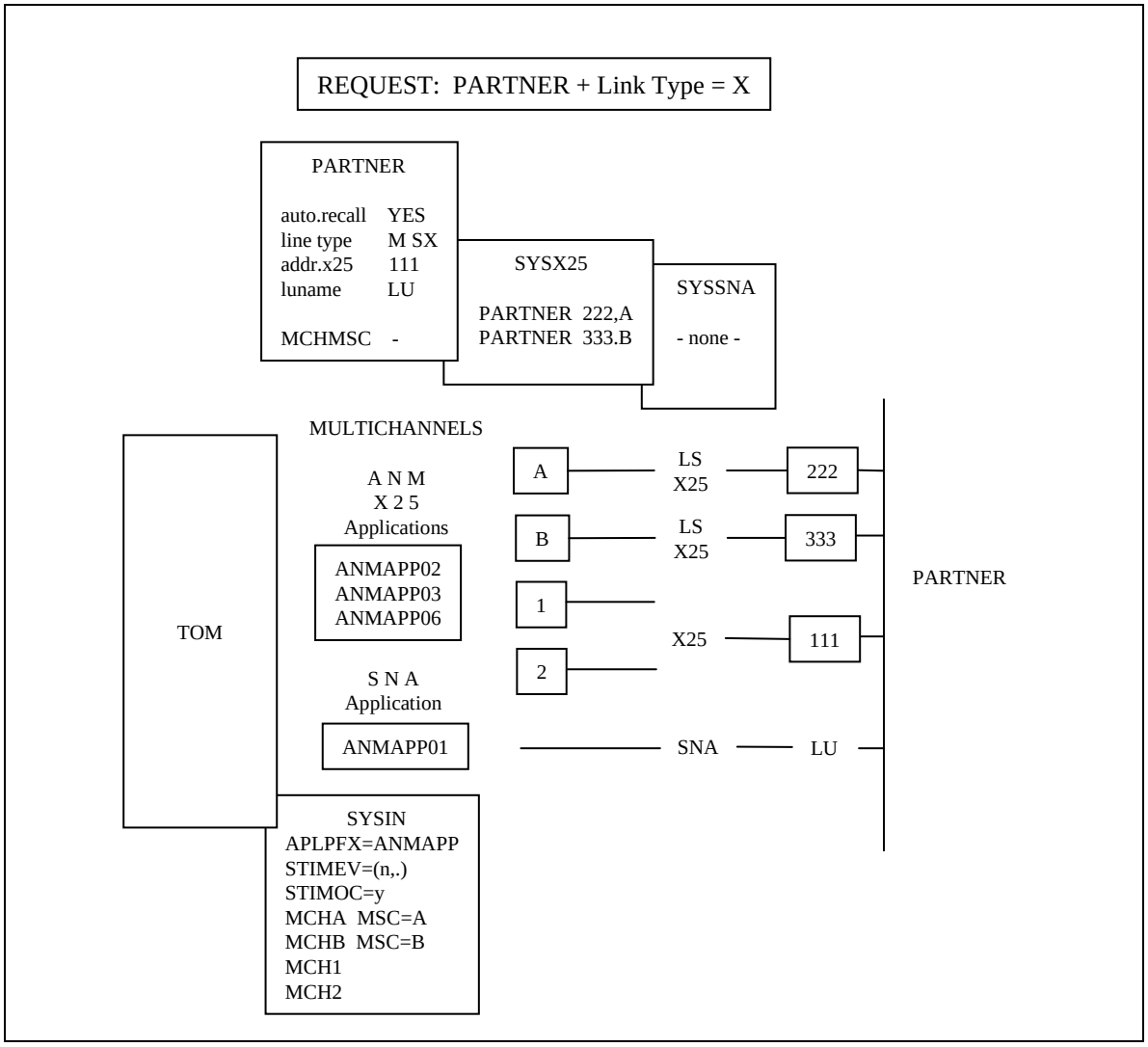
One Partner is assigned a list of possible link types, and you can set the link type when you execute the transfer request. Connect:Express is given a list of network resources and always knows the status of each. Connect:Express chooses the first available link type, respecting the link types list, and starts with the link type of the request, if any. Events are reported in the Connect:Express log file, as shown below.

```
START OF A.N.M IN ASID: 00196 RC=00
START OF A.P.M 01 IN ASID: 00409 RC=00
TOM (GLOBAL) INITIALIZATION COMPLETE 222-PTF2 RUN=(C)
REQUEST 00000146 FILETEST PARTNER SRC=0000 TRC=0000 PRC=0000
REQUEST 00000146 <- TSO00001 ACCEPTED (N)
COMMUNICATION OPENED (O) WITH: PARTNER REQ: 00000146 (X,MCH1,XU451720)
REQUEST 00000146 FILETEST TRANSFER ACCEPTED APM 01 EFF 01
REQUEST 00000146 FILETEST TRANSFER STARTED APM 01 EFF 01
REQUEST 00000146 FILETEST TRANSFER ENDED TRC=0000L APM 01 EFF 01
REQUEST 00000146 TRANSMITTING -> PARTNER, FILE FILETEST NUMBER OF RECORDS
TRANSFER DURATION :00:02:56, RESTART NUMBER : 000 , NUMBER OF S/R
COMMUNICATION CLOSED (O) WITH: PARTNER
REQUEST 00000146 <- TSO00001 COMPLETED
REQUEST 00000146 <- TSO00001 PURGED
```

Connection Failure

The following diagram shows a sample configuration, and illustrates how connection failures are processed using the alternate links that are defined for this partner.

Outgoing Call Resources Configuration



In this example, the Partner is defined in the Connect:Express Partners directory with possible links of SNA or X.25, and the corresponding LUNAME and X.25 address. The X.25 definition points to the remote 111 X.25 address through the non-specific X.25 network. The MCHMSC is not set. The connection will be established from a local multichannel defined for Connect:Express without the MCHMSC parameter. The Automatic Restart option is YES, so that the retry loop will not stop unless the partner is disabled, or if a connection failure occurs.

The Partner has two entries in the SYSX25 file which means that if a connection fails on the first X.25 link, the next one will be processed. The first entry in the SYSX25 points to the A local MCH and the remote 222 X.25 address. This configuration can correspond to the definition of a private X.25 link. The second entry in the SYSX25 points to the B local MCH and the remote 333 X.25 address. No entry exists in SYSSNA for the Partner.

Connect:Express manages four MCHs, as listed below. Their description is found in the SYSIN file.

1. MCHA is identified by A selection code.
2. MCHB is identified by B selection code.
3. MCH1 is a non selected multichannel.
4. MCH2 is a non selected multichannel.

The SYSIN gives Connect:Express the number of call attempts to process before changing links and the time in minutes to wait between each attempt. The number of call attempts is entered in the field, STIMOC=y, and the time in minutes between each call attempt is in the field STIMEV=(n,.).

The first available resource among the following list will be selected for a call: X.25(MCH1 or MCH2,111) X.25(MCHA,222) X.25(MCHB,333) SNA(LU).

A request is sent to Connect:Express to transfer a file with the Partner. The link type X, for X.25, is set in the field of the request. Because the X.25 link type is given at request time, the first call attempt is directed to the X.25 network even though SNA is the first in the link list. If any X.25 link or X.25 handler is inactive at connection time, STIMOC will be reduced to zero and the next link type in the list is selected.

No X.25 network is specified in the MCHMSC field in the directory, so the call is processed on the MCH that is not busy. If the connection is successful, the transfer enters the Session phase.

Connection Retry Processing

If the connection fails, the TRC code is set to 2077 and you see the message “Not Obtained.” Then, Connect:Express starts the retry procedure. Remember that you set retry parameters in the SYSIN file. The number of call attempts is entered in the field, STIMOC=y, and the time in minutes between each call attempt is in the field STIMEV=(n,.). The following screen shows an example of connection retry processing.

```
call attempt:
1. X25 MCH(1 or 2) ----> 111
nok1 (wait n minutes) TRC=2077
2. X25 MCH(1 or 2) ----> 111 TRC=2077
nok2
...
y. X25 MCH(1 or 2) ----> 111 TRC=2077
noky

CONNECTION ERROR --> change link.

call attempt:
1. X25 MCH(A) ----> 222 TRC=2077
nok1 ... TRC=2077
...
y. X25 MCH(A) ----> 222 TRC=2077
noky

CONNECTION ERROR --> change link.

call attempt:
1. X25 MCH(B) ----> 333 TRC=2077
nok1 ... TRC=2077
...
y. X25 MCH(B) ----> 333 TRC=2077
CONNECTION ERROR --> change link.

call attempt:
1. SNA ----> LUNAME TRC=2077
nok1 ... TRC=2077
...
y. SNA ----> LUNAME TRC=2077
noky

CONNECTION ERROR --> first link again and so on.

REQUEST 00000146 FILETEST PARIS SRC=0000 TRC=0000 PRC=0000
REQUEST 00000146 <- TSO00001 ACCEPTED (N)
REQUEST 00000146 SESSION ERROR : Connect NRC=010030 000000
REQUEST 00000146 REJECTED <- PARIS SRC=0000 TRC=2077 PRC=0000
COMMUNICATION NOT OBTAINED-> PARIS REQ:00000146 RETRY IN: 01 MINUTE
REQUEST 00000146 SESSION ERROR : Connect NRC=010030 000000
REQUEST 00000146 REJECTED <- PARIS SRC=0000 TRC=2077 PRC=0000
COMMUNICATION NOT OBTAINED-> PARIS REQ:00000146 RETRY IN: 01 MINUTE
REQUEST 00000146 SESSION ERROR : Connect NRC=010030 000000
REQUEST 00000146 REJECTED <- PARIS SRC=0000 TRC=2077 PRC=0000
COMMUNICATION NOT OBTAINED-> PARIS REQ:00000146 RETRY IN: 01 MINUTE
RETRY WITH PARTNER PARIS, NEW LINK : S
COMMUNICATION OPENED (O) WITH PARIS (S,PARIAP01)
FILETEST TRANSFER ACCEPTED APM 01 EFF 01
...
```

In the example above, Connect:Express waits the number of minutes that you set and retries until the connection is obtained or the maximum number of call attempts is reached. When this happens, Connect:Express moves to the first SYSX25 entry. Then, it selects MCHA if it is active, and sends the call packet with the remote X.25 address = 222.

After attempting the number of connection retries on MCHA, Connect:Express looks for the next SYSX25 entry. Then, it selects MCHB for more connection retries if it is active.

If the connection is still not obtained, the SNA link is used for the next retries. The automatic call retry/restart option is set to YES in the Partner definition, so the loop will continue again with the first X.25 link. To stop the process, you can disable the Partner.

If the automatic retry/restart option was NO, the process stops after the SNA connection failure, the TRC code is set to 2076, and the Partner Disabled message displays. An example of a configuration is shown in *Outgoing Call Resources* on page 4-5.

Outgoing Session

Open and close session operations take place in the TOM address space after connection has been established with the Partner. Session parameters are general session parameters and set in Connect:Express protocol tables T1B2PSxx.

Identification Parameters

Identification parameters are Connect:Express default symbolic parameters or user-specific parameters that are sent through the user connection exit D1B2PCNX data structure. The end session can be one of the following:

- Session reject
- Normal protocol end
- Protocol abort + return codes
- Network disconnection

Parameters used for an outgoing session are defined in the following locations:

Location	Parameters
SYSIN file	DPCSID
PARTNERS directory	Partner name, DPCSID Alias, DPCPSW Alias, Prot. Number, Prot. Tab, ODETTE Identification
Request	Partner name, DPCSID Alias, DPCPSW Alias, Transfer Priority, Transfer Class, API fields
TABLES session (T1B2PSxx)	Message, SYNCHRO, Window, CRC, Direction of session
T1B2PCOD	Protocol, PRC
User Exits	D1B2PCNX structure

The screen below shows an extract from the logging file about an outgoing session.

```
REQUEST 00000146 FILETEST PARIS SRC=0000 TRC=0000 PRC=0000
REQUEST 00000146 <- TSO00001 ACCEPTED (N)
COMMUNICATION OPENED (O) WITH: PARIS (S,PARIAP01)
FILETEST TRANSFER ACCEPTED APM 01 EFF 01
...
COMMUNICATION CLOSED (O) WITH: PARIS
```

Session Failure

If a parameter is rejected during the negotiation, the protocol return code (PRC) is set to a value defined by the protocol specifications. This PRC is compared to the list of PRCs defined in the T1B2PCOD table for this protocol. If the PRC matches one on the list, the call retry procedure is activated, and the connection is postponed. If the PRC is not on the list, the connection is rejected.

Outgoing Selection

Selection takes place in an APM address space after a session has been established with the Partner, and includes the initialization and termination of transfer. Selection parameters are listed below.

Parameter	Description
General presentation parameters	They are set in Connect:Express protocol tables T1B2PSxy and T1B2PPxx.
Identification parameters	They are Connect:Express symbolic parameters and allocation parameters from the directories or the request. They can be changed with user exits by sending parameters through selection and beginning of transfer exits using the D1B2RUEX data structure.
Retry parameters	They are set in the Connect:Express retry table T1B2PCOD.

When a session has been established, several transfers can take place in both directions. Connect:Express transmits the session context to the APM that was previously selected. The APM attaches a transfer sub-task or effector, and then the effector asks Connect:Express about transfers to be done and executes them. When Connect:Express has no more transfers waiting for selection, the APM returns control to the Connect:Express address space to process the end session.

Depending on the transfer protocol, the user can establish a session to send or receive a file in either direction. Some protocols provide read and write commands, while others provide only a write command.

Selection operations are managed from the symbolic file definition and depend on the transfer direction. You can control some operations using the Application Program Interface (API), while some operations are specific to the file organization for VSAM, PDS and TAPE files. Events are reported in Connect:Express log file, as shown below.

```
REQUEST 00000146 FILETEST PARTNER SRC=0000 TRC=0000 PRC=0000
REQUEST 00000146 ← TSO00001 ACCEPTED (N)
COMMUNICATION OPENED (O) WITH: PARTNER REQ: 00000146 (X,MCHA,LUSMN08)
REQUEST 00000146 FILETEST TRANSFER ACCEPTED APM 01 EFF 01
REQUEST 00000146 FILETEST TRANSFER STARTED APM 01 EFF 01
REQUEST 00000146 FILETEST TRANSFER ENDED TRC=0000L APM 01 EFF 01
REQUEST 00000146 TRANSMITTING → PARTNER, FILE FILETEST NUMBER OF RECORDS
TRANSFER DURATION :00:02:56, RESTART NUMBER : 000 , NUMBER OF S/R
COMMUNICATION CLOSED (O) WITH: PARTNER
REQUEST 00000146 ← TSO00001 COMPLETED
REQUEST 00000146 ← TSO00001 PURGED
```

If an error occurs during selection or transfer, return codes are first reported to user exits, then to the SYSLOG and SYSJNL, and finally to the TSO/ISPF interface.

Selection Failure

If a parameter is rejected during the negotiation, the protocol return code (PRC) is set to a value defined by the protocol specifications. This PRC is compared to the list of PRCs defined in the T1B2PCOD table for this protocol. If this PRC matches one on the list, the call retry procedure is activated, and the request is postponed. If the PRC is not on the list, the request is rejected.

Outgoing Transmission

Parameters used for outgoing transmission are defined in the following locations:

Location	Parameters
SYSIN file	WRKUNT, UEXJNL, SMFREC, UPRFCT
Session tables (T1B2PSxx)	Message
Presentation tables (T1B2PPxx)	Compression, Record processing exits such as translation
Files directory	Symbolic file name, Direction, Remote DSN, Presentation Table Number, parameters for transmission, data set name
Request	Symbolic file name, Partner name, Direction, Data set name
Request extension	Origin, Destination, Remote DSN, API fields
Ad Hoc request	Local and remote parameters
T1B2PCOD	Protocol, PRC
User Exits	D1B2PUEX and D1B2PPEX structures

Initialization

When the session has been established, the process is transferred from the TOM address space to the APM address space. The APM asks Connect:Express for transfers to process, and Connect:Express selects the next eligible request and starts the Beginning of Transmission command. Then Connect:Express sends the APM all necessary information, including allocation parameters, user exit names, record processing, and security conditions found in the Request entry or the File Directory entry.

The Beginning of Transmission user exit or the Application Server, if any, is given control to manage file allocation. It can also override information or add applicable parameters that Connect:Express does not implement.

The checkpoint file is allocated for this request and exists until the request is purged at the successful end of transfer. After the Partner accepts the request, Connect:Express indicates that the transfer was accepted. Information transmitted to the Partner includes:

Type	Description
Symbolic	Information processed by Connect:Express
Application	Information from the user
Physical	File attributes processed by Connect:Express or the user (depending on protocol)

One or more protocol messages are exchanged with Partners for identification, data presentation, and transfer restarting negotiations. When you are ready to send data, Connect:Express is informed that the transfer has started.

Termination

One or more protocol messages are exchanged with a Partner at the end of transfer for security purposes. At the end of protocol acknowledgments, the End of Transmission user exit or the Application Server, if any, is then given control.

At this point, the transfer has been accepted by the receiver and the file was sent to the receiving application. Any error detected by the sender in the termination exit is not indicated to the partner. The request checkpoint file is deleted, and the file is unallocated. Connect:Express is then informed about a new request. The next request can be a transmission or reception, depending on the protocol and the session parameters that were negotiated.

When receiving notification that the transfer has successfully ended, Connect:Express adds an entry in the Journal file and sends the journal record to the user. It adds a record in the SMF file, if required, and starts the End of Transmission command and then purges the request. Then, Connect:Express sends the APM the next request to process or notice to end the session.

Outgoing Reception

Parameters used for outgoing transmission are defined in the following locations:

Location	Parameters
SYSIN file	WRKUNT, UEXJNL, SMFREC, UPRFCT
Tables session (T1B2PSxx)	Message
Tables presentation (T1B2PPxx)	Compression, Record processing exits such as translation
Files directory	Symbolic file name, Direction, Remote DSN, Presentation Table Number, parameters for transmission, data set name
Request	Symbolic file name, Partner name, Direction, Data set name
Request extension	Origin, Destination, Remote DSN, API fields
Ad Hoc request	Local and remote definitions.
T1B2PCOD	Protocol, PRC
User Exits	D1B2PUEX and D1B2PPEX structures

Initialization

After the session has been established, the process is transferred from the TOM address space to the APM address space. The APM asks Connect:Express for transfers to process, and Connect:Express selects the next eligible request and starts the Beginning of Reception command. Then Connect:Express sends the APM all necessary information, including user exit names, record processing, and security conditions found in the Request entry or the File Directory entry.

The selection user exit or the application server, if any, is given control to manage file allocation. It can also override information or add application parameters that Connect:Express does not implement.

Information transmitted to the partner includes:

Information Type	Description
Symbolic	Information processed by Connect:Express
Application	Information from you (user)

Information returned by the partner includes:

Information Type	Description
Symbolic	Information processed by Connect:Express
Application	Information from you (user)
Physical	File attributes processed by Connect:Express or you (depending on protocol). Some protocols carry the file attributes like record format, record length and file space. In this case these allocation parameters are not mandatory in the file definition for reception. If the protocol doesn't provide these parameters, then they are mandatory, unless the file is pre-allocated.

When the Partner accepts the transfer, allocation parameters are found in the protocol command and sent to Connect:Express. Then, Connect:Express issues a log message that the transfer was accepted by the Partner. Finally, Connect:Express allocates the file if it is set to do file allocation.

The beginning of reception user exit or the application server, if any, is given control and can process file allocation, if the allocation rule is 3 or A.

The checkpoint record is allocated for this request and exists until the request is purged at the successful end of transfer. When the file space is known, no checkpoint is allocated if it is less than or equal to one kilobyte. One or more protocol messages are exchanged with the Partner for identification, data presentation, and transfer restarting negotiations. When receiving the data, the Connect:Express TOM is informed that the transfer has started.

Termination

At the end of transfer, one or more protocol messages are exchanged with the Partner for security reasons. The end of reception user exit or the application server, if any, is then given control.

Being on the receiving side, the termination user exit decides if the transfer was successful. Then, the request checkpoint record is deleted, and the file is unallocated. At this point, the file is processed by the application, and the Partner is informed using the protocol acknowledgment command.

When Connect:Express receives notification that a transfer has successfully ended, it adds an entry in the journal file and sends the journal record to the user with a record in the SMF file, if required. It also starts the End of Reception command, and purges the request. Connect:Express either sends the next request to the APM or notice to end the session. The new request can be transmission or reception, according to the protocol and the session parameters that were negotiated.

Incoming Transfers

A remote Partner can send a request of transfer to Connect:Express through a network. The incoming call is detected by the ANM network handler and then sent to Connect:Express. The following information is mandatory, and must be provided at connection or selection time. This information must match the symbolic partner and file name entries in Connect:Express directories and SYS-SNA/TCP/X.25 files.

- Symbolic Partner name
- Symbolic Partner password
- Network identification
- Symbolic File name
- Transfer direction

Network identification and symbolic passwords must match the Partner definition, and the transfer direction and symbolic partner must match the file definition. The request is identified by a request number and queued in the Request Control Table with the status set to INIT.

If Connect:Express resources are all available and the number of active sessions for the Partner has not reached the maximum allowed, the request is assigned a transfer resource with the class found in the Partner definition.

Parameters are stored in the following locations:

```
SYSIN file
PARTNERS   directory + SYS-SNA/TCP/X25 files
FILES      directory
REQUEST    fields
TABLES     session, data presentation
USER EXITS
```

Information is reported in the following files, logs, and structures:

```
TSO/ISPF monitor interface: oper:2.1, log:2.2, journal 2.3
SYSJNL file
SYSLOG file
SMF records, Z07 fields of D1B2PSMF Dsect
REQUEST table RCT, EXR fields of D0B2ZEXR Dsect (Application Programming Interface)
USER EXIT parm list fields:
  connection: D1B2PCNX
  session:    D1B2PCNX
  selection:  D1B2RUEX
  transfer:   D1B2RPEX
  journal:    D1B2PJNL
```

Incoming Connections

An incoming call must be identified by the link type, transfer protocol type, and symbolic Partner name. The incoming call is detected by the ANM which identifies the link type and sends Connect:Express the network information. Connection takes place in the TOM address space and includes the open and close session operations. Each connection is processed by a subtask attached when the session opens, and again when the session closes. Connect:Express can manage a number of simultaneous connections, depending on available resources. Connect:Express creates a new request number after a transfer resource has been allocated, and then waits for the first transfer protocol message.

Connect:Express can identify the active protocol (PeSIT, ODETTE-FTP, ETEBAC3...) based on the specifications of each protocol listed in the table below.

Protocol	Description
PeSIT	The pre-connection message carries a symbolic Partner name that is searched for in the directory.
ODETTE-FTP	The X.25 call packet must be answered with an ODETTE-FTP READY message by a server. The ODETTE-FTP incoming call is identified through the user data field (UDF) found in the X.25 packet or the Sub address. They must match the ODTSAD and ODTUDF fields of the SYSIN file.
ETEBAC3	The ETEBAC3 card is recognized from the first character A or R.

When Connect:Express has identified the protocol, it can determine the Partner with its symbolic name, its network address, or a bypass Keyword.

Connect:Express looks for the symbolic Partner name or network address in the symbolic Partners directory until a Partner definition is found with the correct protocol. The network address is controlled unless '*' is found in the remote address field. If this address is different from the current, Connect:Express looks for the network address complementary file (SYS-SNA/TCP/X.25), if any. If a '*' is in the list from the DIRECTORY+SYS-SNA/TCP/X.25 REM address fields of one Partner, no input control is done for the corresponding link type.

Input Control Bypass

Specific bypass keys point to a general Partner profile, as shown below. A bypass key is used as a default Partner if no Partner is found in the directory for the current incoming call. For example, X25BYPASETB3 is used for an unidentified incoming call with ETEBAC3 protocol.

```
TOM4200-------- LIST OF BYPASS FOR CONNECTION ------------------------
OPTION ===>

TOM3        ^-< COMMANDS -PF3- END
83BA  ACTIVE GLOBAL

    CONNECTION      PARTNER

    X25BYPASETB5 -> ETEBAC5
    X25BYPASETB3 -> ETEBAC3
    X25BYPASXPAD -> TEDPAD
    LU2BYPASS    -> TED3270
    FTPBYPASS    -> FTPROF
```

Connection and session operations are managed from the symbolic Partner definition. The connection user exit, if any, is given control after the call, and receives the application parameters that Connect:Express does not implement. It is also given control after the release of the protocol session. Paremeters used for incoming connection are defined in the following locations.

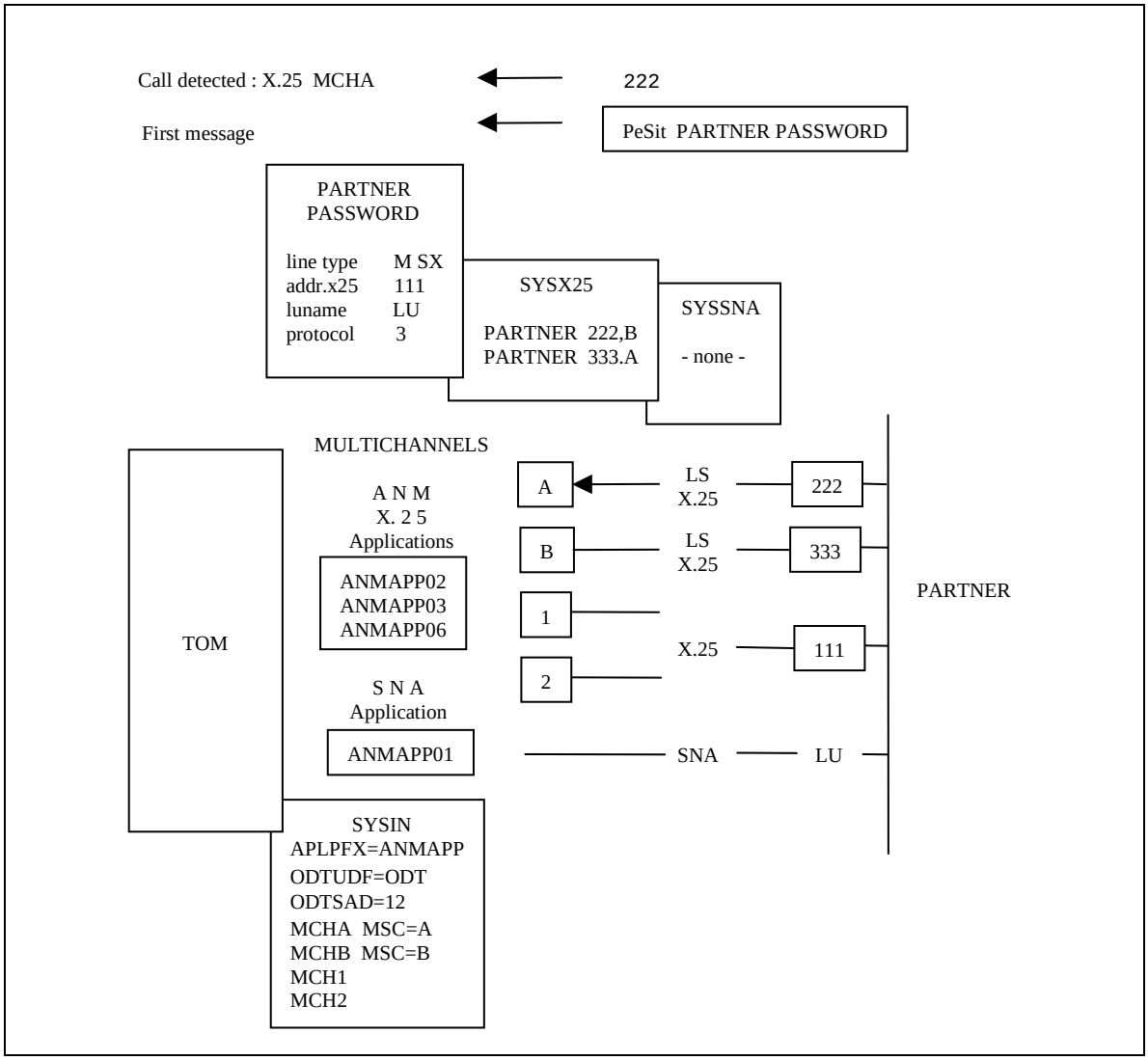
Location	Parameters
SYSIN file	RQEMAX, DAPMxx, TCPORG, TCPPRT, APLPFX, ODTUDF, ODTSAD, MCH definitions
PARTNERS directory + SYS-SNA/TCP/X25 files	PARTNER NAME, TRANSFER CLASS, TYPE OF LINK, LINK TOT.-IN.-OUT., T-FLOW REGULAT., Addresses
User Exits	D1B2PCNX structure

Events are reported in the Connect:Express SYSLOG file, as shown below.

```
START OF A.N.M IN ASID: 00196 RC=00
START OF A.P.M 01 IN ASID: 00409 RC=00
TOM (GLOBAL) INITIALIZATION COMPLETE 222-PTF2 RUN=(C)
COMMUNICATION OPENED (I) WITH: PARTNER REQ: 00000146 (X,MCHB,XU461221)
REQUEST 00000146 FILETEST TRANSFER ACCEPTED APM 01 EFF 01
REQUEST 00000146 FILETEST TRANSFER STARTED APM 01 EFF 01
REQUEST 00000146 FILETEST TRANSFER ENDED TRC=0000L APM 01 EFF 01
REQUEST 00000146 RECEIVING <- PARTNER, FILE FILETEST NUMBER OF RECORDS
TRANSFER DURATION :00:02:56, RESTART NUMBER : 000 , NUMBER OF S/R
COMMUNICATION CLOSED (I) WITH: PARTNER
REQUEST 00000146 <- PARTNER COMPLETED
REQUEST 00000146 <- PARTNER PURGED
COMMUNICATION OPENED (I) WITH: LU2PROF REQ: 00000148 (T,LUPC3270)
REQUEST 00000148 FILETEST TRANSFER ACCEPTED APM 01 EFF 01
REQUEST 00000148 FILETEST TRANSFER STARTED APM 01 EFF 01
REQUEST 00000148 FILETEST TRANSFER ENDED TRC=0000L APM 01 EFF 01
REQUEST 00000148 RECEIVING <- PC3270 , FILE FILETEST NUMBER OF RECORDS
TRANSFER DURATION :00:02:56, RESTART NUMBER : 000 , NUMBER OF S/R
COMMUNICATION CLOSED (I) WITH: LU2PROF
REQUEST 00000148 <- PC3270 COMPLETED
REQUEST 00000148 <- PC3270 PURGED
```

Incoming Call Identification Example

The following diagram shows an example of an incoming call.



In this example, an incoming call is detected by the ANM on an X.25 environment from the remote 222 address. No user data field (UDF) or subaddress is received as protocol identification in the call packet. For example, UDF='ODT' is an identification from the ODETTE-FTP protocol declared in the SYSIN file. After sending the incoming call confirm packet, Connect:Express waits for the first protocol message. For example, if using PESIT, Connect:Express recognizes the PeSIT pre-connection message and looks for the corresponding symbolic Partner definition in its directory.

The Partner is defined in the Connect:Express Partners directory with possible SNA, X.25, or TCP/IP links and the corresponding address. For example, the X.25 definition points to the remote 111 X.25 address. The Partner has two entries in SYSX25:

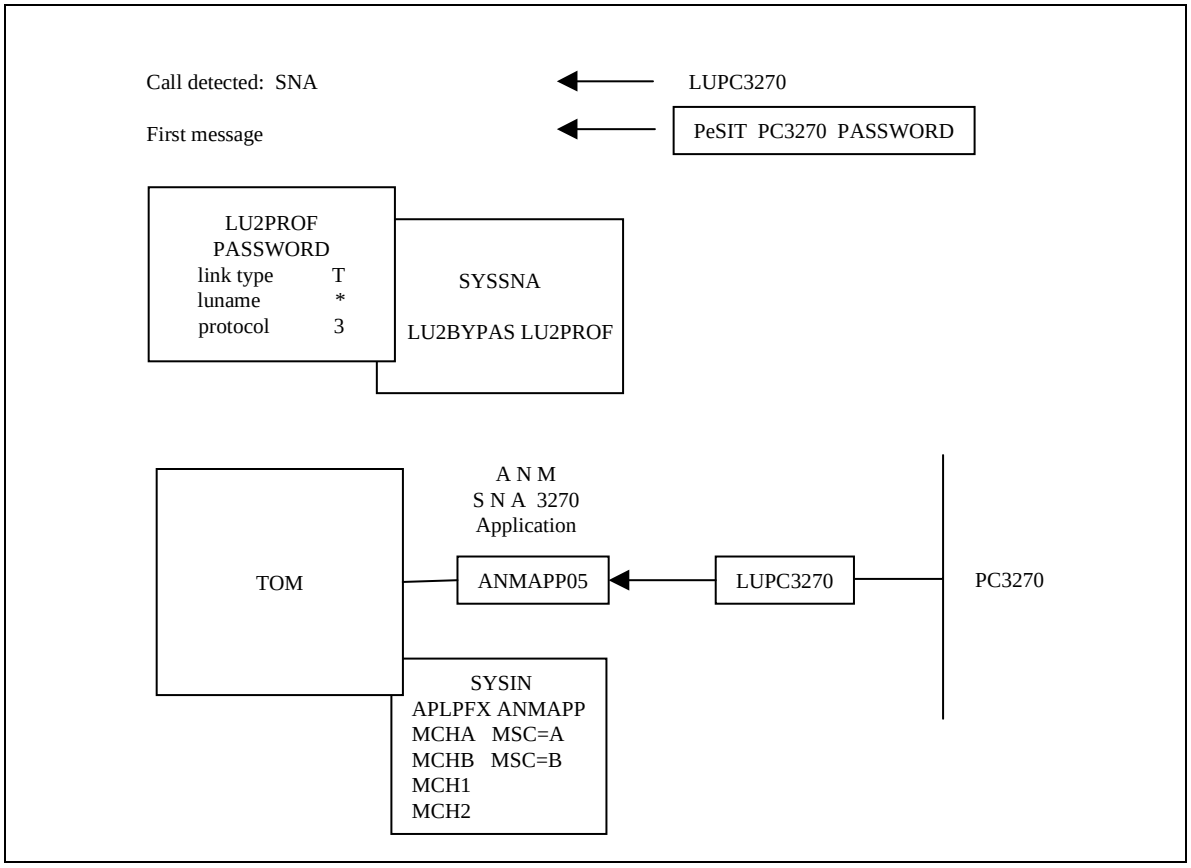
1. The first entry in SYSX25 points to the remote 222 X.25 address.
2. The second entry in SYSX25 points to the remote 333 X.25 address.

Connect:Express compares the network address received with the list made of the 111 X.25 address from the directory, the 222 X.25 address from SYSX25, and the 333 X.25 address from SYSX25.

Connect:Express verifies that the protocol and password match, and then verifies that the X.25 charge rule is correct. If the partner is OK and Connect:Express resources are available, the incoming call is accepted.

Incoming Call Identification Bypass Example

The following diagram shows an example of an incoming call identification bypass.



In this example, an incoming call is detected by the ANM, and the 3270 handler logon exit is scheduled with CINIT queued from the remote LUPC3270 terminal.

Connect:Express creates a new request without determining the Partner name and waits for the first message. Connect:Express must recognize that the PeSIT protocol is in use and that the symbolic Partner name is PC3270, by looking for this in its directory.

When no entry is found, Connect:Express looks for the LU2BYPAS keyword in the SYSSNA file. This keyword points to the symbolic Partner name LU2PROF that is taken as a 3270 Partner session profile. Connect:Express verifies that the profile protocol and password match, and the name PC3270 is processed as the Requestor name associated with the transfer.

Incoming Session

Open and close session operations take place in the TOM address space after connection has been established with the Partner. Session parameters are listed below.

Parameter	Description
General session	They are set in Connect:Express protocol tables T1B2PSxx.
Identification	They are Connect:Express default symbolic parameters or user-specific parameters to be passed through user connection exit D1B2PCNX data structure.

An end session can be any of the following:

- Session reject
- Normal protocol end
- Protocol abort + return codes
- Network disconnection

Parameters used for incoming sessions are defined in the following locations:

Location	Parameters
SYSIN file	DPCSID
Partners directory	PARTNER NAME, DPCSID ALIAS, DPCPSW ALIAS, PROT.NUMBER, PROT. TAB., ODETTE IDENTIFICATION
Tables session (T1B2PSxx)	MESSAGE, SYNCHRO, WINDOW, CRC, DIRECTION OF SESSION
User exits	D1B2PCNX structure

The screen below shows an example of log information for an incoming session.

```
COMMUNICATION OPENED (I) WITH: PARIS (X,MCH451,V4516507)
REQUEST 00000146 FILETEST PARIS SRC=0000 TRC=0000 PRC=0000
FILETEST TRANSFER ACCEPTED APM 03 EFF 14
...
COMMUNICATION CLOSED (I) WITH: PARIS
```

Incoming Selection

Selection takes place in the APM address space after the session has been established with the Partner. This is done during the initialization and termination of transfer. Selection parameters are listed below.

Parameter	Description
General	They are set in Connect:Express protocol tables T1B2PPxx.
Identification	They are Connect:Express symbolic parameters and allocation parameters from the directories or user-specific parameters to be passed through selection and beginning of transfer exits D1B2RUEX data structure.

When a session has been established, several transfers can take place in either direction. Connect:Express transmits transfers to the APM. Then, the APM attaches a transfer sub-task or Effector that waits for incoming transfer requests.

Depending on the transfer protocol, the caller can establish a session to send or receive a file in either direction. Some protocols provide read and write commands, while others provide only a write command.

Selection operations are managed from the symbolic File definition and depend on the transfer direction. Therefore, any transfer protocol whose Partner and File identification conventions do not have 8-character symbolic names must be interpreted by the application before the request is sent to Connect:Express.

Connect:Express provides symbolic security, which means that a File cannot be transferred with unknown Partners. The transfer origin Partner can be different from the session origin Partner. Depending on the Partner type, the transfer origin Partner can be controlled. The following table identfies the Partner types that can be controlled.

Partner Type	Description
Connect:Express	Transfer Partner is controlled and reported in the JOURNAL REQUESTOR field.
OTHER	Transfer Partner is controlled and reported in the JOURNAL REQUESTOR field.
APPLICATION	Transfer Partner is not controlled, and the session Partner is set in the JOURNAL REQUESTOR field.

The file identification can be an application structure more than eight characters long. In this case, the Selection User exit must return a symbolic file name. The exit is called before the request is sent to Connect:Express. Some operations are user-controlled while other operations are specific to the file organization (VSAM, PDS, and TAPE files). The events are reported in Connect:Express SYSLOG file, as shown below.

```
COMMUNICATION OPENED (I) WITH: PARTNER REQ: 00000146 (X,MCHA,XU461234)
REQUEST 00000146 FILETEST TRANSFER ACCEPTED APM 01 EFF 01
REQUEST 00000146 FILETEST TRANSFER STARTED APM 01 EFF 01
REQUEST 00000146 FILETEST TRANSFER ENDED TRC=0000L APM 01 EFF 01
REQUEST 00000146 RECEIVING <- PARTNER, FILE FILETEST NUMBER OF RECORDS
TRANSFER DURATION :00:02:56, RESTART NUMBER : 000 , NUMBER OF S/R
REQUEST 00000146 <- PARTNER COMPLETED
REQUEST 00000146 <- PARTNER PURGED
COMMUNICATION CLOSED (I) WITH: PARTNER
```

If an error occurs during connection or transfer, return codes are first reported to user exits, then to the SYSLOG and SYSJNL, and finally to the TSO/ISPF interface.

Note: No record is added in the Connect:Express journal if an error occurs during selection operation before the transfer has started.

Incoming Transmission

Parameters used for incoming transmission are defined in the following locations.

Location	Parameters
SYSIN file	WRKUNT, UEXJNL, SMFREC, UPRFCT
Tables session (T1B2PSxx)	Message
Tables presentation (T1B2PPxx)	Compression, Record processing EXITS (example: Translation)
Files directory	Symbolic file name, direction, presentation table number, parameters for transmission, data set name
User exits	D1B2PUEX and D1B2PPEX structures

Initialization

When receiving the external request from the network, the application server or selection user exit is given control to process information about transfer identification. The following information is received from the Partner:

Information Type	Description
Symbolic	Information processed by Connect:Express.
Application	Information from the user.

The following information is returned to the Partner:

Information Type	Description
Symbolic	Information processed by Connect:Express.
Application	Information from the user.
Physical	File attributes processed by the partner (depending on protocol). Some protocols carry the file attributes including record format, record length and file space.

When receiving the external request from the Effector, Connect:Express controls the demand which can be an inquiry or normal demand. If it is an inquiry, then Connect:Express looks for a corresponding hold request. If it is a normal demand, then Connect:Express controls the demand through the symbolic file name in the directory. After all verifications are done, Connect:Express indicates that the transfer was accepted.

Connect:Express starts the Beginning of Transmission command and then sends the Effector all necessary information, including allocation parameters, user exit names, and record processing and security conditions found in the File Directory entry.

The Beginning of Transmission user exit or the application server is given control to manage file allocation. It can also override information or add applicable parameters that Connect:Express does not implement.

The checkpoint record is allocated for this request and exists until the request is purged at the successful end of transfer. When the file space is carried by the current transfer protocol fields, no checkpoint is allocated if it is less than or equal to one kilobyte.

One or more protocol messages are exchanged with the Partner for identification, data presentation, and transfer restarting negotiations. When Connect:Express is ready to send the data, it is informed that the transfer has started.

Termination

At the end of transfer, one or more protocol messages are exchanged with the Partner for security reasons. At the end of the protocol acknowledgment, the End of Transmission user exit or the application server, if any, is given control. At this point, the transfer has been accepted by the receiver and the file was sent to the receiving application. Any error detected by the sender in the termination exit is not indicated to the partner. The request checkpoint record is deleted, and the file is unallocated. Then, Connect:Express is informed of the transfer end.

When receiving notification that the transfer ended, Connect:Express adds an entry in its journal file and sends you the journal record. Connect:Express also adds a record in the SMF file if required, starts the End of Transmission command, purges the request, and then sends the effector the current request number. The effector waits for the session close command or the next transfer selection.

Incoming Reception

Parameters used for incoming reception are defined in the following locations:

Location	Parameters
SYSIN file	WRKUNT, UEXJNL, SMFREC, UPRFCT
Tables session (T1B2PSxx)	Message
Tables presentation (T1B2PPxx)	Compression, Record processing EXITS (example: Translation)
Files directory	Symbolic file name, direction, presentation table number, parameters for transmission, data set name
User exits	D1B2PUEX and D1B2PPEX structures

Initialization

When receiving the external request from the network, the application server or selection user exit is given control to process information about transfer identification. The following information is received from a Partner.

Information	Description
Symbolic	Information processed by Connect:Express.
Application	Information from the user.
Physical	File attributes processed by Connect:Express or the user (depending on protocol). Some protocols carry file attributes like record format, record length and file space. In this case, these allocation parameters are not mandatory in the file definition for reception. If the protocol doesn't provide these parameters, then they are mandatory unless the file is pre-allocated.

When receiving the external request from the effector, Connect:Express controls the symbolic file name in the directory. If Connect:Express is set to do the file allocation, it is done at this time, then Connect:Express starts the Beginning of Reception command. After all verification has been done, Connect:Express notifies you that the transfer was accepted and sends the effector all necessary information, including user exit names, and record processing and security conditions found in the file directory entry.

The beginning of reception user exit or the application server, if any, is given control and can process file allocation if the allocation rule is 3 or A.

The checkpoint record is allocated for this request and exists until the request is purged at the successful end of transfer. One or more protocol messages are exchanged with the Partner for identification, data presentation and transfer restarting negotiations. When data is received, Connect:Express is informed that the transfer has started.

Termination

At the end of transfer, one or more protocol messages are exchanged with the Partner for security reasons. The end of reception user exit or the application server, if any, is then given control. Being on the receiving side, the termination user exit decides if the transfer was successful. Then, the request checkpoint record is deleted, and the file is deallocated. At this point, the file will be processed by the application, and the partner is informed with the last protocol acknowledgment command. Then, Connect:Express receives the message that the transfer ended.

When receiving information that the transfer ended successfully, Connect:Express adds an entry in the journal file and sends you the journal record. Then Connect:Express adds a record in the SMF file if required, starts the End of Reception command, purges the request, and sends the effector the current request number. The effector waits for the session close command or the next transfer selection.

Data Exchange

During connection and selection negotiations, Partners have previously agreed about the following transfer conditions:

- Network protocol message length
- CRC activation
- Synchronization (size and window)
- Compression

Connect:Express has set up the local environment from its parameters. This includes initializing record processing, like selecting records to transfer and configuring ASCII-EBCDIC transcription. Connect:Express also allocates and opens the file to transfer and the request checkpoint file.

The data exchange process is performed in the APM address space and includes the checkpoint process. During this process, the transmitter sends synchronization points, and the receiver acknowledges the synchronization points. The synchronization parameters indicate the size or amount of data for one point, and the window indicates the number of synchronization points to send before waiting for an acknowledgement. Synchronization points enable you to restart a transfer at a certain point if a failure occurs during the transfer. In this case, the receiver decides which synchronization point to use to restart the transfer. The transmitter is allowed to send as many synchronization points as defined in the synchronization window. The current window table contains active synchronization points that the receiver did not acknowledge. When the window is full, the transmitter waits for the next ackowledgment from the receiver. Each time the transmitter receives an acknowledgment from the receiver, it updates the window table.

Transfer protocols process logical records. These logical records are blocked into physical records on disk and into network messages. Connect:Express must perform different actions depending on the transfer direction.

Transmission

During transmission, Connect:Express reads physical blocks from disk and extracts logical records, makes any necessary changes, and concatenates logical records into the current protocol message before sending it to the network. Transmission goes through the following process.

Connect:Express, or a private I/O exit, reads one physical block, and then it sends every logical record to record processing user exit1, exit2, and exit3. The logical record is compressed and concatenated into the protocol message to be sent. When the protocol message buffer is full, or the synchronization point is reached, Connect:Express sends the data to the partner after performing the CRC algorithm, if active. If the synchronization point is reached, the synchronization point context is added to the window table and a synchronization message is sent. The current window table, with all active synchronization points, is saved into the request checkpoint file.

Acknowledgment of one synchronization point causes the APM to update the window table and save it to the request checkpoint file. Then, the APM informs the Connect:Express monitor (TOM) of the progression of the transfer so it can update the request table entry.

Note: When the record format (RECFM) is Undefined (U) or Variable Spanned (VS, VBS), one physical block is considered one logical record.

When the HPO option is used in the T1B2Pxx presentation table, the file is transferred in block mode regardless of the record format. One physical block is considered one logical record.

Reception

During reception, Connect:Express extracts logical records from protocol messages received from the network, makes any necessary changes, and concatenates them into physical blocks before writing them to disk. Reception goes through the following process.

Connect:Express receives one protocol message, and then decompresses every logical record and sends it to record processing user exit3, exit2, and exit1. The record is then concatenated into the current physical block. When the I/O buffer is full, Connect:Express or a private I/O exit writes the physical block on disk. When receiving a synchronization point, Connect:Express saves all counts and pointers needed to restart the transfer in the request checkpoint file. Then, the APM informs the Connect:Express monitor (TOM) of the progression of the transfer so it can update the request table entry.

Note: If the HPO option is used, the record received is considered a physical block. The blocking factor defined locally is used if it is different from the remote's blocking factor. The block received is restructured into the local block structure.

Translating Data

When transferring data with ASCII computers, you can use EBCDIC to ASCII translation when transmitting and ASCII to EBCDIC translation when receiving. Translation is done by a middle of transfer user exit declared in the presentation table T1B2PP05. You activate Connect:Express standard translation by entering number 05 in the presentation protocol field of the Connect:Express Files directory.

The transfer protocol can carry a data type field, but no negotiation of data type is done by the Connect:Express standard L1APM050 translation user exit. When you enter 05 in the Presentation Protocol field, the data is translated based on the transfer direction. When you want specific translation tables or processing, you must write your own exit and use one of the free Presentation Tables (09 to 24). The translation L1APM050 source module is provided as an example in the *SAMPLIB* (EX£APM50).

Chapter 5

PeSIT and ODETTE-FTP Protocols

This chapter discusses the PeSIT and ODETTE-FTP protocols. It describes how each parameter is used, how files are exchanged with Partners, and how information is verified. The relationship between internal events and external events is also discussed.

The PeSIT Protocol

The PeSIT protocol is the primary protocol used by Connect:Express. There are two PeSIT levels, D and E, that are defined by protocol numbers 3 and 5. These are declared in the Partners directory, and the corresponding session tables, T1B2PS3x and T1B2PS5x.

The E-level partner provides level negotiation. This means that a D-level partner can call Connect:Express, even if it is defined as an E-level partner in the Connect:Express directory. However, the E-level partner cannot call the D-level partner because PeSIT-D does not provide this capability.

Half way between PeSIT-D and PeSIT-E is the PeSIT-D' sublevel. It is used for specific Connect:Express facilities such as PDS files and file attributes exchange. It is only available between Connect:Express partners.

PeSIT-D’ and PeSIT-E provide for two way communication sessions. PeSIT-D only provides for one way sessions. All levels of the PeSIT protocol provide a user field parameter called P99. This field is involved in specific Connect:Express facilities like transferring PDS or VSAM files, and exchanging file attributes and user identification. Most of these facilities need a long P99 rather than the short one carried by PeSIT-D. Connect:Express provides the following session tables:

Table	Protocol	Transfer Direction	P99 Length
T1B2PS30	PeSIT D	both ways	p99 length = 64 characters
T1B2PS31	Standard PeSIT D	one way	p99 length = 64 characters not used
T1B2PS32	PeSIT D'	both ways	p99 length = 254 characters
T1B2PS52	PeSIT E	both ways	p99 length = 254 characters

Note: T1B2PS31 is mandatory for connections with non-Connect:Express partners using the PeSIT-D protocol.

PeSIT Commands

PeSIT commands and parameters are identified by a hexadecimal structure in the format (code, length). This makes it possible to compress parameters when sending data. If a parameter is not entered, default values are used. The general structure of a PeSIT command is shown below.

```
|length|command |org|dest|p1|l1|  value |p2|l2|     value    | ...
| 000B | 4020   | 01|  02|03|04|41313233|04|07|42313233343536| ...
```

The length of the command is 11 characters (hexadecimal), the command identifier is 4020, and org and dest values are set by partners to identify some internal channel. For Connect:Express, the APM number + the effector number are part of this channel number. The parameter in the first position is P03, which is 4 characters long, and has a value of ‘A123’ which is ASCII coded. The next parameter is P04, which is 07 characters long, and has the value ‘B123456.’

Communicating with non-Connect:Express PeSIT Partners

Some Connect:Express features are not supported by other PeSIT software, and the PeSIT 'Read' command may be processed differently as well. For example, when you call a non-Connect:Express partner to receive a file, you must use the Reception/Inquiry request type. When you are called by a non-Connect:Express partner to transmit a file, you must use the Transmission/Hold request type. The table below shows the relationship between protocol levels, the Connect:Express version number, the session table number, the use of Pi99 and Pi37 parameters, and the partner type. Keywords have been added in version 4.2.0 for using Pi99 with partners of other types.

Protocol Level	Session Table	TRF Way	P99 (Level,Length)	P99 Services (protocol facilities)	P37	Partner Type
PeSIT-D	T1B2PS31	one	1, L=64		&EXTLAB	other
PeSIT-D	T1B2PS30	both	1, L=64	DSN (&EXTDSN, &USRVAR)	&EXTLAB	C:X
PeSIT-D’	T1B2PS32	both	2, L=254	PDS, attributes, VSAM, DSN (&EXTDSN, &USRVAR), &EXTDAT, &EXTREQ	&EXTLAB	C:X
PeSIT-E	T1B2PS52	both	2, L=254		&EXTLAB	other
PeSIT-E	T1B2PS52	both	2, L=254	PDS, attributes, VSAM DSN (&EXTDSN, &USRVAR), &EXTDAT, &EXTREQ, AD HOC	&EXTLAB	C:X
PeSIT-D	T1B2PS31	both	1, L=64	&P99	&EXTLAB	other
PeSIT-E	T1B2PS52	both	2, L=254	&P99	&EXTLAB	other

† or later

Identification with PeSIT

Three levels of negotiation are processed with the PeSIT protocol, including pre-connection, connection, and selection. Parameters can replace default values, and results are found in the Connect:Express journal record (D1B2PJNL, “Z45” fields).

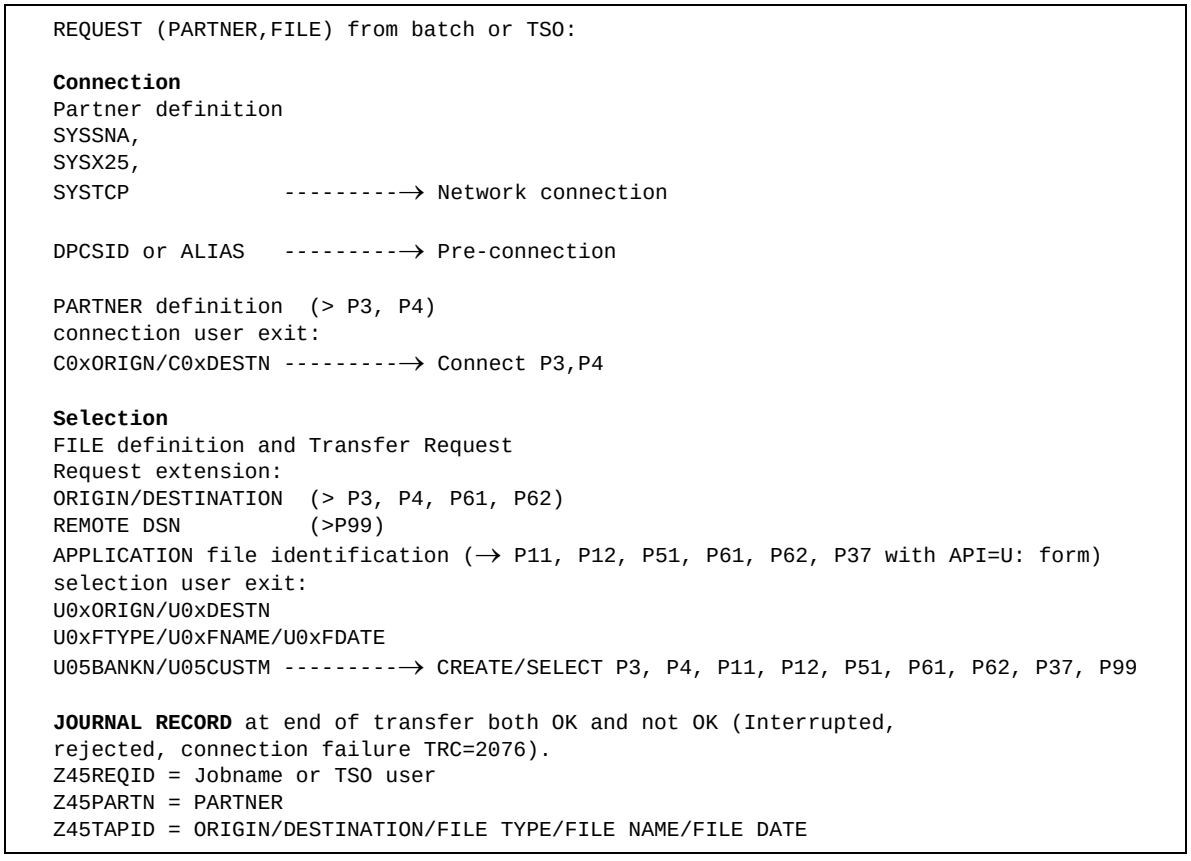
The pre-connection message follows the PeSIT standard but has a non-PeSIT format. It is used by computers to identify each other. The message consists of the following 3 eight-character EBCDIC fields.

Field	Description
Header	Constant = PeSIT
Partner	Symbolic name (Partner or ALIAS)
Password	Symbolic password (Partner or ALIAS)

Connect:Express Parameters and PeSIT Fields

Connect:Express parameters include the partner and the file definitions in the Directories and the SYS-SNA/TCP/X25 files, the DPCSID from the SYSIN file, the transfer request parameters, and the D1B2PCNX, D1B2RUEX structures exchanged with user exits. PeSIT fields include pre-connection fields, P3, P4, P11, P12, P37, P51, P61, P62 and P99.

The figure below shows the parameters that are sent when connecting to a remote partner and requesting a transfer. It also identifies where they are picked up and in which PeSIT message and field they are set. This figure shows how some parameters can be changed as you go through the process from transfer request to user exit. For example, during the selection process, all parameters come from the File definition and the transfer requests. Using the Origin/Destination fields you can change P3, P4, P61 and P62. Using a REMOTE DSN, you can change P99. After processing the user exit, the final set of parameters is moved into the PeSIT command.



The next two figures show the parameters that are received when a remote partner is connecting to you, and the PeSIT message and field where they are picked up. It also shows which internal definition is used for comparison before accepting the transfer.

```
Connection
                    ←---------- Network connection (X25, SNA, TCP/IP)
Waiting for first message

DIRECTORY             ←---------- PRE-CONNECTION (PeSIT,'id','psw')
SYSSNA
SYSTCP
SYSX25

'id' from the pre-connection message is looked for in the DIRECTORY
if found 'psw' is controlled

Network identification is controlled unless
'*' is given in the DIRECTORY,SYSX25, SYSSNA or SYSTCP
list of identification.

SYSX25 : used for X25 network connections
SYSTCP : used for TCP/IP network connections
SYSSNA : used for SNA network connections
contain:
- the list of possible network addresses for 'id', or '*'
- the PARTNER profile to be taken when 'id' is not
  found in the DIRECTORY for
X25 PAD connections
SNA TED3270 connections

            .---------- Connect P4 P3
Connection user exit:
C0xORIGN/C0xDESTN: (< P3 P4)
CNXPARTN:(< P3) can be replaced by a Partner profile
DPCSID or ALIAS:   (< P4) must match
DIRECTORY : (<P3) must be a symbolic name unless the user exit
translates the application P3 to a PARTNER profile set in CNXPARTN.
P3 must be the same as in Pre-connection message
unless the current PARTNER TYPE is neither
' TOM ' nor 'OTHER'.
```

```
Selection
Definition of the transfer REQUESTOR:
The default is the connection PARTNER
If the partner type is TOM or OTHER,
a new PARTNER can be received
as the transfer requestor.
If the partner type is A
the connection PARTNER is kept
as the REQUESTOR.
                    ←---------  CREATE/SELECT P3 P4 P11 P12 P37 P51 P61 P62 P99
Selection user exit: D1B2RUEX
UEXPART = CNXPARTN
UEXDDNM  (< P12)
U0xORIGN/U0xDESTN    (<P3, P4)
U0xFTYPE/U0xFNAME/U0xFDATE (<P11, P12, P51)
U05BANKN/U05CUSTM     (<P61, P62)
U05FLABL (< P37)
U05FUSDF (< P99)

DIRECTORIES: (<P12) must be a symbolic name unless the user exit
translates the application p12 to a FILE profile set in UEXDDNM.
P3 can be different from the connection P3.
If PARTNER TYPE is TOM or OTHER
P3 must be a PARTNER identification found in the
directory unless the transferring PARTNER of the
FILE is $$API$$.

JOURNAL RECORD written at end of transfer OK or not OK
after it has been accepted by Connect:Express :
Z45REQID = U0xORIGN (origin of transfer) for ' TOM ' or 'OTHER'
Z45REQID = UEXPARTN (connection) for type = 'APPLICATION'
Z45PARTN = UEXPARTN (connection)
Z45TAPID = ORIGIN/DESTINATION/FILE TYPE/FILE NAME/FILE DATE
Z45RTDSN = from U05FUSDF (P99)
```

PeSIT Commands and Connect:Express Events

The table below lists 4 events that you can have when transferring data with Connect:Express. For each event, there are some special considerations which are identified in the tables of this section.

Event	Description
CALLER/sender	Initiates a WRITE access on the remote computer.
CALLER/receiver	Initiates a READ access on the remote computer.
CALLED/receiver	Accepts a WRITE access from the remote computer.
CALLED/sender	Accepts a READ access from the remote computer.

The tables in this section describe operations on the network, Connect:Express, and the user interface. You can use the following 3 levels to analyze events. The table below describes each level.

Level	Description
Protocol commands	NETWORK TRACE used for analysis
Connect:Express actions	OS/390 and Connect:Express logs used for analysis
USER actions	USER messages and internal traces used for analysis

PeSIT Caller/Sender Events

PeSIT caller/sender events occur when you initiate a write access on a remote computer. The following table shows the relationship between what happens during a PeSIT session, what is shown in the Connect:Express environment, and how you can access information. It also includes a description of the process and reports that you can view.

PeSIT Caller/Sender Events Table

PROTOCOL Event	Connect:Express Event	USER Interface	Comment	Report
	REQUEST ACCEPTED UNLOAD	L0B2Z20 P1B2PREQ P1B2PRQ2 P1B2PRQ3 TSO/ISPF	You request Connect:Express to send a file to a partner.	LOG RCT
	Network connection		Response ok or nok.	
	Network NOK USER EXIT CONNECTION REJECTED PRC=310	D1B2PCNX CNX,F	Call retry is activated or partner is disabled.	SYSLOG RCT (waiting) JOURNAL
	Network OK		Session established	
Pre- connect			Identification of computers. Response is ACK0 or NAK0.	
NAK0	USER EXIT CONNECTION REJECTED TRC=1000	D1B2PCNX CNX,F	Logon rejected: Call retry is activated or partner is disabled.	LOG RCT (waiting) JOURNAL
ACK0			Logon Accepted.	
	USER EXIT	D1B2PCNX CNX,C	Network identification. Protocol connection application parameters are passed to Connect:Express.	RCT (init.)
Connect X'4020'			Response is A-Connect or R-Connect. Synchronization negotiation	

PeSIT Caller/Sender Events Table (continued)

PROTOCOL Event	Connect:Express Event	USER Interface	Comment	Report
R-Connect X'4022'	USER EXIT CONNECTION REJECTED PRC=yyxx	D1B2PCNX CNX,F	Session is rejected. Call retry is activated or partner is disabled.	LOG RCT (waiting) JOURNAL
A-Connect X'4021'	COMMUNICATION OPENED		The session context is passed to APM. Negotiated to the lower synchronization value	LOG RCT (select prot.)
	USER EXIT FILE and checkpoint ALLOCATION	D1B2RUEX SEL,A D1B2RUEX IEX,I	You can initiate protocol fields and allocate file (UEXALREC, U03,5.FRFMT, U03,5.SPTYP, U03,5.FKBYT expected).	
CREATE X'C011'			Response is A-CREATE or R-CREATE Message length negotiation.	
R-CREATE X'C030'	USER EXIT TRANSFER REJECTED DEALLOCATION DELETE checkpoint	D1B2RUEX TEX,F D1B2RUEX SEL,F	For information PRC=yyxx and deallocation. Selection retry is activated or request is disabled.	LOG RCT (error sel.) JOURNAL
A-CREATE X'C030'	BEGINNING OF TRANSFER COMMAND TRANSFER ACCEPTED		Negotiated to the lower message length value.	LOG
OPEN X'C014'			Compression negotiation Response is A-OPEN or R-OPEN	
R-OPEN X'C033'	USER EXIT ERROR SELECTION DEALLOCATION DELETE checkpoint	D1B2RUEX TEX,F D1B2RUEX SEL,F	For information PRC=yyxx and deallocation	LOG RCT (error sel.) JOURNAL
A-OPEN X'C033'			Negotiated to the lower compression value	
WRITE X'C002'			Start transfer Response is a-WRITE or R-WRITE	
R-WRITE X'C036'	USER EXIT ERROR SELECTION DEALLOCATION DELETE checkpoint	D1B2RUEX TEX,F D1B2RUEX SEL,F	For information PRC=yyxx and deallocation	LOG RCT (error sel.) JOURNAL
A-WRITE X'C036'	Point to RESTART.		Send restart point	
	USER EXIT TRANSFER STARTED	D1B2RPEX PEX,B (open)	Initiate begin of transfer	LOG RCT (started)

PeSIT Caller/Sender Events Table (continued)

PROTOCOL Event	Connect:Express Event	USER Interface	Comment	Report
	READ record USER EXIT	D1B2RPEX PEX,M + record	Read, pass to user, compress fill buffer and send until synch reached	
SYNC X'C003'	Write checkpoint		Many synch points are stored as the value of the window (P7) negotiated at connection.	
A-SYNC X'C038'			Any of the non acknowledged synch points sent can be acknowledged and all previous are then acknowledged.	RCT (started)
DTF-END X'C004'	USER EXIT CLOSE file	D1B2RPEX PEX,C (close)	End of file detected.	
TRANS-END X'C008'			Record and byte counts are sent Response is A-TRANSEND or R-TRANSEND	
R-TRANSEND X'C037'			Transfer is interrupted CLOSE and DESELECT commands sent next.	
A-TRANSEND X'C037'			All previous synch points are acknowledged	
CLOSE X'C015'			Response is A-CLOSE or R-CLOSE	
R-CLOSE X'C034'			Transfer is interrupted DESELECT sent next	
A-CLOSE X'C034'				
DESELECT X'C013'			Transfer is ended, acknowledged by receiver. Response is A-DESELECT or R-DESELECT	
R-DESELECT X'C032'	USER EXIT TRANSFER INTERRUPTED Anything more to do ? If yes, next transfer processed (CREATE or SELECT). If not, control returned to TOM address space for RELEASE.	D1B2RUEX TEX,F D1B2RUEX SEL,f	Transfer is interrupted	LOG JOURNAL RCT (interr.) SMF

PeSIT Caller/Sender Events Table (continued)

PROTOCOL Event	Connect:Express Event	USER Interface	Comment	Report
A-DESELECT X'C032'	USER EXIT END OF TRANSFER COMMAND TRANSFER ENDED DEALLOCATION Delete checkpoint PURGE request Anything more to do? If not, control returned to Connect:Express address space	D1B2RUEX TEX,E D1B2RUEX SEL,E	Transfer is ended, acknowledged by receiver. Transfer context is cleared. If any error occurs at TRANS-END or CLOSE, the transfer is interrupted.	LOG JOURNAL SMF RCT (purged) RCT (interr.)
RELEASE X'4023'				
RELCONF X'4024'	USER EXIT SESSION CLOSED	D1B2PCNX CNX,E		

PeSIT Caller/Receiver Events

PeSIT caller/receiver events occur when you initiate a read access on a remote computer. The following table shows the relationship between what happens during a PeSIT session, what is shown in the Connect:Express environment, and how you can access information. It also includes a description of the process and reports that you can view.

PeSIT Caller/Receiver Events Table

Protocol Event	Connect:Express Event	User Interface	Comment	Report
	REQUEST ACCEPTED	L0B2Z20 P1B2PREQ P1B2PRQ2 P1B2PRQ3 TSO/ISPF	You request Connect:Express to receive a file from a partner.	LOG RCT
	Network connection		Response ok or nok.	
	Network NOK USER EXIT CONNECTION REJECTED PRC=310	D1B2PCNX CNX,F	Call retry is activated or partner is disabled.	LOG RCT (waiting) JOURNAL
	Network OK		Session established	
Pre-connect			Identification of computers. Response is ACK0 or NAK0.	
NAK0	USER EXIT CONNECTION REJECTED TRC=1000	D1B2PCNX CNX,F	Logon rejected: Call retry is activated or request is disabled.	LOG RCT (waiting) JOURNAL

PeSIT Caller/Receiver Events Table (continued)

Protocol Event	Connect:Express Event	User Interface	Comment	Report
ACK0			Logon accepted.	
	USER EXIT	D1B2PCNX CNX,C	Network identification. Protocol connection application parameters are passed to Connect:Express.	RCT(init.)
Connect X'4020'			Response is A-Connect or R-Connect. Synchronization negotiation	
R-Connect X'4022'	USER EXIT CONNECTION REJECTED PRC=yyxx	D1B2PCNX CNX,F	Session is rejected Call retry is activated or request is disabled.	LOG: RCT (waiting) JOURNAL
A-Connect X'4021'	COMMUNICATION OPENED		The session context is passed to APM. Negotiated to the lower synchronization value.	LOG RCT (select prot.)
	USER EXIT	D1B2RUEX SEL,A or D1B2RUEX IEX,A	You can initiate protocol fields	
SELECT X'C012'			Response is A-SELECT or R-SELECT Message length negotiation	
R-SELECT X'C031'	USER EXIT TRANSFER REJECTED	D1B2RUEX TEX,F D1B2RUEX SEL,F	For information PRC=yyxx. Call retry is activated or request is disabled.	LOG RCT (error sel.) JOURNAL
A-SELECT X'C031'	BEGINNING OF TRANSFER COMMAND TRANSFER ACCEPTED USER EXIT FILE and checkpoint ALLOCATION	D1B2RUEX IEX,I	You can allocate file. Negotiated to the lower message length value.	LOG
OPEN X'C014'			Compression negotiation Response is A-OPEN or R-OPEN	
R-OPEN X'C033'	USER EXIT ERROR SELECTION DEALLOCATION DELETE checkpoint	D1B2RUEX TEX,F D1B2RUEX SEL,F	For information PRC=yyxx and deallocation. Send restart point.	LOG RCT (error sel.) JOURNAL
A-OPEN X'C033'	Point to RESTART		Negotiated to the lower compression value.	
READ X'C001'			Send restart point Response is A-READ or R-READ	

PeSIT Caller/Receiver Events Table (continued)

Protocol Event	Connect:Express Event	User Interface	Comment	Report
R-READ X'C035'	USER EXIT ERROR SELECTION DEALLOCATION DELETE checkpoint	D1B2RUEX TEX,F D1B2RUEX SEL,F	For information PRC=yyxx and deallocation	SYSLOG RCT (error sel.) JOURNAL
A-READ X'C035'				
First data	USER EXIT TRANSFER STARTED	D1B2RPEX PEX,B (open)	Initiate begin of transfer	LOG RCT (started)
Data	Process I/O buffer USER EXIT	D1B2RPEX PEX,M + record	Uncompress, pass to user, fill buffer, write until synch received	
SYNC X'C003'	Force physical I/O Write checkpoint		Acknowledge synch	
A-SYNC X'C038'			Any of the non acknowledged synch points received are thus acknowledged.	RCT (started)
DTF-END X'C004'	Force physical I/O Write checkpoint USER EXIT CLOSE file	D1B2RPEX PEX,C (close)		
TRANS-END X'C008'			Record and byte counts are sent All previous synch points are acknowledged Response is A-TRANSEND or R-TRANSEND	
R-TRANSEN D X'C037'			Transfer is interrupted CLOSE and DESELECT commands sent next.	
A-TRANSEN D X'C037'			All previous synch points are acknowledged	
CLOSE X'C015'			Response is A-CLOSE or R-CLOSE	
R-CLOSE X'C034'			Transfer is interrupted DESELECT sent next	
A-CLOSE X'C034'	USER EXIT END OF TRANSFER COMMAND TRANSFER ENDED DEALLOCATION Delete checkpoint	D1B2RUEX TEX,E D1B2RUEX SEL,E	Transfer is ended	
DESELECT X'C013'			Response is A-DESELECT or R-DESELECT	

PeSIT Caller/Receiver Events Table (continued)

Protocol Event	Connect:Express Event	User Interface	Comment	Report
R-DESELECT X'C032'	TRANSFER ENDED PURGE request Anything more to do ? If yes, next transfer processed (CREATE or SELECT). If not, control returned to TOM address space for RELEASE.			LOG JOURNAL RCT (interr.) SMF
A-DESELECT X'C032'	TRANSFER ENDED PURGE request Anything more to do? If not, control returned to Connect:Express address space		Transfer context is cleared. If any error occurs at TRANS-END or CLOSE, the transfer is interrupted.	SYSLOG JOURNAL SMF RCT purged RCT (interr.)
RELEASE X'4023'				
RELCONF X'4024'	USER EXIT SESSION CLOSED	D1B2PCNX CNX,E		

PeSIT Called/Receiver Events

PeSIT called/receiver events occur when you accept a write access from a remote computer. The following table shows the relationship between what happens during a PeSIT session, what is shown in the Connect:Express environment, and how you can access information. It also includes a description of the process and reports that you can view.

PeSIT Called/Receiver Events Table

PROTOCOL Event	Connect:Express Event	USER Interface	Comment	Report
	Incoming call accepted		Network requests Connect:Express to establish session.	
Preconnection	Identification of symbolic Partner or Partner profile. Check resources. Allocate APM/EFF		Remote computer requests Connect:Express to establish PeSIT session.	RCT (init.)
Network cut	USER EXIT CONNECTION REJECTED	D1B2PCNX CNX,F	Major resource (Connect:Express, PCT, FCT) inactive	SYSLOG
NAK0	USER EXIT CONNECTION REJECTED	D1B2PCNX CNX,F	PeSIT LOGON message rejected TRC=100x x= 1: Header, 2: partner or 3: password.	LOG
ACK0			PeSIT LOGON accepted.	RCT (init.)

PeSIT Called/Receiver Events Table (continued)

PROTOCOL Event	Connect:Express Event	USER Interface	Comment	Report
Connect x'4020'	USER EXIT	D1B2PCNX CNX,C	Partner identification. Protocol connection application parameters are passed to USER. Control by you (user). Response is R-Connect or A-Connect.	RCT (init.)
R-Connect X'4022'	USER EXIT CONNECTION REJECTED if previously accepted.	D1B2PCNX CNX,F		LOG
A-Connect X'4021'	COMMUNICATION OPENED		Accepted by USER, Connect:Express Control passed to APM. Negotiated to the lower synchronization value.	LOG RCT (select prot.)
CREATE X'C011'	USER EXIT	D1B2RUEX SEL,A	You are passed protocol fields that identify the transfer and return to Connect:Express the files symbolic name to be processed.	
	Check resources allocate and catalog file allocate transfer checkpoint		Connect:Express checks resources and returns to APM exit names, data set name...	
	USER EXIT BEGINNING OF TRANSFER COMMAND TRANSFER ACCEPTED or REJECTED	D1B2RUEX IEX,I	Response is A-CREATE or R-CREATE user can allocate file	LOG
R-CREATE X'C030'	USER EXIT TRANSFER REJECTED DELETE data set and check point PURGE request	D1B2RUEX TEX,F D1B2RUEX SEL,F	For information TRC=xxxx and deallocation Transfer context is cleared	LOG
A-CREATE X'C030'			Negotiated to the lower message length value.	RCT (selec prot.)
OPEN X'C014'			Compression negotiation Response is A-OPEN or R-OPEN	
R-OPEN X'C033'	USER EXIT ERROR SELECTION DELETE data set DELETE checkpoint PURGE request	D1B2RUEX TEX,F D1B2RUEX SEL,F	For information TRC=xxxx and deallocation Transfer context is cleared	LOG
A-OPEN X'C033'			Negotiated to the lower compression value.	

PeSIT Called/Receiver Events Table (continued)

PROTOCOL Event	Connect:Express Event	USER Interface	Comment	Report
WRITE X'C002'		Point to RESTART Response is A-WRITE or R-WRITE		
R-WRITE X'C036'	USER EXIT ERROR SELECTION DELETE data set DELETE checkpoint PURGE request	D1B2RUEX TEX,F D1B2RUEX SEL,F	For information TRC=xxxx and deallocation Transfer context is cleared	LOG
A-WRITE X'C036'			Send restart point	
First data	USER EXIT TRANSFER STARTED	D1B2RPEX PEX,B (open)	Initiate begin of transfer	LOG RCT (started)
Data	Process I/O buffer USER EXIT	D1B2RPEX PEX,M + record	Uncompress, pass to user, fill buffer, write until synch received	
SYNC X'C003'	Force physical I/O Write checkpoint		Acknowledge synch	
A-SYNC X'C038'			Any of the non acknowledged synch points received are thus acknowledged.	RCT (started)
DTF-END X'C004'	Force physical I/O Write checkpoint USER EXIT CLOSE file	D1B2RPEX PEX,C (close)		
TRANS-END X'C008'			Record and byte counts are received Response is A-TRANSEND or R-TRANSEND	
R-TRANSEND X'C037'			Transfer is interrupted CLOSE and DESELECT commands received next.	
A-TRANSEND X'C037'			All previous synch points are acknowledged	
CLOSE X'C015'			Response is A-CLOSE or R-CLOSE	
R-CLOSE X'C034'			Transfer is interrupted DESELECT received next	
A-CLOSE X'C034'				

PeSIT Called/Receiver Events Table (continued)

PROTOCOL Event	Connect:Express Event	USER Interface	Comment	Report
DESELECT X'C013'	USER EXIT END OF TRANSFER COMMAND TRANSFER ENDED or INTERRUPTED DEALLOCATION Delete checkpoint	D1B2RUEX TEX,E/F D1B2RUEX SEL,E/F	Transfer is ended or interrupted after A-CREATE. response is R-DESELECT or A-DESELECT depending on you (user)	
R-DESELECT X'C032'	TRANSFER INTERRUPTED		Wait for next transfer or end session.	SYSLOG JOURNAL RCT (interr.) SMF
A-DESELECT X'C032'	TRANSFER ENDED PURGE request Delete Checkpoint		Wait for next transfer or end session. Transfer context is cleared. If any error occurs at TRANS-END or CLOSE, the transfer is interrupted.	LOG JOURNAL SMF RCT purged RCT (interr.)
RELEASE X'4023'			Control returned to Connect:Express	
RELCONF X'4024'	USER EXIT SESSION CLOSED	D1B2PCNX CNX,E		

PeSIT Called/Sender Events

PeSIT called/sender events occur when you accept a read access from a remote computer. The following table shows the relationship between what happens during a PeSIT session, what is shown in the Connect:Express environment, and how you can access information. It also includes a description of the process and reports that you can view.

PeSIT Called/Sender Events Table

PROTOCOL Event	Connect:Express Event	USER Interface	Comment	Report
	Incoming call accepted		Network requests Connect:Express to establish session.	
Preconnection	Identification of symbolic Partner or Partner profile. Check resources. Allocate APM/EFF		Remote computer requests Connect:Express to establish PeSIT session.	RCT (init.)
Network cut	USER EXIT CONNECTION REJECTED	D1B2PCNX CNX,F	Major resource (Connect:Express, PCT, FCT) inactive	LOG

PeSIT Called/Sender Events Table (continued)

PROTOCOL Event	Connect:Express Event	USER Interface	Comment	Report
NAK0	USER EXIT CONNECTION REJECTED	D1B2PCNX CNX,F	PeSIT LOGON message rejected TRC=100x x= 1: Header, 2: partner or 3: password.	LOG
ACK0			PeSIT LOGON accepted.	RCT (init.)
Connect x'4020'	USER EXIT	D1B2PCNX CNX,C	Partner identification. Protocol connection application parameters are passed to USER. Control by you (user). Response is R-Connect or A-Connect.	RCT (init.)
R-Connect X'4022'	USER EXIT CONNECTION REJECTED if previously accepted.	D1B2PCNX CNX,F		LOG
A-Connect X'4021'	COMMUNICATION OPENED		Accepted by USER, Connect:Express. Control passed to APM. Negotiated to the lower synchronization value.	LOG RCT (select prot.)
SELECT X'C012'	USER EXIT	D1B2RUEX SEL,A	You are passed protocol fields that identify the transfer and return to Connect:Express the files symbolic name to be processed.	
	Check resources allocate file allocate transfer checkpoint		Connect:Express checks resources and returns to APM exit names, data set name...	
	USER EXIT BEGINNING OF TRANSFER COMMAND TRANSFER ACCEPTED or REJECTED	D1B2RUEX IEX,I	Response is A-SELECT or R-SELECT You (user) can initiate protocol fields and allocate file (UEXALREC, U03,5.FRFMT, U03,5.SPTYP, U03,5.FKBYT expected).	LOG
R-SELECT X'C031'	USER EXIT TRANSFER REJECTED DELETE checkpoint PURGE request	D1B2RUEX TEX,F D1B2RUEX SEL,F	For information TRC=xxxx and deallocation Transfer context is cleared	LOG
A-SELECT X'C031'			Negotiated to the lower message length value.	RCT (select prot.)
OPEN X'C014'			Compression negotiation Response is A-OPEN or R-OPEN	

PeSIT Called/Sender Events Table (continued)

PROTOCOL Event	Connect:Express Event	USER Interface	Comment	Report
R-OPEN X'C033'	USER EXIT ERROR SELECTION DELETE checkpoint PURGE request	D1B2RUEX TEX,F D1B2RUEX SEL,F	For information TRC=xxxx and deallocation Transfer content is cleared.	LOG JOURNAL
A-OPEN X'C033'			Negotiated to the lower compression value.	
READ X'C001'		Point to RESTART Response is A-READ or R-READ		
R-READ X'C035'	USER EXIT ERROR SELECTION DELETE checkpoint PURGE request	D1B2RUEX TEX,F D1B2RUEX SEL,F	For information TRC=xxxx and deallocation Transfer context is cleared.	LOG JOURNAL
A-READ X'C035'				
	USER EXIT TRANSFER STARTED	D1B2RPEX PEX,B (open)	Initiate begin of transfer	LOG RCT (started)
	READ record USER EXIT	D1B2RPEX PEX,M + record	Read, pass to user, compress fill buffer and send until synch reached	
SYNC X'C003'	Write checkpoint		Many synch points are stored as the value of the window (P7) negotiated at connection.	
A-SYNC X'C038'			Any of the non acknowledged synch points sent can be acknowledged and all previous are then acknowledged.	RCT (started)
DTF-END X'C004'	USER EXIT CLOSE file	D1B2RPEX PEX,C (close)	End of file detected.	
TRANS-END X'C008'			Record and byte counts are compared Response is A-TRANSEND or R-TRANSEND	
R-TRANSEND X'C037'			Transfer is interrupted CLOSE and DESELECT commands received next.	
A-TRANSEND X'C037'			All previous synch points are acknowledged	
CLOSE X'C015'			Response is A-CLOSE or R-CLOSE	
R-CLOSE X'C034'			Transfer is interrupted DESELECT received next	

PeSIT Called/Sender Events Table (continued)

PROTOCOL Event	Connect:Express Event	USER Interface	Comment	Report
A-CLOSE X'C034'				
DESELECT X'C013'			Transfer is ended, acknowledged by receiver. Response is A-DESELECT or R-DESELECT	
R-DESELECT X'C032'	USER EXIT TRANSFER INTERRUPTED Wait for next demand.	D1B2RUEX TEX,F D1B2RUEX SEL,f	Transfer is interrupted	LOG JOURNAL RCT (interr.) SMF
A-DESELECT X'C032'	USER EXIT END OF TRANSFER COMMAND TRANSFER ENDED DEALLOCATION Delete checkpoint PURGE request Wait for next demand.	D1B2RUEX TEX,E D1B2RUEX SEL,E	Transfer is ended, acknowledged by receiver. Transfer context is cleared. If any error occurs at TRANS-END or CLOSE, the transfer is interrupted.	SYSLOG JOURNAL SMF RCT purge RCT (interr.)
RELEASE X'4023'				
RELCONF X'4024'	USER EXIT SESSION CLOSED	D1B2PCNX CNX,E		

PeSIT Parameters and Connect:Express Definitions

Connect:Express definitions are parameters stored in Connect:Express files or passed through user interface fields. These definitions are put in protocol fields when sending PeSIT commands to the partner. When receiving PeSIT commands, protocol fields are compared with Connect:Express definitions.

The following table shows the relationship between the protocol phase, the protocol parameters and their length, and the Connect:Express definitions or keywords and their length. It also includes a description of the process, and reports that you can view.

PeSIT Parameters and Connect:Express Definitions Table

Protocol phase	Protocol (param),lg	Connect:Express param,lg	Description	Report
Pre-connection	Field 1,8		« PeSIT »	

PeSIT Parameters and Connect:Express Definitions Table (continued)

Protocol phase	Protocol (param),lg	Connect:Express param,lg	Description	Report
Pre-connection	Field 2,8	Incoming PARTNER,8 &PARTNID,8 Outgoing DPCSID,8 ALIAS,8 &PARTNID ,8	Symbolic identification From SYSIN or directory or request alias	SYSLOG RCT JOURNAL
Pre-connection	Field 3,8	Incoming PASSWD,8 Outgoing DPCPSW,8 ALIAS,8	Symbolic password From SYSIN or directory or request alias	
Connection	p(1),1	SPLOGIC,1	Incoming and outgoing. In the T1B2PS3x and 5x tables: Y or N CRC option for X.25 asynchronous connections. On X25 PAD environment option is strongly advised.	
Any	p(2),3	UEXPRC,3 CNXPRC,3 PEXPRC,3	Incoming and outgoing. PeSIT return code issued by Connect:Express, the user or the partner.	LOG RCT JOURNAL
Connection	p(3),24	Incoming Symbolic PARTNER,8 C03ORGN C05ORGN &ORG Outgoing DPCSID,8 ALIAS,8 C03ORGN C05ORGN &ORG	Session initiator name.	Incoming: LOG RCT JOURNAL SMF
Selection	p(3),24	Incoming Symbolic PARTNER,8 U03ORGN U05ORGN &ORG Outgoing DPCSID,8 ALIAS,8 ORIGIN,8 U03ORGN U05ORGN &ORG	Transfer Partner name. ALIAS from directory or request extension. ORIGIN from request extension. Outgoing API from request extension	LOG RCT JOURNAL SMF

PeSIT Parameters and Connect:Express Definitions Table (continued)

Protocol phase	Protocol (param),lg	Connect:Express param,lg	Description	Report
Connection	p(4),24	Incoming DPCSID,8 ALIAS,8 C03DESTN C05DESTN &DST Outgoing Symbolic PARTNER,8 C03DESTN C05DESTN &DST	Symbolic Session addressee name	Outgoing: LOG RCT JOURNAL SMF
Selection	p(4),24	Incoming DPCSID,8 ALIAS,8 U03DESTN U05DESTN Outgoing Symbolic PARTNER,8 DESTIN.,8 U03DESTN U05DESTN	Transfer initiator name ALIAS from directory or request extension. DESTIN. from request extension. Outgoing API from request extension	LOG RCT JOURNAL SMF
Connection	p(5),16	Incoming symbolic PASSWORD,8 Outgoing DPCPSW,8 ALIAS,8	Password ALIAS from directory or request extension. NOTE: It is possible to bypass password control. If the password field is set to $NONE$, no password verification is performed.	
Connection	p(6),1	SPVERSN,1	In the T1B2PS3x and 5x tables, 1 or 2 Incoming and outgoing PeSIT level: 1 – PeSIT D 2 – PeSIT D' if in T1B2PS3x – PeSIT E in T1B2PS5x	
Connection	p(7),1+2	SPSNCOCC,2 + SPWINDOW,1	Synchronization parameters	
	p(8)	N/A		
	p(9)	N/A	Parameters group p(3)p(4)	
	p(10)	N/A		
Selection	p(11),8	API U03FTYPE,5 U05FTYPE,8	File type Outgoing API from request extension.	RCT JOURNAL
Selection	p(12),76	Symbolic file,8 API U03FNAME U05FNAME &FILENAM	File name Outgoing API from request extension.	LOG RCT JOURNAL

PeSIT Parameters and Connect:Express Definitions Table (continued)

Protocol phase	Protocol (param),lg	Connect:Express param,lg	Description	Report
Selection	p(13),3	Request Number &REQNUMB	Transfer identification • Local REQUEST number if transmission or normal reception. • External identification if reception (REQUEST number if Connect:Express Partner).	LOG RCT JOURNAL
Selection	p(14),1	N/A	Files attributes flag for reception. see p(30), p(40), p(50) Connect:Express requests all attributes (logical, physical and historical).	
Selection	p(15),1	N/A	Restart demand flag. Set when restart activated for a REQUEST through TSO/ISPF or batch utility.	
Selection	p(16),1	PPDATATP	Data type (EBCDIC, ASCII, BINARY) This parameter is picked up from T1B2PPxx table when transmitting the file. It is not processed when receiving a file.	JOURNAL
Selection	p(17),1	TSO/ISPF Batch request File Directory	Transfer priority	RCT JOURNAL
Selection	p(18),3	Internal	Restart point see P(13)	LOG RCT JOURNAL
Selection	p(19),1	Internal	End of transfer code • transfer ended • selection error • transfer will be restarted • canceled (stopped and purged)	LOG RCT JOURNAL
Transfer	p(20)	Internal	Synchronization point	
Selection	p(21),2	PPCOMPR,2	Compression level in the T1B2PPxx table T1B2PP01 = X'00', No compression T1B2PP02 = X'11', Horizontal compression T1B2PP03 = X'12', Vertical compression T1B2PP04 = X'13', Mixed compression The lowest profile is negotiated between the partners.	JOURNAL
Connection	p(22),1	SPDIREC,1	In the T1B2PS3x and 5x tables. * – Both direction U – Unique direction a transfer is chained under a session opened for the same direction. This must be used for PeSIT D Partner type = other, table T1B2PS31 is provided.	JOURNAL
Connection	p(23),1	SPRESYNC,1	In the T1B2PS3x and 5x tables. Resynchronization option for asynchronous links.	JOURNAL
	p(24)	N/A		

PeSIT Parameters and Connect:Express Definitions Table (continued)

Protocol phase	Protocol (param),lg	Connect:Express param,lg	Description	Report
Selection	p(25),2	SPBUFLNG,2 STDMSG= MAXEXT=	In the T1B2PS3x and 5x tables. Buffer length for data FPDU's. The ANM buffer length parameters set in the SYSIN file fit within the protocol message length.	
Connection	p(26),2		In the T1B2PS5x tables. Timer parameter PeSIT E only	
Transfer	p(27),8	internal UEXTLBYT	Number of bytes transferred	LOG JOURNAL
Transfer	p(28),4	internal UEXTLRRW	Number of records transferred	LOG JOURNAL
Connection	selection	p(29),254	U05DIAGN (PeSIT-E)	diagnostic message
Selection	p(30)		Parameter group p(31...39)	
Selection	p(31),1	file directory U03FRFMT U05FRFMT	Record format see *PeSIT Allocation Attributes and IBM OS/390 DCB*, page 5-24 for correspondence between protocol values and IBM OS/390 DCB values.	
Selection	p(32),2	file directory UEXALREC	Record length See *PeSIT Allocation Attributes and IBM OS/390 DCB,* page 5-24 for correspondence between protocol values and IBM OS/390 DCB values.	
Selection	p(33),1	N/A	File organization. not used (files transferred are always declared sequential).	
Selection	p(34)	N/A	Not used	
	p(35)	N/A		
	p(36)	N/A		
Selection	p(37),80	U03FLABL U05FLABL &EXTLAB API=U:	Data set name (IBM OS/390 syntax or not) from user or application data. Outgoing: If request extension API parameter is used API=U:user-data then p(37)='user-data' Default is the request data set name.	
Selection	p(38),2	N/A	Not used	
Selection	p(39),2	N/A	Not used	
Selection	p(40)	N/A	Parameter group p(41,42)	
Selection	p(41),1	U03FSPTY U05FSPTY	File space definition type	
Selection	p(42),4	U03FSPAC U05FSPAC	File space (kilobytes or records) Primary space allocated Outgoing: Computed from the DSCB.	LOG JOURNAL
Selection	p(50)		Parameter group p(51,52)	

PeSIT Parameters and Connect:Express Definitions Table (continued)

Protocol phase	Protocol (param),lg	Connect:Express param,lg	Description	Report
Selection	p(51),12	U03FCDAT U05FCDAT API	File creation date Outgoing: API from request extension. The default is the system date. Incoming: API	RCT JOURNAL
Selection	p(52)	U03FEDAT U05FEDAT	File extraction date	
Selection	p(61)	U05custm API	Customer identification (PeSIT-E) Outgoing: API from request extension. The default is P(3), origin. Incoming: API	RCT JOURNAL
Selection	p(62)	U05BANKN API	Company identification (PeSIT-E) Outgoing: API from request extension. The default is P(4), destination Incoming: API	RCT JOURNAL
Selection	p(63)	U05PASWD	File access password (PeSIT-E)	
Selection	p(64)	U05SRVDT	File access server date (PeSIT-E)	
Selection	p(71)	AUTH UEXSOAUT	Authentication context (PeSIT-E) T1SECPxx	
Selection	p(72)	Internal	Authentication Alias (PeSIT-E)	
Selection	p(73)	SEAL UEXSOSEA	Sealing context (PeSIT-E) T1SECPxx	
Selection	p(74)	Internal	Sealing parameters (PeSIT-E)	UEXSEKES UEXSEIVS
Selection	p(75)	CIPH UEXSOCIP	Ciphering context (PeSIT-E) T1SECPxx	
Selection	p(76)	Internal	Ciphering parameters (PeSIT-E)	UEXSEKEC UEXSEIVC
Selection	p(77)	SIGN UEXSOSIG	Signature context (PeSIT-E) T1SECPxx	
Transfer	p(78)	UEXSESEA	Seal (PeSIT-E)	
Transfer	p(79)	UEXSESG1	Signature (PeSIT-E)	
Selection	p(80)	UEXSEACC UEXSEAC1	Transfer accreditation (PeSIT-E) First signature accreditation (PeSIT-E)	
Transfer	p(81)	UEXSESG1	Receiver signature (PeSIT-E)	
Transfer	p(82)	UEXSESG2	Second signature (PeSIT-E)	
Selection	p(83)	UEXSEAC2	Second signature accreditation (PeSIT-E)	

PeSIT Parameters and Connect:Express Definitions Table (continued)

Protocol phase	Protocol (param),lg	Connect:Express param,lg	Description	Report
Selection	p(99)	C05FUSDF U05FUSDF API=P: &EXTDAT &EXTDSN &EXTNUMB &MBR &x &USRVAR1 &USRVAR2 &Pi99	Protocol and field used with AD HOC, PDS, and VSAM files exchanging specific information. Outgoing If partner types is 'other,' the remote DSN parameter is moved to P(99). Incoming Received by &Pi99 if the partner type is other, and by the other keywords if the partner type is Connect:Express.	

Exchanging Pi99 with PeSIT Partners

Integration capabilities of Connect:Express have been enhanced by extending the use of the PeSIT Pi99 user field. It is possible to send and receive information linked to a transferred file using this field with PeSIT partners that have a type Other. If the Pi99 field is not used, working with Partners of type Other has not changed. The default is: Pi99 not used. Working with 'TOM' Partners has not changed.

Pi99 is processed with Create, Select, Ack-Create and Ack-Select PeSIT commands. Pi99 is processed in the data set name computation and in the beginning and end of transfer command parameter lists. Keywords are available for building this field before sending it. Keywords can also be sent to the remote.

The files directory field REMOTE DSN/PI99 or the transfer request field RDSN/PI99 are used to indicate the string to send in the Pi99. The PI99 fields are 44 characters long. You can also use the API field which can store up to 80 characters, with the 'API P:' syntax. The following screen shows an example:

```
RDSN/PI99 .. ---> &REQNUMB - &&REQNUMB -
The local request number is sent associated with keyword &REQNUMB which will be
replaced on the remote side: Pi99 sent = A0000025 - &REQNUMB.
```

PeSIT Allocation Attributes and IBM OS/390 DCB

The PeSIT protocol enables you to transfer sequential files between heterogeneous environments. Only fixed or variable records are declared by the p(31) PeSIT parameter, and no physical structure is carried through. For this reason, IBM OS/390 record formats are not correctly represented. The record length is a logical value, which means that the real record length of variable records is p(32) PeSIT parameter +4, due to the physical count added. The blocksize is not known by PeSIT, so it is a specific value only transmitted by a Connect:Express facility based on P(99). The size of the file P(42) PeSIT parameter is used for computing the primary space allocation when receiving files and is determined before sending the files. Depending on the file definition, the file received is allocated from the PeSIT parameters if the DCB is not given. If the DCB is given, the PeSIT parameters are ignored and you have the option of using automatic blocking and automatic space computing. See *Defining the DCB for Reception* and *Allocating Files in an SMS Environment* in the User Guide on page 2-21.

Receiving a Sequential File

When receiving a sequential file with no DCB entered in the directory, the following PeSIT fields are used for file allocation.

PeSIT	Allocation by receiving Connect:Express (no DCB given)
PeSIT P(31) = X‘00’	RECFM = FB
PeSIT P(32) = rrrr	LRECL = rrrr; BLOCKSIZE optimized (unless SMS is used)
PeSIT P(42) = ssss	Primary space = blocks equivalent to ssss bytes + 10%, no secondary space, unless SMS is used.
PeSIT P(31) = X‘80’	RECFM = VB
PeSIT P(32) = rrrr	LRECL = rrrr+4; BLOCKSIZE optimized (unless SMS is used)
PeSIT P(42) = ssss	Primary space = blocks equivalent to ssss bytes + 10%, secondary = 20%, unless SMS is used

When receiving a sequential file with the DCB entered in the directory, the following PeSIT fields are used for file allocation.

PeSIT	Allocation by receiving Connect:Express (DCB fixed in the directory)
PeSIT P(31) = X‘00’	RECFM = Any value if RECFMT=N in the SYSIN file (no control) or RECFM = F,FB,FBA only
PeSIT P(32) = rrrr	LRECL = rrrr only; BLOCKSIZE compatible.
PeSIT P(42) = ssss	P(42) omitted. If the space defined is not large enough, abend D37 can occur
PeSIT P(31) = X‘80’	RECFM = Any value if RECFMT=N in the SYSIN file (no control) or RECFM = V,VB,VBA,VBS,U only
PeSIT P(32) = rrrr	LRECL = any if RECFMT=N in the SYSIN file (no control) or rrrr+4 only; BLOCKSIZE compatible.
PeSIT P(42) = ssss	P(42) omitted. If the space defined is not large enough, abend D37 can occur.

Sending a Sequential File

From the existing file structure, Connect:Express sets the PeSIT parameters with the standard values shown below.

File sent by Connect:Express	PeSIT
RECFM = F,FB,FBA	P(31) = X‘00’
LRECL = rrrr	P(32) = rrrr
Space = t tracks	P(42) = t * size track*
RECFM = V,VB,VBA	P(31) = X‘80’
LRECL = rrrr	P(32) = rrrr - 4
Space = t tracks	P(42) = t * size track*

* If the file is compressed, P(42)=t* size track*8

File sent by Connect:Express	PeSIT
RECFM = VBS, U	P(31) = X‘80’
LRECL = block size	P(32) = block size
Space = t tracks	PeSIT P(42) = t * size track

* If the file is compressed, P(42)=t* size track*8

Sending and Receiving the IBM FILE DCB and DSN

When transferring between Connect:Express on IBM OS/390 systems, it is possible to exchange specific IBM allocation parameters. For example, when Partner type is Connect:Express (TYPE = T), and the RECFM, LRECL, and BLKSIZE from the local file DCB are automatically sent in the local DCB fields of p(99). The local data set name is sent in the local DSN field of p(99). The AD HOC request enables you to send the local and remote DCBs, DSN, and UNIT/VOLSER.

If the AD HOC facility is active on the receiver side and the AD HOC option is activated by the sender, the remote DCB, DSN, and Unit/Volser fields from p(99) are used locally.

If the AD HOC facility is not active on the receiver side, you can use the local DCB from p(99). If the DCB attributes are given in the File Directory on the receiver side, they are used to control the record format and record length, and for allocation. In this case, you can use the following keywords:

- Using the &X keywords, RECFM, blocksize and LRECL can be identical on both sides.
- Using the &EXTDSN keyword, the remote DSN can be integrated to the local.
- Using the &USRVAR keywords, the last index of the remote DSN is integrated to the local.

The ODETTE-FTP Protocol

The ODETTE-FTP transfer protocol is a profession-oriented transfer protocol provided by the European Car Constructors organization. Connect:Express facilities like PDS, VSAM, AD HOC, and file DCB are not available with this protocol because there are fewer user fields with this protocol.

Session tables for ODETTE-FTP are in the range of T1B2PS20 tables. The main parameters set in the tables are listed below:

- Compression
- Direction of transfer during session – transmit/receive/both directions
- Data buffer size – negotiated to the lower value
- Exchange buffer credit – negotiated to the lower value
- Asynchronous link data integrity control

Two levels of protocol are supported, ODETTE levels 1.2 and 1.3, X.25 and TCP/IP links are supported. The following table identifies the session tables used with this protocol.

Table	Description
T1B2PS20	This table is reserved and used for the following: (no CRC) • Both direction sessions • Compression • No CRC integrity control

Table	Description
T1B2PS21	This table is reserved and used for the CRC profile: • Both direction sessions • Compression • CRC integrity control (asynchronous sessions – special logic option)
T1B2PS22 to T1B2PS2E	These tables are user-specific.

One feature of the ODETTE-FTP protocol is that only the transmission command is defined, and sending the partner a Token initiates file reception.

When an ODETTE-FTP session is initiated, it cannot be terminated without giving your partner the opportunity to send one or more files. Therefore, an active session can only be closed just after the token is received or later using the ABORT command. You can make a transfer request with the file name $Connect to call a partner and receive a file without sending anything. The $Connect profile must be defined in the file directory.

Note: All ODETTE session tables that ship with Connect:Express are defined for ODETTE level 1.2, version F1. If you need to use ODETTE level 1.3, update one of the available tables and save it in your user library.

ODETTE-FTP Protocol Dynamic

The caller initiates the session and is the first speaker or sender of data. The called party is the listener or receiver of the data. When finished, the token is passed to the called party and they become the speaker and the caller becomes the listener. The last party to receive the token can end the session. The following screen illustrates this process.

```
CALLER            X25 call packet →     CALLED
                  ← READY
                  ODETTE session →

speaker send →                          listener
        send →

                  TOKEN →

listener                      ← send speaker
                            ← send

                  ← TOKEN

speaker           end session →         listener
```

A request sent to Connect:Express to transfer with an ODETTE-FTP Partner can be any of the following transfer types:

Transfer Type	Description
TRANSMIT NORMAL, transmit/receive session	Connect:Express calls the Partner, transmits the File and gives the Partner the token so he can acknowledge (end to end) the transfer and transmit something, or end the session. When several transfers are found waiting for transmission in the Connect:Express request table, the token is passed to the Partner when no more transfer is waiting for selection, unless the Partner asks for it.
TRANSMIT NORMAL, transmit session	Connect:Express calls the Partner, transmits the File and gives the Partner the token so he can acknowledge (end to end) the transfer, or end the session. When several transfers are found waiting for transmission in the Connect:Express request table, the token is passed to the Partner when no more transfers are waiting for selection, unless the Partner asks for it. Use this type for calling to send, without receiving.
TRANSMIT HELD, transmit or transmit/receive session	Use this type to let the Partner come and receive files either without transmitting anything or transmitting his own files in the same session. Transmission of the file is initiated when the Partner calls and gives Connect:Express the token.
$Connect	Use this type to call the Partner and receive files, if you have no file to send. Connect:Express calls the partner and gives him the token immediately so that the partner can send his files. Note: A $Connect symbolic file must be defined in the directory.

ODETTE-FTP Commands

ODETTE-FTP commands are identified by ASCII codes and their structures are fixed and composed of fixed length ASCII fields. The general structure of an ODETTE-FTP command is shown below.

```
                                        p5p6
|cmd|p1 |        p2        |   p3   | p4 | | |p7 |Cr ...
| X | 1 |INITIATOR ID CODE |PASSWORD|4000|B|Y|012|cr
. . . . . . . .
X  = SSID. . . . . . .
p1 = Version number = 1 . . . . .
. . . . . .
p2 = ODETTE identification . . . . .
p3 = Password . . . .
. . . .
p4 = Buffer . .
. . .
p5 = Both direction
. .
p6 = Compression
.
p7 = synchronization credit

Cr = Carriage Return
```

ODETTE-FTP Return Codes

The ODETTE-FTP protocol provides one list of return codes for the connection phase (SSID, ESID) and one list of return codes for the selection phase (SFID). This protocol causes it to appear in Connect:Express reports as a four-character code where the first two numbers mean:

00 = session phase

10 = selection phase.

Refer to Appendix B for a complete listing of ODETTE-FTP return codes.

Identification with ODETTE-FTP

The ODETTE FTP identification process involves some specific features for network, partner, and transfer identification.

Network Identification

One feature of the ODETTE-FTP connection process is the 'ODETTE LISTENER SPEAKING' response to a network inbound call. If ODETTE-FTP is used over X25, Connect:Express must recognize an ODETTE-FTP incoming call by the X25 USER DATA FIELD or by a special sub-address. These two parameters can be declared in the SYSIN fields, ODTUDF and ODTSAD. If ODETTE-FTP is used over TCP/IP the ODETTE-FTP listener must be started, with a special Port number. This parameter can be declared in the SYSIN field TCPPRO.

Note: The default OFTP port number is 3305.

Partner Identification

Another feature of the ODETTE-FTP protocol is the use of specific identification. For example, an Odette partner definition consists of a symbolic identification that can be associated with an Odette identification. The protocol number 2 in the symbolic definition indicates that the partner is an Odette partner. In this case, the following screen is displayed showing the ODETTE-FTP specific identification fields in the Directory.

```
TOM4200     PARTNER OF TOM4 TO UPDATE     (3/4)
 OPTION ===>

 SYMBOLIC NAME      : ODETTE               PROTOCOL  : ODETTE 2


   PESIT DATA :
   --------------------
 APPLICATION TYPE ...    :                 1, 2, 3, 4
 APPLICATION NUMBER .    :                 NUMERIC (00001:65535)




   ODETTE DATA :
   --------------------

 DETAIL (S) -----> ''    : UNSTRUCTURED ODETTE NAME



 OPTION   : UPDATE                         UPD   : PSR0008 00/01/11
 -ENTER-  : NEXT SCREEN                    -PF3- : PREVIOUS SCREEN
```

This screen is common to both PeSIT and Odette partners. You will be prompted for the Odette identification definition, which can be structured in a standard way or used as a single 25-character string. If you select the field, a structured screen is displayed to help you conform to the Odette standard. If no Odette identification is provided, the symbolic partner name is used as the first eight characters of the 25-character string.

```
TOM4200     PARTNER OF TOM4 TO UPDATE     (3/4)
 OPTION ===>

 SYMBOLIC NAME      : ODETTE               PROTOCOL  : ODETTE 2


 FORMATTING ODETTE IDENTIFICATION
 --------------------------------


 INTERNATIONAL CODE . ===> ''''            NUMERIC (0001:9999/NONE)
 ORGANIZATION CODE .. ===> ''''''''''''''  ALPHANUM
 SUB ADDRESS  ....... ===> ''''''          NUMERIC (000000:999999)









 OPTION   : UPDATE                         MAJ   : USR0008 01/04/06
 -ENTER-  : NEXT SCREEN                    -PF3- : PREVIOUS SCREEN
```

Both the remote partner or local partner must be identified in the Connect:Express directory. The local partner is determined from either the DPCSID field in the SYSIN file (default), or in the Alias field from the remote partner definition. This local symbolic name must be defined in the directory to associate the corresponding local Odette identification. This partner identification is used for connection (SSID command).

The following screen shows the ODETTE-FTP local or remote Partner identification.

```
TOM4200     PARTNER OF TOM4 TO  VIEW       (2/4)
 OPTION ===>                        -ENTER- : GO ON, -PF3- : CANCEL   X : EXIT
 TYPE: TOM,ODETTE
 MOD: USR0008  01/04/06 10:35:10        1
 SYMBOLIC NAME         : PARODET            DPCSID ALIAS           : GFIPSR1O
 TOM PASSWORD          : PARODET            DPCPSW ALIAS           : COMFIRST
 INITIAL STATE         : E                  APM RECEPTION CLASS    : A
 RACF USER             : TOMUSR             RACF GROUP             : -

 PARTNER TYPE          : T
 SESSION PROT.NUM.-T.  : 2    : 0           RSA-DES SECURITY T.    : -
 AUTOMATIC RESTART     : NO

 LINK TYPES            : X    : -           ADJACENT PARTNER       : -
 EFF. TOTAL/IN/OUT     : 010  : 005  : 005  FLOW CONTROL T.   SLD  : -

 SNA: LUNAME  : -         LOGMODE    : -        LOGDATA   : -          DISC  : N
 X25: MCHMSC  : A     REM.ADDR.  : *              LOC.ADDR. : 4
      CUG     : -     UDF        : C0D6C4E3       CHARGE    : 1
      FACILITIES      : -
 IP : ADDR.  : -                PORT : -       FTP: PASV   : -  RIGHTS : -
      HOST   : -                                               PROFILE : -
 NOTE  :
```

Transfer Identification

When the connection has been established, the transfer selection takes place and the file identification is composed of the following structure:

```
Origin + Destination + Filename + File date
```

The origin and destination values are set to the connection identification that you just defined. They can be changed by using the API field of the Extended transfer request. The following table identifies the different ways that you can provide identification using the ODETTE protocol.

Identification	Description
SYMBOLIC from DIRECTORY/REQUEST/SYSIN fields	Symbolic identification from DPCSID/ALIAS (local) and Partner from DIRECTORY/REQUEST fields are the default identification.
ODETTE from DIRECTORY	A specific panel is chained to a standard symbolic Partner definition when the protocol is ODETTE (=2). If any ODETTE-FTP definition is found for the current symbolic Partner name in the directory, either at incoming call control or outgoing call processing, or for the remote received or local sent, it will be processed in place of the symbolic name. The local DPCSID/ALIAS identification can have their own entry in the Partner DIRECTORY to define the local ODETTE-FTP file transfer with a specific identification instead of a symbolic name.
ODETTE from USER EXIT	Specific D1B2PCNX/D1B2RUEX 'U02' application fields are passed both ways following the general identification by the user interface provided by Connect:Express.
ODETTE from REQUEST	The API field can be set to Origin + Destination + file name + file date. The file name can be set to the data set name using the keyword DSN. This field has 26 characters.

The following screen shows the API formatting screen for ODETTE.

```
TOM4200     FORMATTING API, ODETTE
 OPTION ===>                                                        SYSA



 ORIGIN:
 INTERNATIONAL CODE...   ---> 0001              NUMERIC (0001:9999)
 ORGANIZATION CODE....   ---> MY ORGANIZATION   ALPHANUM
 SUB ADDRESS.........    ---> 000001            NUMERIC (000000:999999)

 DESTINATION:
 INTERNATIONAL CODE...   ---> 0001              NUMERIC (0001:9999)
 ORGANIZATION CODE....   ---> REMOTE ORG        ALPHANUM
 SUB ADDRESS.........    ---> 000001            NUMERIC (000000:999999)

 FILE................    ---> F1ODT*   (1)

 DATE................    ---> 010412   (2)
 TIME................    ---> 100000   (3)



             X EXIT, -ENTER- GO ON, -PF1- HELP, -PF3- BACK
```

(1) If the Keyword DSN is typed in this field, it is replaced by the physical name of the file which has 26 characters. This parameter is sent in the ODETTE file identification field.

(2) If the keyword $DATE$ is typed in this field, it is replaced by the current beginning of transfer date.

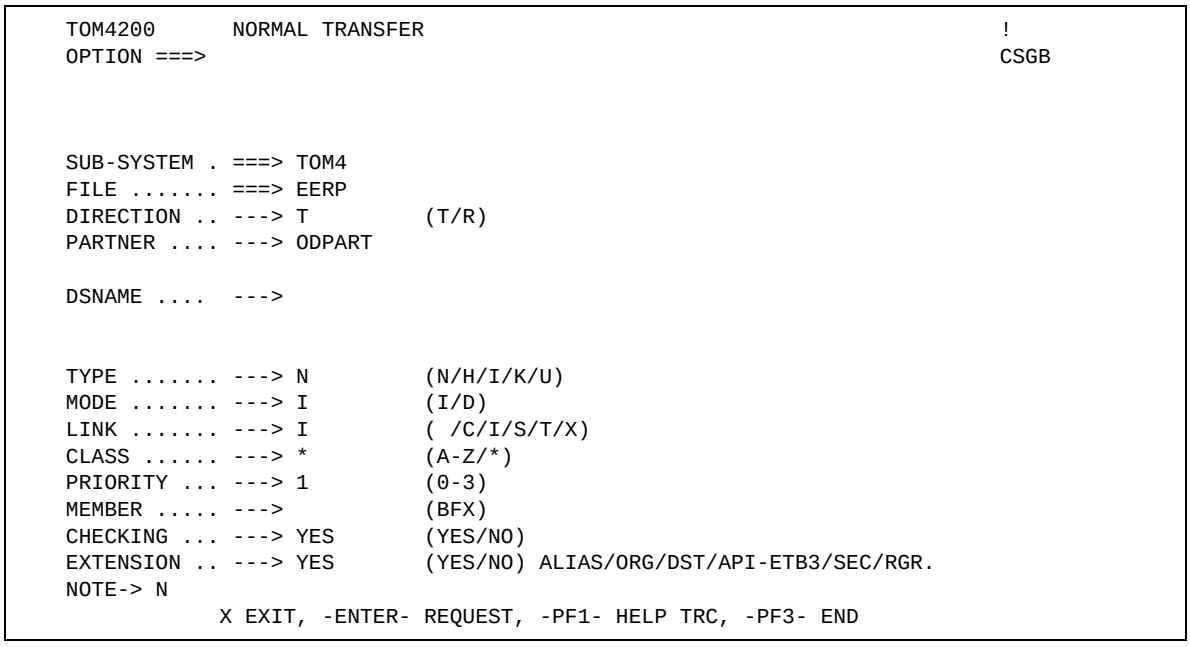
(3) If the keyword $TIME$ is typed in this field, it is replaced by the current beginning of transfer time.

Sending the ODETTE-FTP End to End Response (EERP)

The ODETTE-FTP End to End Response (EERP) is a final acknowledgment message that is forwarded from the final destination to the original sender. Connect:Express can send the EERP message at the end of transfer, or the application can request it later.

You can use an end of transfer user exit to request the EERP upon successful end of transfer. Two user exits are provided, the L1ODEERP and the L1ODEER1 in the *SAMPLIB*. L1ODEERP builds an EERP request for Connect:Express and sends it to the monitor using the Connect:Express programming interface. Then, Connect:Express establishes a new connection to send the EERP. L1ODEER1 tells the current protocol Process to send the EERP as soon as it is the ODETTE speaker, and the EERP is sent during the current connection.

To send the EERP, an application must build the EERP request and send it to Connect:Express using the TSO/ISPF interface, a batch utility, or the programming interface. The following figure shows the TSO/ISPF EERP Request; NO DSNAME is provided.



The EERP symbolic file name must be declared in the files directory, with allocation rule = A as shown in the following screen. This enables you to make a request with no data set name.

```
TOM4200 ---------------- FILES ATTRIBUTES (2/5) -----------------------------
OPTION ===>

SYMBOLIC NAME              : EERP         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 ....... ===> A            0:CREATE/REPLACE 1:PREALL. 2:CREATE
                                          3:EXIT A:APPLICATION SERVER
FILE TYPE ............. ===> S            S/M/P/PU/V/VU/UU/SU
PRESENTATION .......... ===> 04           COMPRESS.,DATA TYPE (01-24)
UNLOAD/RELOAD MEMBER .. ===> -            OPTIONNAL
SECURITY TABLE ........ ===> -            OPTIONNAL

OPTION  : UPDATE                          UPDATE: 04/02/19 06:41 PSR0008
-ENTER- : NEXT SCREEN                     -PF3- : CANCEL
```

Connect:Express Parameters and ODETTE-FTP Fields

Connect:Express parameters include:

- The partner and the file definitions in the Directories and the SYS-SNA/TCP/X25 files.
- The DPCSID from the SYSIN file.
- The transfer request parameters.
- The D1B2PCNX, D1B2RUEX structures exchanged with user exits.

ODETTE-FTP fields include the SSID and SFID fields.

The following examples show how identification works with the ODETTE protocol. The first step is pre-connection which shows where the network is identified. The next step is connection where the partner is identified, followed by selection where the file is identified, and then reporting. Parameters are set in the

command sent, and controlled from the command received. The example below shows how Connect:Express processes identification for an outgoing call with the ODETTE-FTP protocol.

```
REQUEST (PARTNER,FILE) from batch or TSO:

Pre-connection

X25 remote address and user data field or sub address are set from the DIRECTORY

Connection

DPCSID or ALIAS
or
ODETTE IDENTIFICATION from DPCSID/ALIAS entry in the DIRECTORY if any
or
connection user exit:
C02ORIGN            → SSID Command sent

REQUEST
or
ODETTE IDENTIFICATION from PARTNER entry in DIRECTORY if any
or
connection user exit:
C02DESTN            ← SSID Command Review and controlled

The odette identification from the SSID command is expected
to be a symbolic identification or specific ODETTE structure
depending on the DIRECTORY definition of the current
PARTNER processed.

Selection

FILE
Request extension:
ORIGIN/DESTINATION
APPLICATION file Identification (API)
Directory:
ODETTE fields
selection user exit:
U02ORIGN/U02DESTN
U02FLABL             → SFID  Command Sent.

JOURNAL RECORD at end of transfer both OK and not OK:

Z45REQID = Jobname or TSO user
Z45PARTN = PARTNER
Z45TAPID = ORIGIN/DESTINATION/FILE
```

The following screen shows how Connect:Express processes identification for an incoming call with the ODETTE-FTP protocol.

Pre-connection

DIRECTORY or SYSX25 ← X25 call packet/UDF=ODTUDF or sub address=ODTSAD

DIRECTORY : call packet is looked for in the DIRECTORY
If not found, call packet is looked for in SYSX25.
If not found, general ODETTE symbolic name is taken. It must be defined in the DIRECTORY

Connection

DIRECTORY:

PARTNER found ← SSID command received and controlled
or
Connection user exit returned:
CNXPARTN

DIRECTORY: The odette identification from SSID command is expected to be a symbolic identification or specific ODETTE structure depending on the DIRECTORY definition of the current PARTNER processed.

DPCSID/ALIAS
or
ODETTE IDENTIFICATION from DPCSID/ALIAS entry in DIRECTORY if any
or
Connection user exit:
C02DESTN → SSID command sent

Selection

Definition of transfer REQUESTOR:
connection PARTNER in any case

Selection user exit: ← SFID command received and controlled
U02ORIGN/U02DESTN
U02FLABL

DIRECTORIES: U02FLABL must be a symbolic name unless the user exit translates the application name to a FILE profile.

JOURNAL RECORD at end of transfer OK or not OK after it has been accepted by C:X :

Z45REQID = PARTNER
Z45PARTN = PARTNER/CNXPARTN
Z45TAPID = ORIGIN/DESTINATION/FILE/DATE

ODETTE-FTP Commands and Connect:Express Events

The table below lists 4 events that you can have when transferring data with Connect:Express. For each event, there are some special considerations which are identified in the tables of this section.

Event	Description
caller	Initiates a connection with the remote computer.
called	Accepts a connection from the remote computer.
sender	Initiates a WRITE access on the remote computer.
receiver	Accepts a WRITE access from the remote computer.

The tables in this section describe operations on the network, Connect:Express, and the user interface. You can use the following 3 levels to analyze events. The table below describes each level.

Level	Description
Protocol commands	NETWORK TRACE used for analysis
Connect:Express actions	OS/390 and Connect:Express logs used for analysis
USER actions	USER messages and internal traces used for analysis

ODETTE-FTP Caller

ODETTE-FTP caller events occur when you initiate a connection with a remote computer. The following table shows the relationship between what happens during an ODETTE session, what is shown in the Connect:Express environment, and how you can access information. It also includes a description of the process, and reports that you can view.

ODETTE-FTP Caller

Protocol Event	Connect:Express Event	User Interface	Comment	Report
	REQUEST ACCEPTED UNLOAD	L0B2Z20 P1B2PREQ P1B2PRQ2 P1B2PRQ3 TSO/ISPF	User requests Connect:Express to send a file to a Partner.	LOG RCT
	USER EXIT	D1B2PCNX CNX,C	X.25 Network identification. Protocol connection application parameters are passed to Connect:Express.	RCT (init.)
	X.25 Network connection		Response ok or not ok.	
	X.25 Network NOK USER EXIT CONNECTION REJECTED PRC=310	D1B2PCNX CNX,F	Call retry is activated or partner is disabled. The T1B2PCOD table is looked at for a PRC Match.	SYSLOG RCT (waiting) JOURNAL

ODETTE-FTP Caller (continued)

Protocol Event	Connect:Express Event	User Interface	Comment	Report
	X.25 Network OK		Session established.	
REMOTE ODETTE SSRM X'49'			Identification of the remote computer as an ODETTE-FTP partner.	
SESSION SSID X'58'			Local identification is set from DPCSID or ALIAS, symbolic or specific. Response is SSID or ESID. Synchronization, compression, and message length negotiation.	
END SESSION ESID X'46'	USER EXIT CONNECTION REJECTED PRC=00xx	D1B2PCNX CNX,F	Call retry is activated or partner is disabled.	LOG RCT (waiting) JOURNAL
SESSION SSID X'58'			Destination id controlled, from symbolic or specific fields. ESID sent or continue	
END SESSION ESID X'46'	USER EXIT CONNECTION REJECTED PRC=00xx	D1B2PCNX CNX,F	Call retry is activated or partner is disabled.	LOG RCT (waiting) JOURNAL
SSID ok	COMMUNICATION OPENED		The session context is passed to APM. Negotiated to the lower synchronization, compression, and message length values.	LOG RCT (select prot.)
	Anything to do ?		Nothing: change direction else state is SENDER.	
CHANGE DIRECTION CD X'52'			Waiting for next transfer from remote Next state is RECEIVER or CALLED connected.	

ODETTE-FTP Sender

ODETTE-FTP sender events occur when you initiate a write access on a remote computer. The following table shows the relationship between what happens during an ODETTE session, what is shown in the Connect:Express environment, and how you can access information. It also includes a description of the process, and reports that you can view.

ODETTE-FTP Sender

Protocol Event	Connect:Express Event	User Interface	Comment	Report
	USER EXIT FILE and checkpoint ALLOCATION	D1B2RUEX SEL,A D1B2RUEX IEX,I	You can initiate protocol fields and allocate the file.	
SEND FILE SFID X'48'			Response is SFPA or SFNA.	

ODETTE-FTP Sender (continued)

Protocol Event	Connect:Express Event	User Interface	Comment	Report
NEGATIVE ANSWER SFNA X'33'	USER EXIT TRANSFER REJECTED DEALLOCATION DELETE checkpoint	D1B2RUEX TEX,F D1B2RUEX SEL,F	For information PRC=10xx and deallocation. Selection retry is activated or request is disabled. The T1B2PCOD table is looked at for a PRC Match.	LOG RCT (error sel.) JOURNAL
POSITIVE ANSWER SFPA X'32'	Point to RESTART. BEGINNING OF TRANSFER COMMAND TRANSFER ACCEPTED			LOG
	USER EXIT TRANSFER STARTED	D1B2RPEX PEX,B (open)	Initiate beginning of transfer.	LOG RCT (started)
	READ record USER EXIT	D1B2RPEX PEX,M + record	Read, pass to user, compress fill buffer and send until synch reached.	
CREDIT CDT X'43'	Write checkpoint		Synch point is stored.	RCT (started)
END OF FILE EFID X'54'	USER EXIT CLOSE file	D1B2RPEX PEX,C (close)	End of file detected. Record and byte counts are sent. Response is EFPA or EFNA.	
NEGATIVE ANSWER EFNA X'35'	USER EXIT TRANSFER INTERRUPTED	D1B2RUEX TEX,F D1B2RUEX SEL,F	Transfer is interrupted. If any, next transfer processed SFID If not, CHANGE DIRECTION.	LOG JOURNAL RCT (interr.) SMF
POSITIVE ANSWER EFPA X'34'	USER EXIT END OF TRANSFER COMMAND TRANSFER ENDED DEALLOCATION Delete checkpoint PURGE request	D1B2RUEX TEX,E D1B2RUEX SEL,E	Transfer is ended, acknowledged by receiver. If any, next transfer processed SFID. If not, CHANGE DIRECTION.	LOG JOURNAL SMF

ODETTE-FTP Called

ODETTE-FTP called events occur when you accept a connection from a remote computer. The following table shows the relationship between what happens during an ODETTE session, what is shown in the Connect:Express environment, and how you can access information. It also includes a description of the process, and reports that you can view.

ODETTE-FTP Called

Protocol Event	Connect:Express Event	User Interface	Comment	Report
	Incoming call accepted		Network requests Connect:Express to establish session.	
User data field or sub address	Identification of ODETTE-FTP caller Partner or ODETTE-FTP profile. Check resources. Allocate APM/EFF		Remote computer requests Connect:Express to establish ODETTE-FTP session. Response is SSRM or lib.	RCT (init.)
lib.	USER EXIT	D1B2PCNX CNX,F	Connect:Express resources are not up: Connect:Express, RCT, PCT, FCT inactive Partner not found or found not OK	LOG:
ODETTE SPEAKING SSRM X'49'			Connect:Express 's ODETTE facility is ready.	RCT (init.)
SESSION SSID X'58'	USER EXIT	D1B2PCNX CNX,C	Partner identification. Protocol connection application Parameters are passed to USER. Control by the user. Response is SSID or ESID.	RCT (init.)
END SESSION ESID X'46'	USER EXIT CONNECTION REJECTED if previously accepted. PRC=00xx	D1B2PCNX CNX,F		LOG
SESSION SSID X'58'	COMMUNICATION OPENED		Accepted by USER and Connect:Express. Local identification passed to partner Next is SFID, ESID or CD Control passed to APM.	LOG RCT (select prot.)
CHANGE DIRECTION CD X'52'	Anything to do ?		Nothing: ESID else state is SENDER	
END SESSION ESID X'46'	USER EXIT CONNECTION ENDED PRC=0000	D1B2PCNX CNX,E		LOG

ODETTE-FTP Receiver

ODETTE-FTP receiver events occur when you accept a write access from a remote computer. The following table shows the relationship between what happens during an ODETTE session, what is shown in the Connect:Express environment, and how you can access information. It also includes a description of the process, and reports that you can view.

ODETTE-FTP Receiver

Protocol Event	Connect:Express Event	User Interface	Comment	Report
SFID SFID X'48'	USER EXIT	D1B2RUEX SEL,A	The user receives protocol fields, identifies the transfer, and sends Connect:Express the symbolic file name to be processed.	
	Check resources allocate and catalog file allocate transfer checkpoint		Connect:Express checks resources and returns exit names and data set names to the APM.	
	USER EXIT BEGINNING OF TRANSFER COMMAND TRANSFER ACCEPTED or REJECTED	D1B2RUEX IEX,I	Response is SFPA or SFNA. The user can allocate the file.	LOG
NEGATIVE ANSWER SFNA X'33'	USER EXIT TRANSFER REJECTED DELETE data set and check point PURGE request	D1B2RUEX TEX,F D1B2RUEX SEL,F	The user exit receives the TRC=xxxx and can perform deallocation. Connect:Express clears the transfer context.	LOG JOURNAL
POSITIVE ANSWER SFPA X'32'	Point to RESTART TRANSFER ACCEPTED			LOG
First data	USER EXIT TRANSFER STARTED	D1B2RPEX PEX,B (open)	Initiate beginning of transfer.	LOG RCT (started)
	Process I/O buffer USER EXIT	D1B2RPEX PEX,M + record	Uncompress, pass to user, fill buffer, write until synch reached.	
CREDIT CDT X'43'	Write checkpoint Force physical I/O		Synch point is stored.	RCT (started)
END OF FILE EFID X'54'	USER EXIT CLOSE file	D1B2RPEX PEX,C (close)	Response is EFPA or EFNA.	
NEGATIVE ANSWER EFNA X'35'	USER EXIT TRANSFER INTERRUPTED	D1B2RUEX TEX,F D1B2RUEX SEL,F	Transfer is interrupted.	SYSLOG JOURNAL RCT (interr.) SMF

ODETTE-FTP Receiver (continued)

Protocol Event	Connect:Express Event	User Interface	Comment	Report
POSITIVE ANSWER EFPA X'34'	USER EXIT END OF TRANSFER COMMAND TRANSFER ENDED DEALLOCATION Delete checkpoint PURGE request	D1B2RUEX TEX,E D1B2RUEX SEL,E	Transfer is ended, acknowledged by receiver.	LOG JOURNAL SMF

ODETTE-FTP Parameters and Connect:Express Definitions

Connect:Express definitions are parameters stored in Connect:Express files or user interface fields. They are placed in protocol fields when sending ODETTE commands or compared with protocol parameters when receiving ODETTE commands. Parameters can be considered for incoming commands or for outgoing commands.

The following table shows the relationship between the protocol phase, the protocol parameters and their length, and the Connect:Express definitions or keywords and their length. It also includes a description of the process, and reports that you can view.

ODETTE-FTP Parameters and Connect:Express Definitions

Protocol phase	Protocol param,lg	C:X param,lg	Description	Report
connection	p1,25	Incoming: PARTNER,8 (specific) C02ORIGN C02DESTN Outgoing: DPCSID,8 ALIAS,8 (specific) C02ORIGN C02DESTN	Session partner name. This name is a default symbolic name or an ODETTE-FTP specific identification. Remote: Partner is controlled from his symbolic name (find in DIRECTORY if incoming, given at request time if outgoing) or specific name if given in Connect:Express DIRECTORY or C02ORIGN (incoming) or C02DESTN (outgoing) when given by USER EXIT. Local: Partner name sent is the symbolic name (DPCSID or ALIAS) or specific name if given in Connect:Express DIRECTORY or C02ORIGN (outgoing) or C02DESTN (incoming) when given by USER EXIT.	Symbolic: LOG RCT JOURNAL
connection	p2,8	Incoming: PASSWORD,8 Outgoing: DPCPSW (ALIAS,8)	Password is verified in the Partner directory. It is sent from the SYSIN DPCSID= or the Partner directory alias.	
connection	p3,5	SPBUFLNG,5 STDMSG= MAXEXT=	Session exchange buffer length. In the T1B2PS2x tables. The ANM buffer length parameters from the SYSIN file must fit within the protocol's message length.	
connection	p4,1 S, R or B	SPDIREC,1	Session direction In the T1B2PS2x tables. T, R, *, or **U** (unique): only same direction transfers are chained in one session	

ODETTE-FTP Parameters and Connect:Express Definitions (continued)

Protocol phase	Protocol param,lg	C:X param,lg	Description	Report
connection	p5,1 Y or N	SPCOMPR,1	Compression capabilities In the T1B2PS2x tables.	
selection	p6,26	FILE Symbolic API U02FLABL	Transfer file name. The symbolic file name (8 characters) can be replaced by an application name (26 characters) which is set in the API field or the 26 character data set name using the DSN keyword in the API name field. The REMOTE DSN field can be used with symbolic keywords. Note: The online help provides a list of available symbolic variables. Press <PF1> and type VS in the Option to display this list.	LOG RCT JOURNAL SMF
selection	p7,6	System DATE API U02FDATE	Transfer file date.	RCT JOURNAL
selection	p8,6	System TIME API U02FTIME	Transfer file time.	RCT JOURNAL
selection	p9,25	PARTNER DESTINAT. API U02DESTN	Final recipient Outgoing: from request extension DESTINATION or API fields. Incoming: only reported	RCT JOURNAL
selection	p10,25	PARTNER ORIGIN API U02ORIGN	Sender of the file Outgoing: from request extension ORIGIN or API fields. Incoming: only reported	RCT JOURNAL
selection	p11,5	File DCB DIRECTORY UEXALREC	Maximum record size From the file DCB INPUT, or passed to Dynalloc output. Controlled if given in the file DIRECTORY on reception.	
selection	p12,7	File DSCB File DIRECTORY UEXALSPA	Maximum file space From the file DSCB INPUT, or passed to Dynalloc output. Controlled if given in the file DIRECTORY on reception and if the SYSIN LRECLT parameter is yes.	
selection	p13,9	internal Checkpoint	Restart position Stored at synchronization points in the request checkpoint.	
selection	p14,9	internal Checkpoint	Restart position answer Stored at synchronization points in the request checkpoint.	
selection	p15,2	PRC UEXPRC PEXPRC	Protocol return code PRC=10xx	RCT LOG JOURNAL
selection	p16,1	internal	Answer retry flag.	
transfer	p17,9	internal	Number of records transferred.	RCT LOG JOURNAL SMF

ODETTE-FTP Parameters and Connect:Express Definitions (continued)

Protocol phase	Protocol param,lg	C:X param,lg	Description	Report
transfer	p18,12	internal	Number of bytes transferred.	RCT JOURNAL SMF
transfer	p19,1 Y or N	internal	Change direction asked. Connect:Express always sets it to N.	
connection	p20,1 Y or N	internal	Restart option. Connect:Express always sets it to Y	
connection	p21,5	internal	Future use.	
connection	p22,17	internal	Ready message sent by called to caller.	
connection	p23,1	internal	End of command byte.	
connection	p24,1	SPLOGIC	Use of CRC control option. In the T1B2PS2x tables. Y or N T1B2PS21 reserved with CRC USAGE = YES	
selection	p26,1 F, V, U, T	File DCB DIRECTORY	File record format. From the file DCB INPUT, or passed to Dynalloc output. Controlled if given in the file DIRECTORY on reception and if the SYSIN URECFMT parameter is yes.	

ODETTE-FTP Allocation Attributes and IBM OS/390 DCB

No Connect:Express allocation attribute exchange facility is available with the ODETTE-FTP protocol. The file record formats supported by P26 of the Odette parameter are described in the table below. IBM OS/390 record formats are not represented.

ODETTE-FTP Format	Description
Fixed	Blocking is processed by Connect:Express automatically (FB) or from the DCB given in the FILE DIRECTORY fields (F, FB, FBA, FBS, ...).
Variable	Blocking is processed by Connect:Express automatically (VB) or from the DCB given in the FILE DIRECTORY fields (V, VB, VBS, VBA, U...). The logical value of protocol LRECL (P11) is added RDW and BDW bytes.
Undefined	This has nothing to do with the IBM OS/390 Undefined format. An ODETTE-FTP undefined format file is one single record the length of which is the total length of the file. Connect:Express never sends undefined format files, but can receive such files. The file is processed as a Variable file (V or U); the record length is that of the file allocated. U is the default with LRECL= 32760. The DCB can be given in the directory with any variable format (V,VB,VBS,VBA, U...).

ODETTE-FTP Format	Description
Text	This is unknown in an IBM OS/390 environment. An ODETTE-FTP text format file is a variable length record file in which any record is terminated by CRLF (Carriage Return + Line Feed) bytes. Text file processing is conditioned by the TYPE OF DATA parameter defined in the presentation table (Table T1B2PP05 is an example provided). The file is processed this way: Sending a file, any format: the EBCDIC file is translated to ASCII (L1APM050), and sent as a 'T'– Text file, with '0D0A' added at end of each record (DATA TYPE = ASCII). Receiving a text file, any format : Both text files sent as 'T'– Text files in the Odette parameter and text files sent as 'F'– Fixed or 'V'– Variable or 'U'– Unstructured are suppressed 0D0A (DATA TYPE = ASCII), and translated to EBCDIC (L1APM050). NOTE: HFS files are not supported by Connect:Express ODETTE functionality.

Receiving a Sequential File

If the DCB attributes are given in the File Directory on the receiver side, they are used to control the record format, record length, and to allocate files. The &x keywords in the File directory are not available to get the physical attributes from the sender. Giving the DCB in the directory is the only means of creating the same file on the sender side.

Depending on the file definition, the file received is allocated from the ODETTE parameters if the DCB is not given. If the DCB is given, the ODETTE parameters are ignored and you have the option of using automatic blocking and automatic space computing.

Text files are processed from the type of data declared in the presentation table (T1B2PPxx). If data is declared locally as text, it is received as text even if the file is not declared as a text file by the sender. The blocksize is not known by ODETTE-FTP, so the local value is never transmitted to the remote.

When receiving a sequential file **with no DCB** entered in the directory, the following ODETTE fields are used for file allocation.

Odette	Allocation by receiving Connect:Express (No DCB given)
Odette P26 = F	RECFM = FB
Odette P11 = rrrr	LRECL = rrrr; BLOCKSIZE optimized (unless SMS is used)
Odette P12 = ssss	Primary space = blocks equivalent to ssss bytes + 10%, no secondary space TEXT data type (A): search for Carriage Return character and deleted. The maximum size of the line must fit the record length. Records are padded with blanks, a null record is kept as a blank record.
Odette P26 = V	RECFM = VB
Odette P11 = rrrr	LRECL = rrrr+4; BLOCKSIZE optimized (unless SMS is used)
Odette P12 = ssss	Primary space = blocks equivalent to ssss bytes + 10%, secondary = 20% TEXT data type (A): search for Carriage Return character, it is deleted. The maximum size of the line must fit the record length. Records are not padded, a null record is kept as a null record.
Odette P26 = U	The DCB must be defined in the file attributes.
Odette P26 = T	The DCB must be defined in the file attributes.

When receiving a sequential file **with the DCB** entered in the directory, the following ODETTE fields are used for file allocation.

Odette	Allocation by receiving Connect:Express (DCB given in the directory)
Odette P26 = F	RECFM = Any value if RECFMT=N (no control) or RECFM = F,FB,FBA only.
Odette P11 = rrrr	LRECL = rrrr only; BLOCKSIZE compatible.
Odette P12 = ssss	P12 omitted. If the space defined is not large enough, abend D37 can occur. TEXT data type (A): search for Carriage Return character, it is deleted. The maximum size of the line must fit the record length. Records are not padded, a null record is kept as a blank record.
Odette P26 = V	RECFM = Any value if RECFMT=N in the SYSIN file (no control) or RECFM = V,VB,VBA,VBS,U only.
Odette P11 = rrrr	LRECL = any if LRECLT=N in the SYSIN file (no control) or rrrr+4 only; BLOCKSIZE compatible.
Odette P12 = ssss	P12 omitted. If the space defined is not large enough, abend D37 can occur. **TEXT** data type (A): search for Carriage Return character, it is deleted. The maximum size of the line must fit the record length. Records are not padded, a null record is kept as a null record if RECFM = V*. A null record is skipped if RECFM = U.
Odette P26 = U	RECFM = V,VB,VBA,VBS,U only.
Odette P11 = 0	LRECL = any; BLOCKSIZE compatible.
Odette P12 = ssss	Primary space = blocks equivalent to ssss bytes + 10%, secondary = 20% The file is filled up until the last short record. **TEXT** data type (A): search for Carriage Return character, it is deleted. The maximum size of the line must fit the record length. Records are not padded, a null record is kept as a null record if RECFM = V*. A null record is skipped if RECFM = U.
Odette P26 = T	RECFM = any.
Odette P11 = 0	LRECL = any; BLOCKSIZE compatible.
Odette P12 = ssss	Primary space = blocks equivalent to ssss bytes + 10%, secondary = 20% **TEXT** data type (A): search for Carriage Return character, it is deleted. The maximum size of the line must fit the record length. Padding and null record process depends on the record format.

Sending a Sequential File

From the existing file structure, Connect:Express sets the Odette parameters with the standard values. If the data type is set to TEXT, a carriage return character is added at the end of each record and the file is sent as a TEXT file.

When sending a sequential file, the following ODETTE fields are used.

File sent by Connect:Express	Odette
RECFM = F,FB,FBA	P26 = F
LRECL = rrrr	P11 = rrrr

*If the file is compressed, P12=t*size track*8

File sent by Connect:Express	Odette
Space = t tracks	P12 = t * size track
	TEXT data type (A): P26 = T P11 = 0 P12 = t * size track
RECFM = V,VB,VBA	P26 = V
LRECL = rrrr	P11 = rrrr - 4
Space = t tracks	P12 = t * size track
	TEXT data type (A): Odette P26 = T Odette P11 = 0 Odette P12 = t * size track
RECFM = VBS, U	Odette P26 = V
LRECL = block size	Odette P11 = block size
Space = t tracks	Odette P12 = t * size track
	TEXT data type (A): Odette P26 = T Odette P11 = 0 Odette P12 = t * size track

*If the file is compressed, P12=t*size track*8

Chapter 6

Troubleshooting

This chapter describes common problems that you may encounter, tools that you can use to identify the problem, and corrective action that you can take. It also includes information about enhancing system performance and getting help from Technical Support.

Tracking Events

You can access messages and return codes online or with a batch procedure. Online information can help you analyze the problem, control operations, and take action quickly when there is a problem. This section describes first-level diagnostic tools; the next section describes trace tools used for deeper investigation.

Using Diagnostic Tools

Connect:Express provides several diagnostic tools to help administrators when a problem occurs. Some tools are always enabled, some tools can be enabled to get more information, and others can be enabled to track specific data flows. The most common diagnostic tools are listed in the table below:

Tool	Description
TOM SYSLOG file	Used to track chronological events. This file tracks basic information such as file name, partner name, date time, and request number. You can also track all the related events such as a specific data flow, a specific link, or a recurrent return code.
TOM SYSMSG file	Used to track operational events such as initialization.
APM SYSLOG file	Used to track chronological events and get information about file allocation.
AFMLOG file	Used to track chronological FTP transfer events.
APM SYSMSG file	Used to track operational events such as initialization.
ANM SYSMSG file	Used to track operational events such as initialization or diagnostics.
SYSPRINT file	The utilities options and the API provide a SYSPRINT file in which you can find all information related to actions made.
Journal file	Used to archive transfer reports. You can use the filter screen to select what may be related to your current problem.
Request Control Table (RCT)	Gives the current status of transfers.

Tool	Description
Help	Provides information about return codes.
JES SYSOUT files	Contains system messages that can help you take the appropriate corrective action.

Messages in the logging file can help you determine what happened with a transfer. Refer to chapter 4 for detailed information about incoming and outgoing transfers, such as which Connect:Express parameters are involved in establishing a session with a Partner. Refer to Chapter 5 for information about the relationship between Connect:Express parameters and PeSIT or Odette protocol parameters.

Additionally, you can activate network traces or any system tracking facility. You can also use special exits to get online information. Advanced users can install automatic control processes for alerts and statistics using the interfaces provided by the product. For example, the Utilities package provides powerful automation tools. See the *Utilities Guide* for more information about this option.

Using User Exits to Trace Events

You can use user exits to trace events. Connect:Express provides some standard user exits that are described in the table below.

User Exit	Description
L1GFICN1	You can use this exit to track connections. It notifies you about incoming and outgoing connection failures in the TOM SYSMSG file. This connection exit must be defined in the T1B2PCNT table (TSO/ISPF 3.3.CNT).
L1GFICN1	You can use this exit to track transfer selection. It notifies you about incoming and outgoing transfer selectIon failures in the SYSMSG file. This selection exit must be defined in the T1APMSRT table (TSO/ISPF 3.3.SRT).
L1EX£AE2	You can use this exit to track transfer selection. It notifies you about the protocol parameters received during the transfer selection in an AE2* print file. This beginning/end of transfer exit must be defined on the file attributes 3/4 screen (TSO/ISPF 1.2), in the START EXIT, or END EXIT fields. The AE2* print file name consists of the effector number, the step (Init., Term.), and the status (Init, End, Failed). AE203IEI is for a transfer done by ‘effector’ number 03, exit during Initialization of transfer, status ‘I.’ AE205TEF is for a transfer done in ‘effector’ number 05, exit during Termination of transfer, status ‘F’.

Activating Traces

When online information is not sufficient to analyze a problem, you may need to activate traces. This section describes how you can track incoming connection errors only, run protocol traces, or run network traces.

Tracking Incoming Connection Errors

If a problem occurs when you are trying to establish a connection with an incoming call from a Partner, you may need to set up a trace. If it is not possible to activate a VTAM or TCP/IP trace, you can activate an internal trace for one Partner. The critical connection step happens during the network connection. A trace enables you to find out what has been received and who the caller is.

To use the general trace facility, you must first activate the trace service for Connect:Express using a modify command, as shown below.

```
Enable trace : F TOMJOB, TRACE=E
Disable trace : F TOMJOB, TRACE=H
```

The SYSMSG file displays the following message.

```
10.30.07 STC07436  TOM4204I MODIFY COMMAND TRACE=E
10.30.07 STC07436  TOM4029I TRACE IS NOW ENABLED
```

Then, any unidentified call displays a message in the SYSLOG file.

```
14:29:35 TRACE  UNKNOWN  :  910  4 T=0 G=   U=  "PESIT   INTRUS PASSWORD  "
```

In the example above, an unknown system made an X.25 incoming call. The calling address was 910, the sub-address was 4, the taxation rule = 0, there is no closed user Group, no User data filed, and the first message received appears in quotes.

Tracking One Partner

From the Partner list screen, you can activate a trace for one partner by entering a “+” in the command field. You can enter a “-” to deactivate the trace.

Note: If you update a Partner record, you will deactivate the trace.

```
TOM4200     PARTNERS 1/4    F=    ID=          MODE= *            Row 1 of 500
 OPTION ===>                                                     PAGE -> PAGE

 TOM4         -< 'COMMANDS'   -PF3- RETURN   -PF7/8/10/11- SCROLL
 SYSA
 --------------< S/F DETAILS  D DIRECTORY    E ENABLE   H DISABLE
 !
 ! PARTNER   STATUS     PROTOCOL / TABLE  LINK       TYPE   REST.  EFF.   MODE
 V           1          2          3      4          5      6      7        8
 _ $$A       ENABLED    PESIT-D    30     X25               NO     000/004  -
 _ AS400IP   ENABLED    PESIT-E    52     MIXED      XCI    YES    000/008  -
 _ CXNT      ENABLED    PESIT-D    32     TCP-IP            YES    000/020  -
 _ ODETTE    ENABLED    ODETTE     20     X25               YES    000/002  -
 + OS390     ENABLED    PESIT-D    50     SNA               YES    000/050  -
 _ UNIX      ENABLED    PESIT-D    50     X25               YES    000/020  -
 _ USRTOM    ENABLED    PESIT-D    50     TCP-IP            YES    000/002  -
 *** END OF LIST
```

The online help shows these two commands.

```
TOM4200 -- HELP : PARTNERS STATUS ------------------------------------------
 OPTION ===>                                              'B'/'INDEX'/'TOP'
             -ENTER- : NAMES OF ZONES.
             O       : COMMAND IN OPTION AREA.

 VIEW/UPDATE A SPECIFIC ENTRY, IN THE SELECTION FIELD:
  +   : ACTIVATE TRACE FOR THIS PARTNER.
  -   : DESACTIVATE TRACE FOR THIS PARTNER.
  D   : DIRECTORY DEFINITION.
  E   : ENABLES THE PARTNER.
  F   : DISPLAYS FILES DEFINED FOR THIS PARTNER.
  H   : DISABLES THE PARTNER.
  S   : GIVES THE STATUS OF REQUESTS FOR THIS PARTNER IF THEY EXIST
        AND THEN TRANSFERS SIMILARLY.
  X   : DISPLAY SUBSEQUENT SCREENS.

 YOU MAY SELECT SEVERAL PARTNERS AT A TIME.
```

The SYSLOG file displays the following information.

```
14:27:16 PARTNER  TRACE=ON    BY USER001
```

Then any rejected incoming calls from this partner generate a message in the SYSLOG file, as shown below.

```
14:27:42 TRACE PARTNER  :  158.023.023.004  "PESIT   PARTNER  PSW     "
14:27:43 REQUEST 06800011 REJECTED -> PARTNER   SRC=0000 TRC=2134  PRC=3310
```

Running the Trace Manager

A trace utility is provided to analyse information and data exchanged during file transfers, whatever the protocol is.

This is an address space called Auxiliary Trace Manager (ATM). It can be activated and deactivated by a command to the Connect:Express monitor. The trace manager records trace requests that you pass to the monitor through commands: a trace request determines which partner or group of partners to trace and whether the trace must include both negotiation commands and file transfer data or only negotiation commands.

Each trace file is written in a SYSPRINT file and shows all events of one protocol session, from session initialization to session termination. The SYSPRINT file is identified from the request number or from a time stamp if the request number is unknown (this is the case with an FTP server session).

Information is structured in XML format. It includes protocol messages, a translation of the protocol commands, and values that Connect:Express uses during execution of the file transfer.

The trace aspect differs from one protocol to another (PeSIT, FTP, Etebac3, Odette), but Connect:Express fields conform to a general table shown below.

Configuring, Starting and Stopping the Trace Manager

The ATM address space is started by the monitor from a JCL that you copy in the *PROCLIB* and the name of which you indicate in the Connect:Express SYSIN file.

Configuration - JCL and SYSIN

A JCL example £ATMPRC is provided in *INSTLIB*. Proceed the same way as for customizing APM and ANM procedures: copy the ATM procedure in *PROCLIB* and customize it.

Indicate the name of the procedure to start in the SYSIN file as shown below:

ATMPRC='procedure name'

You must pass a command to Connect:Express to start the trace manager.

Commands to Connect:Express

Use the following command to start the ATM:

```
/F jobmon,S ATM
```

Use the following command to stop the ATM:

```
/F jobmon,P ATM
```

When Connect:Express stops, all active address spaces are stopped, including the trace manager.

Refer to the ATM SYSLOG file to know about the trace activity.

```
05/12/20 09:25:23 ATM?002I SYSLOG FILE OPENED
05/12/20 09:25:23 ATM?006I EXEC PARMS: SSN=TOM4 REQ=128
05/12/20 09:25:23 ATM4030I L1ATMTRC IS INITIALIZING
05/12/20 09:25:23 ATM4022I L1ATMTRC INITIALIZATION COMPLETE
05/12/20 09:25:23 ATM4009I ATM INITIALIZATION COMPLETE
05/12/20 09:25:49 ATM4016I TYPE : PART- ALL
05/12/20 09:25:49 ATM4017I MASK : FTP4
05/12/20 09:25:49 ATM4015I TRACE REQUEST ACCEPTED ID=00000001
05/12/20 09:32:35 ATM4032I TRACE FILE OPENED R0042562
05/12/20 09:32:35 ATM4032I TRACE FILE OPENED A9323570
```

Refer to Connect:Express SYSMSG file to know about operator activity and ATM status.

```
09.25.13 STC02840  TOM4084I APM 01 INITIALIZATION COMPLETE
09.25.23 STC02840  TOM4204I S ATM
09.25.23 STC02840  S     PSRAT4.PSRAT4,SSN=TOM4,REQ=128
09.25.23 STC02840  TOM4192I ATM PROCEDURE IS INITIALIZING  ASID=00134
09.25.23 STC02840  TOM4194I ATM STARTED ASID=00134 TRACE IS ACTIVE
09.25.49 STC02840  TOM4204I TRACEPAR=(FTP4,ALL)
09.25.49 STC02840  TOM4198I ATM TRACE REQUEST ACCEPTED ID=00000001
```

Trace Request to the Monitor

When the ATM is initialized, you can request to trace one partner or a group of partners, as shown below:

```
/F jobmon,TRACEPAR=(Partner,type)
```

The *partner* parameter can be either a partner name or a root such as 'PAR*' for all partners the name of which begins with 'PAR'.

Note: The partner must exist in the Partners directory.

The *type* parameter indicates if the trace must include both negotiations and data or only negotiations.

Examples:

To trace partner PARIS, only negotiations:

```
/F jobmon,TRACEPAR=(PARIS,DIA)
```

Note: Partner 'PARIS' must exist in the directory.

To trace all partners which name begins with 'PAR', negotiations and data:

```
/F jobmon,TRACEPAR=(PAR*,ALL)
```

The trace request is identified by a number. To disable the trace request, use the following command to Connect:Express:

```
/F jobmon,TRACEOFF=n
```

'n' is the trace request number that Connect:Express returns after the ATM recorded the request:

```
09.25.49 STC02840  TOM4198I ATM TRACE REQUEST ACCEPTED ID=00000001
09.28.09 STC02840  TOM4204I TRACEOFF=1
09.28.09 STC02840  TOM4200I ATM TRACE REQUEST DELETED  ID=00000001
```

You can activate 128 trace requests at the same time.

Executing a Protocol Trace

When a protocol session opens, Connect:Express inquires the ATM to know if the current partner is required a trace, and which type of trace (dialog or all). If such a trace request exists, the ATM opens a SYSPRINT file which name is computed from the request number (Rnnnnnnn) or from the current time (Ahmmsscc). The ATM SYSOUT file shows the trace files and the SYSLOG file.

```
   PSRTOM4  PSRTOM4  DEV$400  STC02840 PSRTOM4    NS  FB 1573
   PSRAN4   PSRAN4   DEV$400  STC02841 PSRAN4     NS  FB 2144
   PSRAP4   PSRAP401 CEE$400  STC02842 PSRAP4     LO  FF 1361
   PSRAP4   PSRAP402 CEE$400  STC02843 PSRAP4     LO  FF 1328
 ? PSRAT4   PSRAT4   DEV$400  STC07291 PSRAT4     NS  FB 2332
```

```
   JESJCLIN                    1 PSRAT4   X                      2
   JESMSGLG JES2               2 PSRAT4   X                      2
   JESJCL   JES2               3 PSRAT4   X                     52
   JESYSMSG JES2               4 PSRAT4   X                      2
   $INTTEXT JES2               5 PSRAT4   A                      0
   SYSLOG   PSRAT4           104 PSRAT4   V                      0
   R0042562 PSRAT4           110 PSRAT4   A LOCAL               83
   A9323570 PSRAT4           111 PSRAT4   A LOCAL               84
   R0042601 PSRAT4           112 PSRAT4   A LOCAL              191
   R0042602 PSRAT4           113 PSRAT4   A LOCAL              188
   R0042604 PSRAT4           114 PSRAT4   A LOCAL              191
   R0042605 PSRAT4           115 PSRAT4   A LOCAL              106
   R0042607 PSRAT4           116 PSRAT4   A LOCAL              191
   R0042608 PSRAT4           117 PSRAT4   A LOCAL              188
   R0042610 PSRAT4           118 PSRAT4   A LOCAL              191
   R0042611 PSRAT4           119 PSRAT4   A LOCAL              188
```

Refer to “Using Protocol Traces” on page G-1 for information on reading the ATM SYSOUT file and trace files based on the file transfer protocol.

Running Network Traces

It may be necessary to take network traces for X.25 and SNA links. To do this, you can activate a TYPE=BUFFER trace on the ANM applications:

- ANMAPP01 for SNA and pad connections
- ANMAPP02 for X.25 (Date)
- ANMAPP03 for X.25 (Date, Gate, Pad)
- ANMAPP06 for X.25 (Gate)

You can activate TYPE=BUFFER traces on MCH LUs, and you can activate TYPE=LINE traces on any MCH.

The following example shows ACFTAP SYSIN parameters that you can use to edit the trace.

```
000048 INPUT=LINE
000049 NODE=MCH056
000050 SOURCE=GTF
000051 LUPRT=YES
000052 LSPRT=YES
000053 LDPRT=YES
000054 PRINT=YES
000055 RRSUP=NO
000056 VTPRT=YES
000057 GSPRT=YES
000058 SUMMARY=YES
000059 SDPRT=YES
000060 SSPRT=YES
000061 DTPRT=YES
000062 NEPRT=YES
000063 NPPRT=YES
000064 NTPRT=YES
000065 LUPRT=YES
000066 LONGPIU=YES
000067 READ
000068 GO
000069 QUIT
```

Common Problems

When setting up the transfer operations environment, you may encounter some of the common problems described in this section. For example, connection failures can occur if you did not agree with your partner about addresses, names and passwords. Transfer errors can occur if you did not agree with your partner about the file to transfer, and allocation errors can occur if your local file definition was not correct.

Connection Errors

You can view incoming connection errors in the Connect:Express SYSLOG file, and you can check outgoing connections in the Request Control table. You can also do any of the following to get more information about connection errors:

- Check the request status and TRC on the ISPF 2.1.R screens. Press <PF11> to scroll, then place your cursor on an entry and press <ENTER> to access help.

- Check the monitor SYSLOG file for network return codes and links used.
- Check the drivers status on the ISPF 2.1.N screens.
- Activate the tracking incoming connection error facility.
- Perform a network trace.

Refer to “Incoming and Outgoing Transfers” on page 4-1 and “PeSIT and ODETTE-FTP Protocols” on page 5-1 for more information.

Transfer Errors

You must view incoming transfer request errors in the Connect:Express SYSLOG file. In the first step of negotiating a transfer request, an entry is created in the Request Control table. If a problem occurs before the transfer is started, no record is kept in the Request Control table or in the Journal file. You can check the Request Control Table for outgoing transfers and incoming transfers after they have started. You can also do any of the following to get information about transfer errors:

- Check the request status and TRC on the ISPF 2.1.R screens. Press <PF11> to scroll, then place your cursor on an entry and press <ENTER> to access help.
- Check the monitor SYSLOG file for return codes and links used.
- On the request panel 2.1.R, press <PF1> and then enter the command 'WR' or 'RA' to display the possible reasons for the Waiting Request message.
- Use the L1EX£AE2 selection exit SYSPRINT file to check for the protocol parameters received.
- Perform a network trace.

Refer to Chapter 4 *Incoming and Outgoing Transfers* and Chapter 5 *PeSIT and ODETTE-FTP Protocols* for more information.

Allocation Errors from Creating a File

During reception of a new file, the allocation operations can fail from different components. If this happens, you can check the Connect:Express SYSLOG file for the following messages:

```
REQUEST 00000008 00000008 ALLOCATE FAILED SPM= B 00013680 00006 000001 -  -  00000080
00013680
REQUEST 00000008 00000008 ALLOCATE FAILED SMS= -    -    -    -      00000001
REQUEST 00000008  00000008 ALLOCATE FAILED DYN= E 970C I 0000 E 0000 I 0000 S 000042CD
R90  S
REQUEST 00000008 FICTST   ALLOCATE FAILED DSN=PSR$TST.GFIPSR4.D001221           Ep I
E  I  S      REQUEST 00000008 FICTST   GFIPSR4          SRC=970C TRC=2085  PRC=2204
```

These messages are explained in the help screen. From the TSO/ISPF interface, type OPTION ==> RC from any screen. Then select –SRC and – TOM SYSLOG. The results display in the screen below.

```
 -- S: DETAIL
 V
 _ TOM SYSLOG, 'ALLOC FAILED SPM= / SPW= / SMS= / DYN= / DSN='
```

You can also do any of the following to get more information about allocation errors:

- Check SDSF ('INPUT ON') the JESmsg at the top of the SYSOUT file for « IKJ ? ? ? allocation-messages »
- Check for SMS or VAM restrictions. You can perform a job or ISPF 3.2 for this DSN disp= NEW with the same parameters.
- Activate a snapdump. You can get a DYNALLOC trace in the Monitor //SNAPDUMP file. Send the following OS/390 command to Connect:Express: « F monitor,SNAP=E » followed by: « F monitor, SNAP=ON,12.» Then reproduce the allocation error, and keep the snapdump file.

Address Space Errors

An abend can occur in a Connect/Express address space due to a product problem, but most system abends, like ABENDD37, are common and can be fixed by the user.

Some abends can provoke a crash of the address space in which it occurs. If the TOM or the ANM address space crashes, all other address spaces started by Connect:Express are stopped. If the APM or the AFM address space crashes, Connect:Express continues working without this resource. Some abends only provoke a transfer interruption. For example, an ABENDD37 means that the file received is full and the transfer was interrupted.

If a dump occurs in some of the Connect:Express address spaces or in a user-application, do the following:

- Check the abend code to determine if it is a product problem.
- Note the address space name, module name, and the displacement or offset of the abend and then contact the support team.

Note: All procedures provided with installation are included in the SYSDUMP DD cards. SYSUDUMP DD cards should not be deactivated in any of the Connect:Express address spaces. If an abnormal abend occurs, the development team can analyze the problem. The SYSUDUMP ABEND dumps contain data and areas about the failing program. SYSMDUMP ABEND dumps contain additional system areas and SYSABEND ABEND dumps are the largest of the ABEND dumps and contain more system areas.

CICS Errors

Some errors can occur when using the CICS option of Connect:Express. Most of these are caused because the transaction stop sequence was not respected, or Connect:Express or CICS abended before a normal stop.

- Type the following OS/390 command: « /F monitor, $LOAD$ » then stop the monitor and CICS, and restart both normally.
- Check the T3B2ZSSN table used, and //DFHRPL and //STEPLIB.
- Check the application using TSO/ISPF 2.5.

Refer to the CICS documentation for more information.

Enhancing System Performance

You can enhance file transfer performance using protocol facilities like compression. You must, however, consider some network and system parameters before optimizing the process.

Connect:Express OS/390 ships with default information needed for data transfer. You may need to customize this for your environment to assure good performance across different link types. To tune Connect:Express, the administrator should be familiar with VTAM, NCP, or X25NPSI parameters. It is also helpful to have a general understanding of the effects that some of the parameters can have on performance. Be careful when tuning Connect:Express. What works well for one file may not work well for another. Monitoring Connect:Express activity is an ongoing process. You can adjust parameters like network and file buffer length and address space storage to enhance system performance.

Network Buffer Parameters

Network buffers are defined on three levels: the transfer protocol message length, the ANM buffers, and the VTAM and TCP/IP buffers.

The APM sends data to the ANM using transfer protocol messages, and then the ANM sends data to the network. The ANM receives data from the network and then sends it to the APM. The length of exchanged messages can determine link throughput.

The length of transfer protocol messages is defined in Connect:Express session tables. The standard ANM buffers (STDMSG) and extended ANM buffers (MAXEXT) in kilobytes are defined in the Connect:Express SYSIN file. VTAM and TCP/IP buffer exchange capabilities are defined in the appropriate configuration files. All of these parameters must be related.

The length of transfer protocol messages is limited by the Connect:Express session table Message parameter. You must customize Connect:Express session tables based on your network environment. For example, you can specify the Connect:Express session table parameters depending on the link type you are using. Communications between Partners transmitting data is a negotiated process. Values such as the maximum message length, synchronization values, compression, and segmentation determine the data transfer performance within Connect:Express. As the Connect:Express administrator, you should be aware that the following rules apply during transfer negotiation:

- The smaller of two differing parameter values is the one that will be used.
- You should not modify the default values for Session Protocol tables, but you can create your own tables.

Note: When you want to modify session parameters, only modify the modules described as “Available for Customization.” Start with the last available entry, then copy the customized tables into your USERLOADLIB (it must be APF) so that no procedure update can replace a user module. Your USER LOADLIB must be concatenated before the procedure LOADLIB.

The protocol message must be transported through the ANM, in its STDMSG or MAXEXT buffers to the network. Over an X25 link, the Switch Major node MAXDATA parameter and those defining VTAM-NCP data exchange capability are considered. Refer to Appendix E, *Definition of VTAM Resources* for a list of general information and examples.

Data Set Blocking Factors

Connect:Express uses IBM algorithms to efficiently block data sets when doing file allocation, unless the file is managed by SMS. With appropriate data set blocking factors, you can increase throughput. You should enable Connect:Express to optimize data set blocking whenever possible.

System Performance Parameters

Connect:Express uses multiple address spaces to give you greater flexibility to achieve performance goals. Managing performance groups is very important to help you take advantage of this design.

The different address spaces use system resources according to their role. For example, an APM or an EAS may have to wait for disk I/O operations and consumes a lot of system resources, while the ANM must respond quickly to a network solicitation, but consumes very few system resources. The Monitor and the FTP manager must handle TSO/BATCH or internal requests and consume some system resources, like an interactive process. Each address space (ASCB) can be tuned depending on its specific function like ANM=NETWORK, TOM and AFM =INTERACTIVE, or APM and EAS =BATCH. It is best to keep these resources in different address spaces so that each resource can have its own priority. This enhances system performance for Connect:Express as well as for the OS/390 environment.

It is mandatory that the ANM address space executes in a performance group just below VTAM. The APM and EAS address spaces must execute in a performance group like a JOB batch. To enhance performance, you should ensure that the TOM Monitor and the FTP manager execute in a performance group like a TSO address space.

Storage Estimates

This section describes the storage requirements for different Connect:Express facilities.

Connect:Express CSA Storage Estimates

The CSA storage depends on SYSIN parameters such as APLNUM= , MAXSRQ, and the maximum number of APMs and effectors. The table below lists the CSA storage estimates.

Table Name	Size	Acquired At
T1B2PSCE	7552 bytes	$LOAD$ – Initialization
T1B2ZAPL	20 + (16*APLNUM) †	$LOAD$ – Initialization
T1B2ZAEX	(20 + (1034*APLAEX))*APLNUM †	$LOAD$ – Initialization
CSA PGM	35112 bytes	$LOAD$ – Initialization
T1B2PSVT	400 bytes	**Connect:Express START**
GM2PG40	20*[(NBEFF*NBAPM)+(2*MAXSRQ)]†	**Connect:Express START**
T1B2PRCB	528 bytes	IEFSSREQ issued
T1B2PRC2	4096 bytes	IEFSSREQ issued
T1B2PRC3	340 bytes	IEFSSREQ issued
T1B2PRC5	408 bytes	IEFSSREQ issued
T1B2PRC6	956 bytes	IEFSSREQ issued

Connect:Express Private Area Storage Estimates

The internal tables of Connect:Express are built from the SYSIN parameters RQEMAX=, PCTADD, FCTADD, and the number of partners and files defined in the directories. The following table lists the private area storage estimates.

Table Name	Size (Bytes)
T1B2PRCT	16 + (64*RQEMAX) †
T1B2PTCT	16 + (480*RQEMAX) †
T1B2PSST	16 + (1956*NBEFF) †
T1B2PPCT	16 + (216*(NBPAR+PCTADD)) †
T1B2PACT	16 + (520*(current number+PCTADD+1)) †
T1B2PFCT	16 + (216*(NBFIL+FCTADD)) †
T1B2PXCT	(600*PlexServ) †
T1B2PAPL	(20 + (16*APLNUM))* PlexServ) †

† NBEFF = Number of APM effectors
NBPAR = Number of Partners definitions in the directory
NBFIL = Number of file definitions in the directory
APLNUM, APLAEX, MAXSRQ, RQEMAX from the SYSIN file
PlexServ = Number of Plex Servers authorized (Asset Protection File)

Table Name	Size (Bytes)
PARTNER LIST	16 + (12*NBENTRY)
SYSX25	16 + (76*NBENTRY)
SYSSNA	16 + (52*NBENTRY)
SYSTCP	16 + (76*NBENTRY)
T1B2PCNT	user exits storage (CNX)

† NBEFF = Number of APM effectors
NBPAR = Number of Partners definitions in the directory
NBFIL = Number of file definitions in the directory
APLNUM, APLAEX, MAXSRQ, RQEMAX from the SYSIN file
PlexServ = Number of Plex Servers authorized (Asset Protection File)

ANM Private Area Storage Estimates

The ANM internal file depends on the maximum number of network sessions. It is also computed from the SYSIN parameters, STDMSG and MAXEXT. The number of sessions is equal to the total number of effectors, NBEFF. They correspond to the maximum number of parallel transfers. The table below lists the ANM private area storage estimates.

Area Description	Size
GENERAL USE:	
Supervisor + control blocks	20 kilobytes + (NBEFF*652 bytes)
Standard buffers	2*NBEFF*STDMSG kilobytes
Extended buffers	2*MAXEXTnb*MAXEXTln kilobytes
Welcome routine	4 kilobytes + (NBEFF*884 bytes)
HANDLERS:	
SNA AP-AP	5 kilobytes + (NBEFF*412 bytes)
X.25	12 kilobytes + (NBEFF*1024 bytes)
3270	6 kilobytes + (NBEFF*660 bytes)
LU6.2	6 kilobytes + (NBEFF*1040 bytes)
TCP/IP	6 kilobytes + (NBEFF*1024 bytes)

APM Private Area Storage Estimates

The APM internal size depends on the number of effectors allocated with the DAPMXX= parameter in the SYSIN file. The total size is equal to the size of the APM supervisor plus the size of one effector multiplied by the number of effectors. The following table lists the APM private storage area estimates.

Area Description	**Size**
GENERAL USE	
APM supervisor (attaches 1 to 16 effectors)	60 kilobytes + user application server storage (T1APMSRT)
EFFECTOR driver (attaches 1 protocol module)	20 kilobytes + user exits storage (PEX)
TEMPORARY USE: (PROTOCOL attached)	
PROTOCOL module	20 to 80 kilobytes (loads transfer areas)
TRANSFER AREAS: TRANSMISSION	BLKS+(2*LRECL) + SYNCHRO + 2*MSGLG + user exits storage (IEX,TEX)
TRANSFER AREAS: RECEPTION	BLKS+(2*LRECL) + 2*SYNCHRO + MSGLG + user exits storage (IEX,TEX)

SYNCHRO, in kilobytes, is the synchronization parameter from the T1B2PSxy session table. MSGLG is the network buffer length parameter from the T1B2PSxy session table in kilobytes. For example, you could have the following situation:

- APM storage for 8 effectors, 8 simultaneous files receiving.
- Session table is T1B2PS52, link=SNA (SYNCHRO=64, MSGLG=4).
- File (LRECL=240, BLKSIZE=24000).
- No user exit executed.

Storage = APM + 8*(EFFECTOR) + 8*(PROTOCOL + RECEPTION)

= 60 + = 8*20 + = 8*(80 + (2*synchro=128 + msglg=4 + blk+lrec=24))

= 2.5 Mbytes.

For example, you should declare only 8 effectors (not 16) with large user exits in COBOL. AFM storage estimates are discussed in the FTP Guide.

Communicating with Technical Support

If you encounter a problem with Connect:Express, it could be related to one of the following issues:

- Existing Connect:Express fix
- Installation or configuration problem
- System environment problem
- Usability question
- User problem
- Documentation issue

If you are unable to diagnose a problem, you can contact technical support. This section will help you analyze the problem and let you know what kind of information that you will need when you contact technical support.

Analyzing a Problem

The questions below will help you analyze a problem in Connect:Express.

Describe the system context: Where does the problem occur?

What is the local Connect:Express release + maintenance (PTF number, dates)? Note the initialization messages shown below:

```
TOM3016I TOM INITIALIZATION COMPLETE V4200-PTF0-0
TOM3264I TOM LAST UPDATE DATE-TIME : 061013-1047
```

What is the local OS and Network configuration and releases?

Describe the problem context: When does the problem occur?

During Connect:Express installation? During Connect:Express configuration? During Connect:Express operations? Is it a general Connect:Express process? Is it a Network connection error? Is it a File transfer error? Is the local Connect:Express calling or called? Is the local Connect:Express sender of the file or receiver of the file?

Describe the problem type: How does the problem occur?

Is it an OS/390 system error (for example: allocation, security)? Is it a Connect:Express system error (for example: abend, loop)? Does the problem occur while using an interface? Is it an operator interface error? Is it a batch interface error? Is it a user command interface error? Is it a user exit interface error? Does the problem occur while using a protocol? File transfer protocols are PeSIT D, PeSIT E, ETEBAC 3, ODETTE-FTP, and FTP. Network protocols are TCP, SNA, and X25.

Describe the symptoms: What happens?

Is the action rejected, interrupted, stopped, abended, frozen? Is it a Local or remote error? (TRC=0 indicates a remote error.) Note the local and remote error codes: TRC, PRC, SRC, NRC, and user interface RC. Note the local and remote Messages from Connect:Express and system. Did you change anything before the problem occurred for the first time? What is the number of occurrences? Is it reproducible? Is there any significant pattern that produces the problem? What is your perception of the severity? Is there an alternate solution?

Gathering Information for Technical Support

The following table lists all the information that you may need when you talk to technical support. Depending on the problem, you will need to provide some of the files listed below.

File or Type	Description
Configuration files	Connect:Express parameters, network parameters, system parameters, job streams.
Transfer parameters	File/partner local definitions, partner software and machine. Provide hardcopies of TSO/ISPF definitions.
TOM SYSLOG file	Tracks chronological events. From basic information such as file name, partner name, date time, request number or anything else. You can track all the events related for example to a specific data flow, a specific link or a recurrent return code.
TOM SYSMSG file	Tracks operational events such as initialization and connection user exit messages.
APM SYSLOG file	Tracks chronological events. Information are found about file allocation.
APM SYSMSG file	Tracks operational events such as initialization.
APM exit SYSPRINT files	Tracks protocol parameters.
ANM SYSMSG file	Tracks operational events such as initialization or exceptional diagnostics.
Any utilities and the API	Provides a SYSPRINT file in which you can find all needed information. Refer to the appropriate pages for example, P1B2PRQ2, L0B2Z20.
JOURNAL file	Achieves transfer reports. You can use the filter screen to select what may be related to your current problem. Provide hardcopies of TSO/ISPF view.
Request Control Table (RCT)	Shows current status of transfers. Provide hardcopies of TSO/ISPF view.

Abend files and trace files always need to be related with SYSLOG and SYSMSG files.

- If an abend occurs, send it to support.
- If you get traces of the line or the buffer type, send them to support.

Note: To make a file from a sysout, under IBM SDSF use the ‘PRT ODSN ….’ and ‘PRT’ commands.

Performing a Multi-Address Space Dump

Some situations may require a general image of all Connect:Express address spaces running at the same time. To do this, type the OS/390 command: «F monitor, DUMP» to provide a SYS1.DUMP file without terminating Connect:Express. The dump will contain all TOM, ANM, APM running address space images. Use a tape to send the resulting SYS1.DUMP to support.

Performing an External Address Space Dump

Some situations may require an image of an address space running with Connect:Express such as APM-BSC, AFM, EAS or user-application (CICS, IMS, TSO, BATCH …). To do this, type the OS/390 command: «DUMP COMM=….» and specify the necessary OS/390 parameters. Use a tape to send the resulting SYS1.DUMP to support.

Sending Information to Support

When you contact technical support, you will receive a case number. You will need to refer to this number when you send information to support. Follow the guidelines listed below.

1. Using a tape for a SYS1.DUMP.
   - Put the files on a 3480/3490 18/36 tracks cartridge VOL=SER= “what-you-want and be case-id”
   - The DSN begins by “PSR$TST.case-id. ? ? ?”
   - The JOB printed SYSOUT must be included with the cartridge.
2. Using the Internet.
   - Send an email with material attached to:

     connect_eurosupp@stercomm.com

Appendix A

Connect:Express User Libraries

This appendix describes the Connect:Express user libraries which provide examples to help you customize and automate Connect:Express.

User Libraries

The following table lists and describes the Connect:Express user libraries. Each library is discussed in detail in the sections that follow.

Library	Description
C:X DISTLIB	Provides distribution JCL examples for the installation tape.
C:X INSTLIB	Provides sample installations.
C:X SAMPLIB	Provides examples of user exits and batch processes, as well as assembly and COBOL.
C:X SAMPOPT	Provides examples of optional features such as the Utilities package.
C:X MACLIB	Provides DSECTs that you can use in any applicable module or exit, and compile and link for any product upgrade.
C:X PARMLIB	Provides examples of edit files for parameter extensions, such as SYSX25 and SYSSNA for alternate address parameters.
C:X SYSJCL	Provides JCL skeletons that Connect:Express uses to run jobs like Unload/Reload.
C:X SYSPRM	Provides examples of edit files for parameter extensions, including selection members for PDS UNLOAD PROCs.

C:X DISTLIB

This library contains edit files that you can use for your first installation of Connect:Express. This library includes the following JCL:

- Tape creation
- SMP
- Allocation
- OZS data base reload
- PTF notification £GFIMSG

C:X INSTLIB

This library contains edit files that you can use to install and customize Connect:Express, and includes the following installation material:

- ISPF customization
- VSAM directory definitions
- SIT directories samples
- Directories migration
- Stage 1 parameters
- MONITOR/ANM/APM/AFM/EAS/ATM procedures
- VTAM samples
- Security recommendations

C:X SAMPLIB

This library contains examples that you can use to automate general procedures like user exits and transfer requests. The following screen shows part of the contents of the $$$$SAMP file with the module name syntax and a brief description.

```
Transfer request                                $REQ...
MONITOR checkpoint OFFLOAD/RELOAD               $CKOFLOD
P0B2Z20 purge requests                          $P0B2Z20
MONITOR recovery OFFLOAD/RELOAD                 $RCOFLOD
TABLES                                          TB£..
ACS: Batch create of table                      $TB£ACS,TB£ACS
CLIST: End of transfer                          C1USR...
COBOL DSECTS, JCL and exits                     D1COB...,EX£C...
APM JCL customization start cmd                 EX£PJCL
JOURNAL: query sample pgm                       EX£UJNL
JOURNAL: Check SYSJNL                           EX£992
JOURNAL: user exit                              EX£msg1
JOURNAL: L1B2PDIX dispatcher                    EX£DIX
L0B2ZAPL connection to MONITOR                  EX£APL
APM exits pgm                                   EX£APM...
L0B2Z20 calling program                         EX£Z20
L0B2Z20 monitoring program                      EX£Z20S
C:X  Batch stop                                 EX£ZEOD
C:X  Batch start proc UPRCPI                    EX£ZIPL
User exit: Connection access control M1USRCNA L1USRCNA
User exit: Begin of transfer                    EX£A...
user exit: End of transfer                      EX£A...
User exit: Middle of transfer EBCDIC/ASCII      EX£A...
User exit: Transfer request control             EX£A
ACS: Batch use pgm                              L0USRZA,L0USRACS
User exit: Connection access control            L1USRCNA
User exit: Sending messages to SYSLOG           L1USRLOG
RACF security user exit                         L1USRTAC
 C:X  transfer request control user exit        L1USR006
 C:X  reload transfer request pgm               L1USR801
 C:X  DUMP procedures                           TOMDUMP
```

C:X SAMPOPT

This library contains examples that you can use to automate procedures using optional features like Connect:Express Utilites and CICS. The following table lists part of the contents of the $$$$SAOP file with the module name syntax and a brief description.

Module Name Syntax	Description
L1GFI..., P1GFI...	Includes examples of parameters for these Connect:Express Utilities option programs.
L1USR...	Includes examples of user exits for these Connect:Express Utilities option programs.
P3...	Includes examples of the Connect:Express CICS option program.

C:X MACLIB

The MACLIB contains DSECTS that can be used in user programs or exits. The table below lists the module name syntax in the MACLIB and the program in which they can be used.

Module Name Syntax	Description
D0B2Z...	Assembler DSECTS for the API L0B2Z0/L1B2P20
D0COB...	COBOL COPYS for the API L0B2Z20/L1B2P20
D1B2P..., D1B2R...,	Assembler DSECTS for the user exit interface
D1COB...	COBOL COPYS for the user exit interface
D1GFI...	Assembler DSECTS for the Utilities option
D3B2...	Assembler DSECTS for the CICS option
M0...,M1...	Assembler DSECTS for creating C:X TABLES
M3..	Assembler DSECTS for creating macros with the C:X CICS-OPTION
M1USRCNA	Assembler DSECT for customizing the RACF connection exit

Note: The reference DSECT is always in ASSEMBLER. If it is not furnished in COBOL, copy the ASSEMBLER DSECT and use the 'M1COBASM' ISPF EDIT MACRO to create the COBOL copy structure.

C:X PARMLIB

This library contains edit files used to create Connect:Express parameters extensions such as alternate addresses, planned transfers, and Partner lists. After you modify these files, they must be refreshed. Examples of initialization parameters such as SYSIN and CXPLEX files are also provided. Some of the files included in

the *PARMLIB* are listed below with the DDNAMES by which they are referenced in the JCL cards of the Connect:Express address spaces.

Module	Description
£EVENT	Planned transfers. This file is referenced by the SYSEVT DD card in the TOM JCL.
£LIST	Partners list. This file must be in a library referenced by the SYSPRM DD card in the TOM JCL.
£MANAGER	Manager CXPLEX file. This file is referenced by the CEXPLEX DD card in the TOM JCL.
£PRMETB3	ETEBAC3 card. This file is referenced by the PARMETB3 DD card in the APM JCL.
£PRMFTPL	FTP list customization. This file is referenced by the PARMFTPL DD card in the EAS JCL.
£SERVER	Server CXPLEX file. This file is referenced by the CEXPLEX DD card in the TOM JCL.
£SYSX25	X25 alternate addresses. This file is referenced by the SYSX25 DD card in the TOM JCL.
£SYSSNA	SNA alternate addresses. This file is referenced by the SYSSNA DD card in the TOM JCL.
£SYSTCP	TCP/IP alternate addresses. This file is referenced by the SYSTCP DD card in the TOM JCL.
£SYSEXT	L1B2PDIX exit list. This file is referenced by the SYSINEXT DD card in the TOM JCL.
£SYSUE1	L1GFIUE1 parameters. This file is referenced by the SYSUE1 DD card in the APM and EAS JCL.

C:X SYSJCL

This library contains JCL skeletons used by Connect:Express. After you modify these files, they must be refreshed. This library is referenced by the SYSJCL DD card in the TOM JCL. The files included in the *SYSJCL* are listed below with a short description.

File	Description
$JOBREL1	Job card for reloading a PDS data set. This file must only be customized once.
$RELOAD1	Execution parameters for reloading a PDS data set. Connect:Express uses this file as a model to build the actual job.
$JOBUNL1	Job card for unloading a PDS data set. This file must only be customized once.
$UNLOAD1	Execution parameters for unloading a PDS data set. Connect:Express uses this file as a model to build the actual job.
$JOBREL2	Job card for reloading a VSAM data set. This file must only be customized once.
$RELOAD2	Execution parameters for reloading a VSAM data set. Connect:Express uses this file as a model to build the actual job.
$JOBUNL2	Job card for unloading a VSAM data set. This file must only be customized once.
$UNLOAD2	Execution parameters for unloading a VSAM data set. Connect:Express uses this file as a model to build the actual job.
$JOBREL3	Job card for reloading a USER data set. This file must only be customized once.
$RELOAD3	Execution parameters for reloading a USER data set. Connect:Express uses this file as a model to build the actual job.
$JOBUNL3	Job card for unloading a USER data set. This file must only be customized once.

File	Description
$UNLOAD3	Execution parameters for unloading a USER data set. Connect:Express uses this file as a model to build the actual job.
$JOBREL4	Job card for reloading a SYSOUT data set. This file must only be customized once.
$RELOAD4	Execution parameters for reloading a SYSOUT data set. Connect:Express uses this file as a model to build the actual job.
$JOBUNL4	Job card for unloading a SYSOUT data set. This file must only be customized once.
$UNLOAD4	Execution parameters for unloading a SYSOUT data set. Connect:Express uses this file as a model to build the actual job.

C:X SYSPRM

This library contains edit files that you can associate with the JCL Skeleton from the *SYSJCL* library. It also includes examples for selecting members for PDS UNLOAD and DELETE/DEFINE for VSAM RELOAD. This library is referenced by the SYSPRM DD card in the TOM JCL.

Appendix B

Error Codes and Messages

This appendix describes error codes and messages from the Connect:Express application.

Error Codes

The error codes section lists and explains the types of return codes that you may encounter with Connect:Express. Return codes identify errors and can come from the network, system, Connect:Express, or the transfer protocol you are using. You can then correct errors based on their origin. The sections below list system return codes (SRC), network return codes (NRC), Connect:Express return codes (TRC), and protocol return codes (PRC).

Note: If an error is not listed, press the Help key <PF1> two times when you are on a return code screen. This will display information about the origin of the return code. From the TSO/ISPF screens in Connect:Express, you can also type: 'TRC tttt', or 'PRC ppp' for direct Help or 'RC' for general 'C:X RETURN CODES' Help.

System Return Codes (SRC)

System return codes are returned by OS/390 functions during any process running under its control (DYNALLOC/DASDM/...). In this case, the error is detected by the system.

System return codes will not display alone. You will always see a Connect:Express return code (TRC). If you receive a TRC and think the error may be a system error, display the help for the TRC to see if there is an SRC associated with it. Refer to IBM documentation for help with system errors.

Network Return Codes (NRC)

The ANM return codes format is AAxxyy and BBxxyy. The xxyy values change depending on the connection type. Check for special WTO messages in the ANM ASID. The NRC provided by the C:X GLOBAL/LOCAL interface has only one CCxxyy component.

NRC Codes From the ANM

NRC codes from the ANM have the format AAxxyy BBxxyy. The first part of the code gives the result of the action, and the second part gives the reason why the action failed. AA or BB identifies the event or action

concerned and the xx and yy variables depend on the current link. The table below identifies the xx and yy return codes for each link type.

Link	Description of xx and yy Return Codes from NRC.
SNA LU0, LU2	VTAM return code and FEEDBACK2 (RPLRTNCD, RPLFDBK2)
SNA LU6.2	APPC primary and secondary return codes APPC RPLGRCPR and RPLGRCSC 2nd bytes (RCPRI, RCSEC)
X.25	CAUSE and DIAGNOSTIC
TCP-IP	TCP-IP return codes (errno-1 and errno-2)

Note: If xx = X'FF,' the error is from an internal component of a local or remote ANM.

'AA' describes an error detected by the ANM when executing a service request from the TOM or APM address space. It is called the Action field. 'BB' describes an error detected from the Network by the ANM. It is called the Event field. The table below lists the possible values of AA and BB.

Code		Description
AA	= 00	xxyy is the return code of TOM/APM/ANM interface
		xxyy = 0004 APM/ANM buffers length incompatibility
		xxyy = 0008 ANM service asked for session already terminated
		xxyy = 000C ANM service asked for inactive handler
	= 01	Function failed
	= 02	LOSTERM detected
	= 04	TPEND detected
	= 10	RESTART X.25 command received
	= 80	Sequence error
BB	= 04	Receive data error
	= 08	DISCONNECT detected
	= 40	TIME-OUT detected
	= 80	CLOSE detected

When xx=FF, the FFyy return codes are generally issued only by X.25 handler. The following table describes the possible values for yy codes in this case.

Code		Description
yy	= 01	All SVCs are already in use.
	= 02	No MCH available (busy or inactive).

Code	Description
= 04	SVC was successfully opened but SMN level session establishment failed (SMN inactive or IDNUM error).
= 07	ANM CTCP session with MCH LU lost.
= 09	Unexpected packet with Q-bit on received.
= 31	DATE MCH in use. Time-out on call confirm detected.
= FF	Any involved handler **abended**.

Global/Local NRC Codes

In global/local sessions, a global Connect:Express system on one host communicates with a local Connect:Express system (a subset of the monitor) on another host through an open VTAM session. These codes are an optional feature with Connect:Express OS/390, and are listed below.

Code	Description	
CC = 80	UNBIND	received
= 81	TPEND	detected
= 82	DFASY	detected
= 83	RECEIVE	rejected
= 84	RECEIVE	error
= 85	LOSTERM	detected
= 86	OPNSEC	error
= 87	GENACB	error
= 88	OPEN ACB	error
= 89	GENRPL	error
= 8A	OPNDST	error
= 8B	SEND	error
= 8C	GETMAIN	rejected
= 8D	Termination requested by operator	

Connect:Express Return Codes (TRC)

Each time an error is detected locally by Connect:Express, a TRC is issued. Return codes from Connect:Express have 4 characters and are classified into eleven groups. These codes are listed in the sections that follow. TRC codes followed by the character R refer to a remote C:X Partner, and TRC codes followed by the character L are from a local C:X system.

WARNING: The TRC field can be set for a remote TRC. This means that there will be NO local errors, only remote errors. The remote Partner must be a Connect:Express monitor, otherwise when the TRC equals 0000, the error was detected on the remote side.

The example below shows how the remote TRC is processed internally.

```
*----- TRC IF REMOTE  C:X  FLAGGED BY : BYTE0 BIT1 SET TO ZERO,
* TO TEST BY             : TM   ..TRC,X'40' remote trc type?
*                          BNO  ...         no
* TO MAKE IT EDITABLE BY : OI   ..TRC,X'40' editable now
See *MACLIB* D1B2RUEX D1B2PJNL D0B2ZEXR.
```

The low order bytes indicate the type of TRC. The following programs have to consider this detection method if the TRC is local or remote:

- User exit
- Journal inquiry utility, read function
- Request control table scanning utility

The table below shows the ranges and groupings of Connect:Express return codes (TRC).

Code Range	Description
1000 – 1999	Error detected by the protocol during protocol controls
2000 – 2999	Error detected by Connect:Express during user action controls
3000 – 3999	Error detected by the protocol during file management
4000 – 4999	Error detected by the protocol from a user exit
5000 – 5999	Error detected by the protocol during any negotiation process
6000 – 6999	Error detected by the protocol during storage acquisition
7000 – 7999	Error detected by the protocol during checkpoint management
8000 – 8999	Error detected by Connect:Express after protocol "ABEND"
9000 – 9999	Error detected by the protocol after network error
A000 – A999	Error detected by ANM (Handlers)
B000 – B999	Error detected by APM (BSC)
F000 – F999	Error detected by TOM Plex Supervisor (Fxxx is equivalent to 2xxx)

TRC Return Codes from Protocols (1 xxx)

Refer to the protocol description in the Administration Guide for a list of possible commands and parameters.

TRC Return Codes from Protocols, 1 xxx

Return Code	Description
1 000	LOGON rejected by Partner
1 00x	Invalid LOGON received – invalid X field
1 1XX	Invalid length for command XX
1 2YY	Unknown or absent (PRC=318) or forbidden (PRC=311) YY parameter

TRC Return Codes from Protocols, 1 xxx

Return Code	Description
1 3XX	Invalid structure for command XX
1 4YY	Parameter YY not in the right place
1 5YY	Invalid value in parameter YY
1 6XX	Invalid command XX
1 7XX	Invalid header for command XX
1 8YY	Invalid length for parameter YY

TRC Return Codes from Transfer Operations (2xxx)

TRC Return Codes from Transfer Operations, 2xxx

Return Code	Description
2001	Connect:Express is not active
2002	Function not supported (INIT/TERM in process...)
2003	Logical error in subsystem request
2004	Fatal error in subsystem request
2005	Invalid post number
2006	Invalid user type
2007	Invalid user identity
2008	Invalid request type
2009	Invalid operation type
2010	Invalid File name (not found in directory)
2011	Invalid Partner name (not found in directory)
2012	Invalid direction
2013	Invalid DSNAME
2014	Invalid member
2015	Invalid request number
2016	Transfer table is full
2017	Request table is full
2018	Invalid allocation rule
2019	Allocation error PHASE 1
2020	Allocation error PHASE 2
2021	Invalid TQE chain
2022	Request table is empty

TRC Return Codes from Transfer Operations, 2xxx (continued)

Return Code	Description
2023	Active or selected request
2024	Reserved
2025	Request refused; Connect:Express inactive and RACF control mandatory
2026	Request refused; disabled Partner on remote location
2027	Request refused; disabled file on remote location
2028	Command refused; user not authorized for this request
2029	Request active; try later
2030	Selected Partner or file – PCT/FCT update failed
2031	Partner or file not found – PCT/FCT update failed
2032	Duplicate Partner or file – PCT/FCT add failed
2033	Partner or file cannot be added (PCT is full) – PCT/FCT add failed
2034	Partner or file is not allowed for dynamic update
2035	More than one file requested
2036	User/dsn access denied by RACF (see LOG and SRC)
2037	Invalid SIT data direction (French SIT network)
2038	Invalid SIT application identity (French SIT network)
2039	Invalid Request (French SIT network)
2040	Invalid RECFM
2041	Invalid LRECL
2042	Request not found
2043	Restart impossible
2044	Network circuit lost
2045	Network time out
2046	Inactive request
2047	Request not interrupted
2048	Reserved
2049	Reserved
2050	Reserved
2051	Invalid origin
2052	Reserved
2053	Invalid destination
2054	No transfer in queue (HOLD)
2055	Invalid direction for inquiry selection

TRC Return Codes from Transfer Operations, 2xxx (continued)

Return Code	Description
2056	Invalid direction for queued request (HOLD)
2057	Invalid state for this request
2058	Invalid priority for this request
2059	Major resource disabled
2060	Partner disabled
2061	File disabled
2062	Held request not found in Partner's queue
2063	Session table full
2064	Recovery inactive
2065	Maximum for active requests is reached (MAXSRQ)
2066	No more space available in CSA
2067	XPOST error
2068	Insufficient authority (APF)
2069	Maximum for allocation is reached
2070	Re-load failed
2071	Invalid protocol number
2072	Invalid data set organization
2073	Transfer failure – RJE/ETEBAC transfer
2074	Partially accepted
2075	Totally rejected
2076	Communication with Partner not obtained
2077	Communication rejected by Partner
2078	Invalid link type requested
2079	Reserved
2080	Allocation or DADSM error, see SRC on help screens
2081	Invalid or unknown VSAM file
2082	Unknown file
2083	File error (DSCB4) – internal error
2084	File error (DSCB1) – internal error
2085	Data set duplicate name
2086	Not enough space available on volume(s)
2087	ADY PARMLIST invalid (internal error) see SRC
2088	Incompatible DSORG

TRC Return Codes from Transfer Operations, 2xxx (continued)

Return Code	Description
2089	Allocation error (MSS)
2090	Allocation error (DADSM) see SRC with continued helps
2091	Allocation error (CATALOG)
2092	Other allocation error see SRC with continued helps
2093	Delete data set error
2094	Uncataloged error
2095	Scratch data set error
2096	Multivolume processing error, see SRC with continued helps
2097	PDS processing error, see SRC with continued helps
2098	Connect:Express /dsn access denied by RACF, see LOG and SRC
2099	Reserved
2100	Reserved
2101	Invalid parameter FCT
2102	Update impossible if alternate Partner
2103	Invalid X25 charge parameter
2104	Invalid select function for protocol
2105	Network receive error
2106	Incompatible levels of C:X -SPF and Connect:Express subsystem
2107	Reserved
2108	Reserved
2109	Reserved
2110	Reserved
2111	Reserved
2112	Reserved
2113	Message/RUSIZE too big (GLOBAL/LOCAL session)
2114	VTAM error (GLOBAL/LOCAL session)
2115	"BSC" APM out-of-service
2116	Request rejected by user exit
2117	Invalid first msg returned to ANM
2118	Invalid disp= for ADHOC
2119	RACF control failed for adhoc
2120	Full GM2 for L1B2P048
2121	Different BLOCKSIZE (RECFM=U)

TRC Return Codes from Transfer Operations, 2xxx (continued)

Return Code	Description
2122	Refused PARTNER list (#LIST: L1B2P049 MAX GM2 reached)
2123	Empty PARTNER LIST (#LIST: L1B2P049)
2124	SLD entry not found or invalid
2125	Invalid alternate PARTNER
2126	BLOCKED REQUEST invalid with UNLOAD/RELOAD
2127	Entry in error (NOT USABLE), check in C:X SYSMSG
2128	DELETE forbidden for this profile (PARTNER/FILE)
2129	Reserved
2130	Reserved

TRC Resources Return Codes (2131 - 2169)

TRC Resources Return Codes, 2131 - 2169

Return Code†	Description
2131	Rejected because Connect:Express is disabled.
2132	Rejected because ANM is disabled.
2133	Rejected because RCT is disabled.
2134	Rejected because PCT is disabled.
2135	Rejected because FCT is disabled.
2136	Rejected because SST is full.
2137	Rejected because NO APM available.
2138	Rejected because NO EFFECTOR available.
2139	Rejected because NO APM CLASS available.
2140	Rejected because PARTNER is disabled.
2141	Rejected because FILE is disabled.
2142	Rejected because PARTNER is not FOUND. (CHECK PROTOCOL)
2143	Rejected because FILE is not FOUND.
2144	Rejected because MAXIMUM PC reached.
2145	Rejected because MAXIMUM LINK REACHED.
2146	Rejected because PRIORITY/LINK error in SLD, or MAX EFFECTOR reached.
2147	Rejected because LINK COUNT error (SLD).
2148	Rejected because SELECTED LINK-HANDLER(S) inactive.

† Check Connect:Express resource and enable/activate/change or wait until available.

TRC Resources Return Codes, 2131 - 2169 (continued)

Return Code†	Description
2149	Rejected because MCH/LINK-HANDLER inact or MCHMSC invalid.
2150	Invalid PROTOCOL.
2151	UNKNOWN NETWORK identification (DIRECTORY/SYS...).
2152	UNKNOWN NETWORK link-type (DIRECTORY/SYS...).
2153	Rejected because TAXATION RULE error.
2154	Rejected because invalid ETEBAC/PESIT MSG2.
2155	Rejected because invalid ETEBAC/PESIT PROF.
2156	Rejected because APPC unavailable.
2157	Rejected JCL UNLOAD/RELOAD (L1B2P043: CHECK *SYSJCL* AND 'REFRESH').
2158	Rejected JCL 'BFX' (L1B2P043: CHECK *SYSJCL* AND 'REFRESH').
2159	Rejected JCL 'RJE' (L1B2P043: CHECK *SYSJCL* AND 'REFRESH').
2160	JCL FUNCTION stopped (L1B2P043: CHECK *SYSJCL* AND 'REFRESH').
2161	Unknown IP address, incoming call.
2162	Unused.
2163	SSL handler is disabled.
2164	SSL is forbidden for this partner.
2165	SSL is required for this partner.
2166	Unused.
2167	Unused.
2168	Unused.
2169	Unused.

† Check Connect:Express resource and enable/activate/change or wait until available.

TRC Asset Protection Return Codes (2170 - 2199)

TRC Asset Protection Return Codes, 2170 - 2199

Return Code†	Description
2170	Option 01 not authorized Connect:Express
2171	Option 02 not authorized BSC-ETEBAC1/2
2172	Option 03 not authorized FTP
2173	Option 04 not authorized NSC-BFX
2175	Option 06 not authorized LU6.2
2176	Not used

† Check your Asset Protection file and call support for a new one. Authorized option are shown under ISPF '0.O'.

TRC Asset Protection Return Codes, 2170 - 2199 (continued)

Return Code†	Description
2177	Option 08 not authorized PAC
2178	Option 09 not authorized LOCAL
2179	Option 10 not authorized CICS
2180	Option 11 not authorized DIFFUSION
2181	Option 12 not authorized MAILBOX
2182	Option 13 not authorized
2183	Option 14 not authorized RJE
2184	Option 15 not authorized TCP-IP
2185	Option 16 not authorized IMS
2186	Option 17 not authorized REXX
2187	Option 18 not authorized
2188	Option 19 not authorized DES
2189	Option 20 not authorized SECURITY
2190	Option 21 not authorized
2191	Not used
2192	Option 23 not authorized
2193	Option 24 not authorized
2194	Not used
2195	Not used
2196	Not used
2197	Not used
2198	Not used
2199	Not used

† Check your Asset Protection file and call support for a new one. Authorized option are shown under ISPF '0.O'.

TRC ISPF/BATCH Return Codes (28xx)

Return Code†	Description
28XX	XX is the true ACS return code.
2801	Access refused to this PARTNER.
2802	Access refused to this FILE.
2803	Access refused to the REQUEST from this USER.
2805	Command or function forbidden (logged in OS/390).

† Check ACS access-security table (ISPF 3.T.ACS).

Return Code†	Description
2806	Access refused to this SUBSYSTEM (logged in OS/390).
2891	L0B2PACS error: Parameter field length invalid.
2892	L0B2PACS error: Function unknown.
2893	L0B2PACS error: End of table found.
2894	L0B2PACS error: Null parameter (or R1=0).
2896	L0B2PACS error: Invalid table level.
2897	L0B2PACS error: Invalid table name.

† Check ACS access-security table (ISPF 3.T.ACS).

TRC Return Codes from the L0B2Z20 Routine or Subfunctions (29xx)

Return Code	Description
2900	L0B2Z20 anomaly, no entry matched.
2903	L1B2ZAPL anomaly, timer expired (WAI=).
2907	L1B2ZAPL anomaly, no match.
29XX	IEFSSREQ error; XX = IEFSSREQ return code, R15=4:16.
2997	L1B2ZAPL error, internal.
2998	L0B2Z20 invalid parameters or DISPLAY entries left.
2999	L0B2Z20 internal error. Check EMSG zone or JCL '//SYPR20 DD SYSOUT=*' for a detailed message.

TRC Return Codes from File Processing, 3xxx

TRC Return Codes from File Processing, 3xxx

Return Code	Description
3 Z01	Allocation†
3 Z02	Deallocation†
3 Z03	Open input
3 Z04	Open output
3 Z05	Read error
3 Z06	Note error
3 Z07	Point error
3 Z08	Write error
3 Z09	Invalid record length

† Check the APM log for OS/390 DYNALLOC RBX WTO messages and for DYNALLOC error codes '0210'/'020C.' L1APMDYA will show who holds this data set.

TRC Return Codes from File Processing, 3xxx (continued)

Return Code	Description
3 Z10	DD SYSVOL absent
3 Z11	Record format not supported
3 Z12	Record length not supported
3 Z92	Invalid LRECL. The record length of the file received doesn't match the file definition.
3 Z93	Invalid BLKSIZE. The files block size is does not match the file definition.
3 Z94	Invalid RECFM. The format of the file received doesn't match the file definition.
Z=0	Normal transfer
Z=1	Restarted transfer
Z=2	Re-synchronization processing

† Check the APM log for OS/390 DYNALLOC RBX WTO messages and for DYNALLOC error codes '0210'/'020C.' L1APMDYA will show who holds this data set.

TRC Return Codes from User Exits (4xxx)

TRC Return Codes from User Exits, 4xxx

Code	Description
4 0RC	Error in beginning transfer user exit (Return Code=RC from user exit)
4 iRC	Error in transfer user exit (Return Code=RC from user exit) (i from 1 to 3)
4 4RC	Error in security interface Return Code=RC
4 5RC	Error in I/O user exit, Return Code=RC
4 6RC	Error in application server exit (connection, selection), Return code=RC from user
4 700	Error in loading begin transfer user exit
4 7ii	Error in loading T1B2PPii table (ii from 01 to 24)
4 797	Error in loading T1SECPss table
4 798	Error while loading ETEBAC3 exit
4 799	Error in loading end transfer user exit
4 8RC	Error in compression routine
4 814	Error in compression routine: null length segment
4 898	Fixed record length error (even if compression is not active)
4 899	Variable record length error (even if compression is not active)
4 9RC	Error in end transfer user exit (Return Code=RC 01:90 from user exit). See the Administration Guide for more information.
4 X91	Error in user exit. RC range must be from 1 to 90
X = 0	Transfer initialization exit

TRC Return Codes from User Exits, 4xxx (continued)

Code	Description
X = 1,2,3	Record processing exit
X = 4	Security
X = 6	Connection or selection exit
X = 9	Transfer termination exit

TRC Return Codes from Transfer Negotiation, 5xxx

Code	Description
5 001	Restart point – negotiation failure
5 002	Synchronization point – checkpoint error
5 003	Re-synchronization point – negotiation error
5 004	Too many bytes without a checkpoint
5 005	Null length for a record
5 006	Invalid number of bytes transferred – different from announced value
5 007	Invalid number of records transferred – different from announced value
5 008	More bytes received than announced
5 009	Error threshold reached
5 010	Unfilled record

TRC Return Codes from Storage, 6xxx

Code	Description
6 001	GETMAIN error
6 002	OPEN SYSLIB error
6 003	PARMETB3 file syntax error
6 iXX	Subsystem function IEFSSREQ error

TRC Return Codes from the Checkpoint, 7xxx

Code	Description
7 Z01	Allocation
7 Z02	Open error

Code	Description
7 Z03	Read error
Z=0	Normal transfer
Z=1	Restarted transfer
Z=2	Re-synchronization processing

TRC Return Codes from an ABEND, 8xxx

Code	Description
8 XYY	APM X, EFFECTOR YY ABENDED Associated SRC=U065 means that the timer has expired during user exit execution. Associated SRC=U064 means that the APM request to Connect:Express has been rejected.

TRC Return Codes from the Network, 9xxx

Code	Description
9 0XX	Null length message, command XX
9 1XX	CRC control (Asynchronous), command XX
9 2XX	Transfer error, sequence check failed, command XX
9 4XX	Last request failed
9 8XX	Abnormal state of ANM handler, command XX
9 999	Buffer extension demand rejected (check REGION=)

TRC Return Codes from the ANM

TRC Return Codes from the ANM

Return Code	Description
A XAS	ANM component abended (system)
A XAU	ANM component abended (user)
A X01	ANM DD card absent
A X02	ANM Invalid initialization parameters
A X03	ANM handler GETMAIN error
A X04	ANM ATTACH or LOAD error
A X05	Abnormal end for an ANM handler
A X10	ANM GENACB error

TRC Return Codes from the ANM (continued)

Return Code	Description
A X11	ANM OPEN ACB error
A X12	ANM GENRPL error
A X13	ANM SETLOGON error
A X20	Reserved
X	= Identity of ANM component
X=0	MANAGER (dispatcher)
X=1	Sub-system welcome routine
X=2	SNA LU 0 handler
X=3	SNA 3270 handler (SNA LU2 handler)
X=4	X25 CTCP (MCH handler)
X=5	X.25 protocol handler
X=6	PAD/PCNE handler
X=7	SSL Handler
X=8	reserved
X=9	LU6.2 handler
X=A	TCP-IP handler
B 101	Error line open
B 102	Error 'READ-TI' (BTAM)
B 103	'READ TI' ended and ENQ received (BTAM)
B 104	'READ TI' ended and READ error (BTAM)
B 105	Error 'READ-TT' (BTAM)
B 106	Error card, 'STX' missing
B 107	Error card,'STX''ETX' found

Protocol Return Codes (PRC)

Protocol return codes typically display with a TRC. Some PRC codes however, provide enough information to explain the problem and in this case, no TRC is issued. When a TRC is absent, this usually means that the error was detected on the remote side.

Some TRCs are sent by the issuer and display as TRC=xxxxR. You can return PRC codes from your own user exits. This section lists PeSIT and ODETTE-FTP return codes.

PeSIT Return Codes

This section lists all PeSIT return codes while the next section describes the relationship between PeSIT PRCs and TRCs. The PeSIT return code structure is x xnn. The codes appear in Connect:Express reports as a four character code where the first character means:

2 = transfer aborted
3 = session aborted

The following table lists the PeSIT return codes.

PeSIT Return Codes

Return Code	Description
1100	Transmission error
2200	Insufficient file attributes provided
2201	System resources temporarily unavailable
2202	User resources temporarily unavailable
2203	Low priority transfer
2204	Data set already exists
2205	Data set does not exist
2206	Reception of file would exceed the defined file size
2207	File busy
2208	File too old
2209	Not used
2210	Failure during presentation protocol negotiation
2211	Open file error
2212	Close file error
2213	Severe input/output error
2214	Failure during negotiation of restart point
2215	Local internal system error
2216	Voluntary premature stop
2217	Too many checkpoints without acknowledgements
2218	Error during transfer
2219	File space exhausted
2220	Record length longer than negotiated value
2221	Timeout waiting for end of transmission
2222	Too much data without a checkpoint
2223	Abnormal end of transfer
2224	File size greater than negotiated value
2225	PARTNER overloaded
2226	Transfer rejected
2299	Miscellaneous error (see TRC if not null)

PeSIT Return Codes (continued)

Return Code	Description
3300	Local system overflow (called PARTNER is locked and resource held – APM, PCT, FCT, RCT, or MAX.NO. OF CONNECTIONS already active)
3301	Invalid monitor ID (DPCSID)
3302	Called PARTNER is inactive
3303	Remote site network is overloaded
3304	PARTNER not authorized
3305	Negotiation failure – SELECT
3306	Negotiation failure – RESYNC
3307	Negotiation failure – SYNC
3308	Release level not supported
3309	Maximum number of connections already active
3310	Network error
3311	Protocol error
3312	Session closure requested by user
3313	Session interrupted; timeout
3314	Session interrupted; a new one will be established
3315	Negotiation failure
3316	Transfer session interrupted by operator
3317	Timeout detected
3318	Invalid protocol parameter
3319	Incorrect number of bytes or records transferred
3320	Too many checkpoints during a transfer
3399	Miscellaneous error (see TRC if not null)
34xx	DES/RSA SECURITY negotiation error
3400	Authentication negotiation
3401	Authentication rejected
3402	Authentication not supported
3403	Authentication algorithm not supported
3404	Authentication protocol error
3404	Authentication failed
3410	Sealing not supported (ETEBAC5)
3411	Sealing algorithm not supported
3412	Sealing process not supported

PeSIT Return Codes (continued)

Return Code	Description
3413	Sealing facility disabled
3414	Sealing parameters xfer mode not supported
3415	Sealing parameters protection required
3416	Sealing required
3417	Invalid sealing parameters
3418	Incompatible sealing parameters (p73)
3419	Intermediary seal error
3430	Ciphering not supported (ETEBAC5)
3431	ciphering algorithm not supported
3432	ciphering process not supported
3433	ciphering facility disabled
3434	ciphering parameters xfer mode not supported
3435	ciphering parameters protection required
3436	ciphering required
3437	Invalid ciphering parameters
3438	Incompatible ciphering parameters (p75)
3450	Signature not supported (ETEBAC5)
3451	Signature algorithm not supported
3452	Signature process not supported
3453	Signature facility disabled
3454	Double signature facility not supported
3455	Double signature required
3456	Signature required
3457	First signature error
3458	First signature error (p77)
3459	Second signature error
3460	Second signature error (p77)
3461	Signature error (ACK)
3462	Date/Time error (ACK)
3470	Invalid accreditation (p71)
3471	Invalid accreditation (type)
3472	Invalid accreditation (ID)
3473	Accreditation rejected

PeSIT Return Codes (continued)

Return Code	Description
3474	Accreditation disabled
3475	Accreditation disabled
3476	Accreditation not supported
3477	Accreditation error
3478	Accreditation test mode not supported
3479	Accreditation real mode not supported
3490	Invalid accreditation (p77)
3491	First accreditation invalid (type)
3492	Invalid first accreditation (ID)
3493	First accreditation rejected
3494	First accreditation disabled
3495	First accreditation disabled
3496	First accreditation not supported
3497	First accreditation error
3498	Second accreditation test mode not supported
3500	Invalid first accreditation (p77)
3501	Second accreditation invalid (type)
3502	Invalid Second accreditation (ID)
3503	Second accreditation rejected
3504	Second accreditation disabled
3505	Second accreditation disabled
3506	Second accreditation not supported
3507	Second accreditation error
3508	Second accreditation test mode not supported

PeSIT Return Codes and Connect:Express Codes

When a Partner is not a Connect:Express system, the fact that the TRC is null (TRC=0000) is sufficient in most cases to determine that the error was detected on the remote side. The table below shows the general relationship between PeSIT return codes and TRC.

PRC	TRC	Type	Action
000	0000		OK
301	0000	Invalid local identification PI (4)	Error detected by local or remote addressee

PRC	TRC	Type	Action
304	0000	Invalid remote identification PI (3)	Error detected by local or remote addressee
308	0000		Error detected by local or remote addressee
310	0000	Network error (+ NRC)	Detected locally, can be on both sides
312	0000	Session interruption	Local or remote requested interrupt
216	0000	Transfer interruption	Local or remote requested interrupt
XXX	0000		PRC issued by remote
XXX	yyyyR		PRC issued by remote
XXX	YYYY		PRC issued by local

The next sections describe the PRC and TRC that you can receive during the connection and file selection phases. During these phases, the automatic retry process may be activated depending on the PRC received. Refer to the Administration Guide for more information.

The PRC code is issued by the remote Partner. The local TRC code issued is 2077, as long as the retry process is operating. It changes to 2076 when the process is aborted and the Partner is disabled. If the RESTART option is YES in the Partner directory, then the retry procedure is never stopped unless you disable the Partner.

Some connection return codes invoke the call retry procedure activation. For PeSIT, the default T1B2PCOD entry contains the codes as indicated in the tables below.

PeSIT Codes and Connect:Express Codes for Outgoing Connections

PRC	TRC	Type	Action
300	2077, 2076	Remote system overflow	Call retry, aborted
303	2077, 2076		Call retry, aborted
309	2077, 2076		Call retry, aborted
310	2077, 2076		Call retry, aborted
317	2077, 2076		Call retry, aborted
311	2076	Protocol error	ABORT received, aborted
318	2076	Protocol error	ABORT received, aborted
XXX	2076		ABORT received, aborted
310	2076	Network error	Network abort, aborted

PeSIT Codes and Connect:Express Codes for Incoming Connections

PRC	TRC	Type	Action
300	21xx	Connect:Express resource disabled	RCONNECT(300) sent
311	1yyy	Protocol error	ABORT(311) sent
318	1yyy	Protocol error	ABORT(318) sent
XXX	tttt		ABORT(XXX) sent
310	0000	Network error	Session cut

PeSIT Transfer Selection Failure

If the RESTART option is YES in the Partners directory, then all outgoing transfers rejected with the following PRC codes are automatically restarted by Connect:Express. For PeSIT, the default T1B2PCOD entry contains the codes as indicated in the table below.

PRC	TRC	Type	Action
100	0000		Issued by remote
201	0000		Issued by remote
202	0000		Issued by remote
203	0000		Issued by remote
207	0000		Issued by remote
221	0000		Issued by remote
225	0000		Issued by remote
233	0000	Restart rejected: next retry will be without restart	Issued by remote
300	0000		Issued by remote
317	0000, 2045	Time out detected	Issued by remote or local
310	2044, 2045, 2105		Issued by local

PeSIT PRC/TRC Codes Issued by Connect:Express

Listed below are the PRC and TRC codes issued by Connect:Express. Some PRC codes can only be issued by a user exit, and in this case the TRC form is 4....

PRC and TRC Issued by Connect:Express, 200-229

PRC	TRC	Error Description	
200		4 . . .	(EXIT)
201	2069 2016 2 . . .	4 . . .	None
202		4 . . .	(EXIT)

PRC and TRC Issued by Connect:Express, 200-229 (continued)

PRC	TRC		Error Description
203		4 . . .	(EXIT)
204	2 . . .	4 . . .	None
205	2013 2082 2054	4 . . .	(EXIT)
206		4 . . .	(EXIT)
207	2 . . .	4 . . .	(EXIT)
208		4 . . .	(EXIT)
209		4 . . .	(EXIT)
210		4 . . .	(EXIT)
211		4 . . .	(EXIT)
212		4 . . .	(EXIT)
213	3 . . .	4 . . .	I/O erros
214	5001 3.07 3.05	4 . . .	Invalid restart point
215		4 . . .	(EXIT)
216	0000 5005 4 . . .	4 . . .	Transfer interrupted
217		4 . . .	(EXIT)
218	5009	4 . . .	Cannot resynchronize
219	5008	4 . . .	File space full
220		4 . . .	(EXIT)
221		4 . . .	(EXIT)
222	5004	4 . . .	Restart point does not match
223		4 . . .	(EXIT)
224		4 . . .	(EXIT)
225		4 . . .	(EXIT)
226	2010, 2011, 2012, 2028, 2036, 2047, 2040, 2041, 2047, 2059, 2101, 2118	4 . . .	Error detected during directories control
227		4 . . .	(EXIT)
228		4 . . .	(EXIT)
229		4 . . .	(EXIT)

PRC and TRC Issued by Connect:Express, 230-299, 100

PRC	TRC		Error Description
230		4 . . .	(EXIT)
231		4 . . .	(EXIT)
232		4 . . .	(EXIT)
233	2043	4 . . .	Cannot restart
234		4 . . .	(EXIT)

PRC and TRC Issued by Connect:Express, 230-299, 100 (continued)

PRC	TRC		Error Description
235		4 . . .	(EXIT)
236		4 . . .	(EXIT)
237		4 . . .	(EXIT)
238		4 . . .	(EXIT)
239		4 . . .	(EXIT)
240		4 . . .	(EXIT)
241		4 . . .	(EXIT)
242		4 . . .	(EXIT)
243		4 . . .	(EXIT)
244	1521	4 . . .	Compression not available
245		4 . . .	(EXIT)
246		4 . . .	(EXIT)
247		4 . . .	(EXIT)
248		4 . . .	(EXIT)
249		4 . . .	(EXIT)
250		4 . . .	(EXIT)
251		4 . . .	(EXIT)
252		4 . . .	(EXIT)
253		4 . . .	(EXIT)
254		4 . . .	(EXIT)
255		4 . . .	(EXIT)
256		4 . . .	(EXIT)
257		4 . . .	(EXIT)
299	9 . . . 4 . . . 6 . . . 1507 5002	4 . . .	(EXIT)
100		4 . . .	(EXIT)

PRC and TRC Issued by Connect:Express, 4xx, 300

PRC	TRC		Error Description
4XX	4. .		RSA/DES Security code
300	0000 → 2131 . . 2155		Resource disabled
301	0000 2011	4 . . .	(EXIT)
302		4 . . .	(EXIT)
303		4 . . .	(EXIT)
304	0000 2011	4 . . .	None
305		4 . . .	(EXIT)
306		4 . . .	(EXIT)
307		4 . . .	(EXIT)
308	0000	4 . . .	(EXIT)

PRC and TRC Issued by Connect:Express, 4xx, 300 (continued)

PRC	TRC		Error Description
309		4 . . .	(EXIT)
310	0000 9999	4 . . .	None
311	1 . . .	4 . . .	Protocol error
311	44 . .	4 . . .	Security protocol error
312	8 . . .	4 . . .	(EXIT)
312		4 . . .	Effector abended
313		4 . . .	(EXIT)
314		4 . . .	(EXIT)
315		4 . . .	(EXIT)
316		4 . . .	(EXIT)
317	0000	4 . . .	(EXIT) RESTART
318	1 . . . 44 . .	4 . . .	Protocol error
319	5006 5007 15 . .	4 . . .	Bytes/records count error
320		4 . . .	(EXIT)
321		4 . . .	(EXIT)
322		4 . . .	(EXIT)
323	1501	4 . . .	CRC Control mandatory
324		4 . . .	(EXIT)
325		4 . . .	(EXIT)
326		4 . . .	(EXIT)
327		4 . . .	(EXIT)
328		4 . . .	(EXIT)
329		4 . . .	(EXIT)
330		4 . . .	(EXIT)
331		4 . . .	(EXIT)
332	1611	4 . . .	One way session
333	1612	4 . . .	One way session
399	1522 46 . . 9 . . .	4 . . .	(EXIT)

ODETTE-FTP Return Codes

When a TRC is absent, typically this means that an error has been detected on the remote side. When no remote TRC is received, all TRCs displayed are local. Some PRC codes are self sufficient and no TRC is issued. Some ODETTE-FTP return codes are never issued by Connect:Express. You can return ODETTE-FTP codes from your own exits.

The ODETTE-FTP return codes are divided into Session return codes (PRC=0xxx) and Transfer return codes (PRC=1xxx). The ODETTE-FTP return codes are listed below.

ODETTE-FTP Session Return Codes

Return Code	Description
0000	Normal session termination
0001	Command not recognized
0002	Protocol violation
0003	User code not known
0004	Invalid password
0005	Local site emergency close down
0006	Command contain invalid data
0007	NSDU size error
0008	Resources not available
0009	Time out
0010	Mode or capabilities incompatible
0099	Unspecified reason

ODETTE-FTP Transfer Return Codes

Return Code	Description
1001	Invalid file name
1002	Invalid destination
1003	Invalid origin
1004	Storage record size not supported
1005	Maximum record length not supported
1006	File size is too big
1007	Not used
1008	Not used
1009	Not used
1010	Invalid record count
1011	Invalid byte count
1012	Access method failure
1013	Duplicate file
1099	Unspecified reason

ODETTE-FTP Codes and Connect:Express Codes

When a Partner is not a Connect:Express Partner and the TRC is null (TRC=0000), this suggests that the error was detected on the remote side. The table below shows the general relationship between ODETTE-FTP return codes and TRC.

PRC	TRC	Result	Action
0000	0000		OK
0003	0000	User code not known	
0004	0000	Invalid password	
XXXX	0000		PRC issued by remote
XXXX	YYYY		PRC issued by local

The next sections show the PRC and TRC that you can receive during the connection and file selection phases. During these phases, the automatic retry process may be activated depending on the PRC received. Refer to the Administration Guide for more information.

ODETTE-FTP Outgoing Connections

The PRC code is issued by the remote Partner. The local TRC code issued is 2077, as long as the retry process is operating. It changes to 2076 when the process is aborted and the Partner is disabled. If the RESTART option is YES in the Partners directory, then the retry procedure is never stopped, unless you disable the Partner.

Some connection return codes invoke the call retry procedure activation. For ODETTE-FTP, the default T1B2PCOD entry contains the code 0005, as shown in the following tables.

PRC	TRC	Result	Action
0005	2077, 2076	Session interrupted	Call retry, aborted
xxxx	2076		Aborted

ODETTE-FTP Incoming Connections

PRC	TRC	Result	Action
0005	XXXX		ESID sent

ODETTE-FTP Transfer Selection Failure and Transfer Restart

If the RESTART option is YES in the Partners directory, then all outgoing transfers interrupted with the following PRC codes are automatically restarted by Connect:Express. For ODETTE-FTP, the default T1B2PCOD entry contains the code 005, as shown in the table below.

PRC	TRC	Result	Action
0005	XXXX		ESID sent

ODETTE-FTP PRC and TRC Codes Issued by Connect:Express

Listed below are the PRC and TRC codes issued by Connect:Express. Some PRC codes can only be issued by user exits, and in this case the TRC form is 4 . . . .

PRC	TRC		Error Description
0001	16. .	4. . .	Command not recognized
0002	5004/8	4. . .	Protocol violation
003	1501	4. . .	User code not known
004	1502	4. . .	Invalid password
005	0000 2140/145	4. . .	Local site emergency close down
006	15..	4. . .	Command contain invalid parameter
007	11..	4. . .	NSDU size error
008		4. . .	Resources not available
009		4. . .	Time out
0010	1504/28/20/24/03/26	4. . .	Mode or capabilities incompatible
0099	9. . . 46. . 6. . .	4. . .	Unspecified reason
1001	2. . . 2013/82/43/54	4. . .	Invalid file name
1002	2011	4. . .	Invalid destination
1003	2011	4. . .	Invalid origin
1004		4. . .	Record size not supported
1005		4. . .	Max record length not supported
1006		4. . .	File size is too big
1007		4. . .	
1008		4. . .	
1009		4. . .	
1010		4. . .	Invalid record count
1011	5006	4. . .	Invalid byte count
1012	2. . . 3. . . 6. . . 45. .	4. . .	Access method failure
1013		4. . .	Duplicate file
1099	2. . . 5. . . 6. . . 9. . .	4. . .	miscellaneous

Messages

This section lists WTO messages issued by the Monitor, the APM, and the ANM, as well as SYSLOG messages from the Monitor and APM.

WTO Messages Issued by the Monitor

Local operational errors are reported in the SYSMSG file, following the current messaging rules.

```
10/04/00  14:04:32  TOM8046I SUBSYSTEM REQUEST CLEAN UP COMPLETED
10/04/00  14:04:32  TOM8047I SUBSYSTEM INTERFACE INHIBITED
10/04/00  14:04:32  TOM8050I SHORT  ABNORMAL TERMINATION END
```

Time stamp-subsystem Name (TOM8)–message number (046)–message type (I),TEXT. Message types are standard IBM message types as shown in the table below.

Message	Description
W	Warning
E	Error
I	Information

The following table lists the WTO messages issued by the monitor.

WTO Messages Issued by the Monitor

Code	Description
dtsm000E	NIP MESSAGE NOT EDITABLE Explanation: Unrecoverable error before initialization. System action: The task is terminated. User action: Call programming support.

dtsm001E	INVALID EXEC PARM
	Explanation: Parameters passed to the main monitor program P1B2P000 are missing or invalid.
	System action: The task is terminated.
	User action: Check the JCL procedure for valid "PARM" value.
	Specify the PARM of the exec as follows:
	//STEP01 EXEC PGM=P1B2P000,
	// PARM='RUN=x,SSN=TOMx,PLEX=x,XRF=x,XCFGROUP=cccccccc',

dtsm002E	xxxxxxxx DYNAMIC LOADING ERROR Explanation: An error has been detected when loading a module. System action: The task is terminated. User action: Check the "LOADLIB" in the JCL or in the LINKLIST, the module specified by "xxxxxxxx" is missing.
dtsm003E	xxxxxxxx DD STATEMENT MISSING Explanation: A file used by the monitor is not defined by a "DD" statement in the JCL. System action: The task is terminated. User action: Check the JCL: a statement "//xxxxxxxx DD ..." is missing.
dtsm004E	INVALID SSCVT CHAIN Explanation: An error has been detected in the subsystem list structure. System action: The task is terminated. User action: Take A CSA dump. Call your PROGRAMMING SUPPORT PERSONNEL.

The 'dtsm' prefix is replaced by your Connect:Express subsystem name.

WTO Messages Issued by the Monitor (continued)

Code	Description
dtsm005E	xxxx IS NOT A PREDEFINED SUBSYSTEM Explanation: The subsystem "xxxx" does not exist. System action: The task is terminated. User action: If the subsystem "xxxx" should really exist it has not been defined in the "IEFSSNxx" member of the SYS1.PARMLIB.
dtsm006E	xxxx ALREADY STARTED Explanation: The monitor "TOMx", the "ANM" or an "APMxx" is already started. System action: The task is terminated. User action: Before starting a monitor, check that all related address spaces (Monitor, ANM, AFM, APMxx) are ended.
dtsm008E	GETMAIN REJECTED TABLE=xxx Explanation: Insufficient storage. System action: If the error occurred during the monitor initialization step, the task is terminated. User action: A "GETMAIN" has been issued for table "xxx" but there is not enough virtual storage available. Check the REGION value in the JCL.
dtsm009E	xxxxxxxx INVALID/MISSING SYSIN CARD RC=xx Explanation: The monitor SYSIN contains an invalid card. System action: The task is terminated. User action: Check the monitor sysin file for invalid or missing card. All mandatory cards must exist even if it is a hot start. See Appendix D.
dtsm010E	ERROR DETECTED IN SYSIN FILE Explanation: The monitor SYSIN file cannot be processed. System action: The task is terminated. User action: Check the RECFM (F) or LRECL (80) of the monitor sysin file and the contents (a mandatory parameter may be missing). See Appendix D.

The 'dtsm' prefix is replaced by your Connect:Express subsystem name.

WTO Messages Issued by the Monitor

Code	Description
dtsm011S	STOPPED FOLLOWING PREVIOUS MESSAGE RC=xxxx Explanation: The monitor has detected an error. System action: The task is terminated. User action: Check previous messages for the reason.

Reason Code	Description
016	DD STATEMENT MISSING
018	ERROR LOADING C:X PGM
019	EXEC PARM ERROR (NO PARM GIVEN)
020	EXEC PARM ERROR (PARM RUN=)
021	EXEC PARM ERROR (PARM SSN=)
022	OPEN ERROR ON SYSCHK
023	OPEN ERROR ON SYSLOG
024	P1B2P000 INVALID LEVEL
025	INVALID SSVT CHAIN

WTO Messages Issued by the Monitor (continued)

Code	Description
dtsm011S	STOPPED FOLLOWING PREVIOUS MESSAGE RC=xxxx Explanation: The monitor has detected an error. System action: The task is terminated. User action: Check previous messages for the reason.

Reason Code	Description
032	C:X ALREADY STARTED
033	ANM ALREADY STARTED
034	APM ALREADY STARTED
035	SSN NOT PREDEFINED SUBSYSTEM
036	T1B2PS?0 NOT FOUND
037	T1B2PCOD NOT FOUND
038	T1B2PP?? NOT FOUND
039	ERROR FOUND IN SYSIN
040	T1B2PTCT GETMAIN ERROR
041	T1B2PRCT GETMAIN ERROR
048	CHECKPOINT ERROR READING PCT
049	CHECKPOINT ERROR READING FCT
050	CHECKPOINT ERROR READING RCT
051	CHECKPOINT ERROR READING TCT
052	ERROR ON SYSX25
053	ERROR ON SYSSNA
067	T1B2PSCE GETMAIN ERROR
070	DSNRCY (HOT) /= DSNRCY (COLD)
071	CSA MODULE DO NOT MATCH
074	SSVT CSA GETMAIN ERROR
078	L1B2P046 COMAREA GETMAIN ERROR
079	OPEN ERROR ON SYSLIB
080	ERROR LOADING EXITJNL
081	L1B2P049 COMAREA GETMAIN ERROR
082	L1B2P049 ATTACH ERROR
083	L1B2P049 ATTACH FAILED
084	L1B2P04? ATTACH FAILED
085	L1B2P046 ATTACH ERROR

WTO Messages Issued by the Monitor (continued)

Code	Description
dtsm011S	STOPPED FOLLOWING PREVIOUS MESSAGE RC=xxxx Explanation: The monitor has detected an error. System action: The task is terminated. User action: Check previous messages for the reason.

Reason Code	Description
086	L1B2P046 ATTACH FAILED
087	L1B2P04? ATTACH ERROR
088	L1B2P045 ATTACH ERROR
089	L1B2P045 ATTACH FAILED
090	L1B2P047 COMAREA GETMAIN ERROR (C:X local)
091	L1B2P047 ATTACH ERROR (C:X local)
092	L1B2P047 ATTACH FAILED (C:X local)
093	L1B2P047 ATTACH FAILED (C:X local)
096	L1B2P04? ATTACH FAILED
097	INVALID MAXSRQ
098	L1B2P040 COMAREA GETMAIN ERROR
099	L1B2P044 ATTACH ERROR
100	L1B2P044 ATTACH FAILED
101	L1B2P04? ATTACH FAILED
102	L1B2P043 ATTACH ERROR
103	L1B2P043 ATTACH FAILED
104	L1B2P04? ATTACH FAILED
105	L1B2P042 ATTACH ERROR
112	L1B2P042 ATTACH FAILED
113	L1B2P04? ATTACH FAILED
114	L1B2P041 ATTACH ERROR
115	L1B2P041 ATTACH FAILED
116	L1B2P04? ATTACH FAILED
117	L1B2P040 COMAREA GETMAIN ERROR
118	L1B2P040 ATTACH ERROR
119	L1B2P040 ATTACH FAILED
120	L1B2P04? ATTACH FAILED
121	OPTS FOR INFORMATION

WTO Messages Issued by the Monitor (continued)

Code	Description
dtsm011S	STOPPED FOLLOWING PREVIOUS MESSAGE RC=xxxx Explanation: The monitor has detected an error. System action: The task is terminated. User action: Check previous messages for the reason.

Reason Code	Description
122	L1B2P072 ATTACH FAILED
128	ERROR START ANM PROCEDURE
130	START ANM FAILED
131	ANM PREMATURE END
133	START APM PROCEDURE FAILED
134	START APM PROCEDURE FAILED
135	ERROR LOADING L1B2PCNX
136	L1B2PCNX INIT ERROR
137	ANM PREMATURE END
144	APM PREMATURE END
145	L1B2P04? PREMATURE END
146	L1B2P040 PREMATURE END
147	T1B2PZAPL GETMAIN ERROR
149	P1B2P000 INVALID LEVEL
150	P1B2P000 INVALID LEVEL
151	ERROR LOADING T1B2PSLD
152	ERROR READING CVT FROM SYSCHK
153	ERROR READING CVT FROM SYSCHK
154	SSN (HOT/COLD) MISMATCHED
155	INVALID T1B2PCVT LEVEL
160	INVALID T1B2PCVT LEVEL
161	ERROR LOADING CSA PGM
162	OPEN ERROR ON SYSFIL
163	POINT ERROR ON SYSFIL
164	READ ERROR ON SYSFIL
165	OPEN ERROR ON SYSPAR
166	POINT ERROR ON SYSPAR
167	READ ERROR ON SYSPAR

WTO Messages Issued by the Monitor (continued)

Code	Description
dtsm011S	STOPPED FOLLOWING PREVIOUS MESSAGE RC=xxxx Explanation: The monitor has detected an error. System action: The task is terminated. User action: Check previous messages for the reason.

Reason Code	Description
168	T1B2PFCT GETMAIN ERROR
169	POINT ERROR ON SYSFIL
173	P1B2P000 STAE ENTERED
174	P1B2P000 STAE ENTERED
177	READ ERROR ON SYSFIL
180	CLOSE ERROR ON SYSFIL
182	T1B2PPCT GETMAIN ERROR
183	POINT ERROR ON SYSPAR
184	READ ERROR ON SYSPAR
185	CLOSE ERROR ON SYSPAR
192	T1B2PSST GETMAIN ERROR
193	OPEN ERROR ON SYSX25
194	T1B2PX25 GETMAIN ERROR
195	OPEN ERROR ON SYSX25
196	OPEN ERROR ON SYSSNA
197	T1B2PSNA GETMAIN ERROR
198	OPEN ERROR ON SYSSNA
199	OPEN ERROR ON SYS???
200	BLDL ERROR ON SYS???
201	POINT ERROR ON SYS???
208	READ ERROR ON SYS???
209	READ ERROR ON SYS???
210	T1B2P??? GETMAIN ERROR
211	CLOSE ERROR ON SYS???
212	ERROR ON SYSTCP
213	CXPLEX MISSING
214	SYSPRTX MISSING
215	ERROR FOUND IN CXPLEX FILE

WTO Messages Issued by the Monitor (continued)

Code	Description
dtsm011S	STOPPED FOLLOWING PREVIOUS MESSAGE RC=xxxx Explanation: The monitor has detected an error. System action: The task is terminated. User action: Check previous messages for the reason.

Reason Code	Description
216	JOINXCF ERROR
217	ERROR LOADING L1B2PXRF
218	LIB2PAPS ATTACH FAILED
219	LIB2PAPS ATTACH FAILED
240	ERROR ON SYSCHK

WTO Messages Issued by the Monitor

Code	Description
dtsm012E	"ATTACH" ERROR xxxxxxxx RC= Snnn/Uvvv Explanation: An error has been detected by the "ATTACH" routine. System action: The task is terminated. User action: Check the C:X "LOADLIB" in SYSLIB/STEPLIB/LINKLIST for the member specified by "xxxxxxx". If module is 'L1B2P055', check your journal exit 'UEXJNL='.
dtsm013E	NO REQUEST IN SCHEDULING LIST Explanation: An error has been detected. System action: The task is terminated. User action: Call programming support.
dtsm014I	SUBSYSTEM FUNCTION SUCCESFULLY STARTED Explanation: Subsystem is initialized. System action: None. User action: None.
dtsm015I	COMMUNICATION WITH OPERATOR OPENED Explanation: OS/390 'modify' commands accepted. System action: None. User action: None.
dtsm016I	TOM INITIALIZATION COMPLETE Explanation: Monitor and network are fully started. System action: None. User action: None, check for APM's.
dtsm017E	ACCESS ERROR ON FILE " SYSLOG " RC=?? Explanation: Severe i/o error. System action: The task is terminated. User action: Call programming support.
dtsm018E	ACCESS ERROR ON FILE " SYSCHK " RC=?? Explanation: Severe i/o error. System action: The task is terminated. User action: Call programming support.

The 'dtsm' prefix is replaced by your 'C:X' subsystem name.

WTO Messages Issued by the Monitor (continued)

Code	Description
dtsm019E	ERROR ON ???????? FOR INIT ??? OP=?? Explanation: Severe error on function. System action: The task is terminated. User action: Call programming support.
dtsm020E	INVALID AUTH FILE, CALL SUPPORT. Explanation: The asset protection key file is not valid. System action: Initialization is stopped. User action: Activate the SYSPAPS DD card in the TOM JCL, restart Connect:Express, and send this file to technical support .
dtsm021E	INVALID COMMAND Explanation: The operator entered an invalid command. System action: None. User action: Correct and retry.
dtsm022W	TOM IS ALREADY IN THIS STATUS Explanation: The operator entered a command, but the required state is the same as the actual one. System action: None. User action: Check and retry.
dtsm023I	TOM IS IN STOP STATUS Explanation: The operator entered a modify-stop command. System action: The monitor is temporarily stopped. User action: None.
dtsm024I	TOM IS IN ACTIVE STATUS Explanation: The operator entered a modify-start command. System action: The monitor is re-activated. User action: None.
dtsm025I	TOM ACKNOWLEDGES STOP COMMAND Explanation: The operator entered a stop command. System action: The monitor is stopping normally, after interrupting the current transfers. User action: None.
dtsm026I	TOM ACKNOWLEDGES IMMEDIATE STOP COMMAND Explanation: The operator entered a cancel command. System action: The monitor is stopping shortly, after interrupting the current transfers. User action: Never enter a second cancel command.
dtsm027I	TOM : ? / FCT: ? / PCT: ? / RCT: ? Explanation: Monitor states. System action: None. User action: None.
dtsm028I	APM?? : ? A-EF: ?? U-EF: ?? ETC: ? Explanation: Monitor states. System action: None. User action: None.
dtsm029I	??? IS NOW ENABLED Explanation: Monitor component new state. System action: The component is enabled. User action: None.
dtsm030I	??? IS NOW DISABLED Explanation: Monitor component new state. System action: The component is disabled. User action: None.

The 'dtsm' prefix is replaced by your 'C:X' subsystem name.

WTO Messages Issued by the Monitor (continued)

Code	Description
dtsm031W	??? IS ALREADY IN THIS STATUS Explanation: Command entered, but the required state is the same as the actual one. System action: None. User action: Check and retry.
dtsm032I	CSA RELOAD REQUESTED BY OPERATOR Explanation: $LOAD$ command entered. System action: The action will be performed at restart. User action: Restart the monitor.
dtsm033I	???????? / RELOAD PARTNER LIST RC=?? Explanation: ACF= command entered. System action: The action is performed. User action: Check rc.
dtsm034E	APM ?? HAS NOT A VALID STATE Explanation: A modify command against an APM was not executed. System action: None. User action: Check the reason (APM active/stooping/...).
dtsm035I	APM ?? (????) COMMAND EXECUTED Explanation: APM component new state. System action: The component will take this new state. User action: None.
dtsm036E	INITIALIZATION ERROR RC=?? Explanation: Connect:Express could not start. System action: Connect:Express startup fails. User action: Check previous error message.
dtsm037E	???????? PARTNER LIST MEMBER NOT FOUND Explanation: Invalid ACF= command entered. System action: None. User action: Correct and retry,
dtsm038E	ANM INITIALIZATION PENDING Explanation: The ANM initialization is not complete for now. System action: The monitor is waiting the normal completion. User action: Check the ANM is well started, or check the sysout of the failing ANM for more information.
dtsm039E	ANM INITIALIZATION FAILED Explanation: The ANM initialization cannot complete successfully. System action: The task is terminated. User action: The monitor cannot start without the ANM. Check the sysout of the failing ANM for more information.
dtsm040E	APM INITIALIZATION PENDING Explanation: The APM initialization is not complete for now. System action: The monitor is waiting the normal completion. User action: Check the APM is well started, or check the sysout of the failing APM for more information.
dtsm041E	APM INITIALIZATION FAILED Explanation: The APM initialization cannot complete successfully. System action: The task continues, but the APMxx is set "ABENDED". User action: Check the sysout of the APM for more information, and start it again.

The 'dtsm' prefix is replaced by your 'C:X' subsystem name.

WTO Messages Issued by the Monitor (continued)

Code	Description
dtsm042I	TOM IS WAITING FOR A.P.M. TERMINATION Explanation: The APM termination is not complete. System action: Monitor will stop in any case. User action: Check the sysout of the APM for more information. Some minutes of delay is normal, else cancel it with dump.
dtsm043S	"ABEND" : Sxxx/Uyyy , TOM IS ENDING Explanation: An ABEND occurred within monitor main task. System action: The task is terminated. User action: The monitor can be restarted (hot or cold). Depending on the problem, call your PROGRAMMING SUPPORT.
dtsm044S	SUB-TASK ERROR -> STOP Explanation: An ABEND occurred within a monitor sub-task. System action: The task is terminated. User action: The monitor can be restarted (hot or cold). Depending on the problem, call your PROGRAMMING SUPPORT.
dtsm045I	COMMUNICATION WITH OPERATOR CLOSED Explanation: Operator stop command entered. System action: The monitor is terminating. User action: None.
dtsm046I	SUBSYSTEM REQUEST CLEAN UP COMPLETE Explanation: Stop process entered. System action: The monitor is terminating. User action: None.
dtsm047I	SUBSYSTEM REQUEST INTERFACE INHIBITED Explanation: Stop process entered. System action: The monitor is terminating. User action: None.
dtsm048I	NORMAL TERMINATION PROCEDURE ENDED Explanation: Stop process entered. System action: The monitor is terminating. User action: None.
dtsm049E	ANM ABNORMAL TERMINATION DETECTED Explanation: The ANM reports to the monitor an abnormal termination. System action: The task is terminated. User action: The monitor can be restarted (hot or cold). Depending on the problem, call your PROGRAMMING SUPPORT.
dtsm050I	SHORT ABNORMAL TERMINATION END Explanation: During monitor recovery process a new error has been detected. System action: Short termination processed. User action: Check previous messages for more details.
dtsm051E	???????? INVALID FUNCTION SPECIFIED Explanation: An invalid function was entered. System action: None. User action: Correct and retry.
dtsm052E	MONITOR NOT DORMANT, REQUEST DENIED Explanation: The requested function cannot be performed. System action: None. User action: Wait complete stop and retry.

The 'dtsm' prefix is replaced by your 'C:X' subsystem name.

WTO Messages Issued by the Monitor (continued)

Code	Description
dtsm053E	APM ?? , ACTIVE EFF: ??, REQUEST DENIED Explanation: The requested function cannot be performed. System action: None. User action: Wait complete stop and retry.
dtsm054E	INIT SESSION PENDING, REQUEST DENIED Explanation: The requested function cannot be performed. System action: None. User action: Wait complete stop and retry.
dtsm055E	COMMAND REJECTED BY ANM, RETRY LATER Explanation: The requested function cannot be performed. System action: None. User action: Check reason and retry.
dtsm056I	SDUMP COMMAND PROCESSED RC=?? Explanation: The requested SDUMP of all Connect:Express asids were tried. System action: One SYS1.DUMP?? was taken. User action: Check for successful completion rc=00, else check messages (sys1.dump too small/...).
dtsm057W	xxxxxxxx CSA MODULES DO NOT MATCH Explanation: At each start, the monitor checks the CSA modules. If they are not identical to the modules in the Connect:Express library a message is issued. System action: If in the previous Connect:Express session the command "$LOAD$" has been issued, the task continues, else the task terminates. User action: The message can be issued if a new release of the product has been installed. In other cases check the reasons before restarting the monitor and depending on the problem call your PROGRAMMING SUPPORT.
dtsm058I	xxxxxxxx CSA MODULE RELOADED Explanation: The specified module has been loaded in CSA. System action: None. User action: The message can be issued if an IPL OS/390 has been executed or if a command "$LOAD$" has been issued.
dtsm059W	$LOAD$ AUTOMATIC SET, IF "OK" START AGAIN Explanation: The message is issued after message 057. System action: The task is terminated. User action: Refer to message 057 for more details.
dtsm060W	OPTION NOT AUTHORIZED TRC=xxxx Explanation: The message is issued if the option specified by xxxx is used. System action: None. User action: If the message is issued during the monitor initialization, check the partner directory for unauthorized option (TCP/IP, LU6.2 ...).
dtsm061E	INVALID CHECKPOINT SERVICE RC:??? Explanation: The checkpoint utility has detected an error. System action: The task is terminated. User action: Call programming support.
dtsm062E	INVALID FILE INITIALIZATION rc:?? Explanation: The checkpoint utility has detected an error. System action: The task is terminated. User action: Check installation and customization, then call programming support.

The 'dtsm' prefix is replaced by your 'C:X' subsystem name.

WTO Messages Issued by the Monitor (continued)

Code	Description
dtsm063E	INVALID FILE LEVEL (SYSCHK/SYSCHK2) Explanation: The checkpoint utility has detected the SYSCHK file has a wrong level. System action: The task is terminated. User action: Check the monitor JCL for right SYSCHK dsname. If a new product release has been installed refer to the installation documentation. A cold start is mandatory after formatting the checkpoint file.
dtsm064E	INVALID RQEMAX VALUE IN TOM SYSIN Explanation: The value indicates for RQEMAX in the monitor sysin file is invalid (default maximum value is 1020). System action: The task is terminated. User action: Check the monitor SYSIN and correct the RQEMAX card. If a new RQEMAX value is set the monitor must be cold start.
dtsm065E	OPEN ERROR ON xxxxxxxx RC=rrrr Explanation: The OPEN function has failed for the file specified by xxxxxxx. System action: See following messages. User action: Check the definition of the specified file.
dtsm066E	READ ERROR ON xxxxxxxx RC=rrrr Explanation: The READ function has failed for the file specified by xxxxxxx. System action: See following messages. User action: Check the definition of the specified file. If SYSCHK/SYSRCY/SYSLOG has been migrated/freed, refer to the Installation Guide and complete step 13, Create the Monitor Environment. You only have to do this for the file that has the problem.
dtsm067E	WRITE ERROR ON xxxxxxxx RC=rrrr Explanation: The WRITE function has failed for the file specified by xxxxxxx. System action: See following messages. User action: Check the definition of the specified file.
dtsm068E	CLOSE ERROR ON xxxxxxxx RC=rrrr Explanation: The CLOSE function has failed for the file specified by xxxxxxx. System action: See following messages. User action: None.
dtsm069E	POINT ERROR ON xxxxxxxx RC=rrrr Explanation: The POINT function has failed for the file specified by xxxxxxx. System action: See following messages. User action: None.
dtsm070E	NOTE ERROR ON xxxxxxxx RC=rrrr Explanation: The POINT function has failed for the file specified by xxxxxxx. System action: See following messages. User action: None.
dtsm071E	HOT RESTART: xxx INITIALIZATION ERROR Explanation: During hot restart the table specified by xxx cannot be initialized. System action: The task is terminated. User action: Check previous messages.
dtsm072E	HOT RESTART: SSN ARE DIFFERENT Explanation: During hot restart the SSN specified in the PARM of the EXEC is different from the one specified for the cold start. System action: The task is terminated. User action: Check the JCL for the parm of the EXEC and the DSNAME of the CHECKPOINT file.

The 'dtsm' prefix is replaced by your 'C:X' subsystem name.

WTO Messages Issued by the Monitor (continued)

Code	Description
dtsm073E	HOT RESTART: INVALID LEVEL DETECTED Explanation: During hot restart an invalid level has been detected. System action: The task is terminated. User action: Check the C:X level in LOADLIB/LINKLIST and the level of the CHECKPOINT file. If a new released has just been installed a cold start may be necessary.
dtsm074E	INVALID P1B2P000 LEVEL DETECTED Explanation: Programs have different level. System action: The task is terminated. User action: Call your PROGRAMMING SUPPORT.
dtsm075i	???????? CONNECTION EXIT INVOKED Explanation: User table invoked this exit. ('Y' fit in the WTO option field). System action: The user exit was performed. User action: None.
dtsm076i	???????? CONNECTION EXIT ERROR RC:?? Explanation: This user Connection exit returned an error code. System action: The user exit was performed with error. User action: Check the RC explanation.
dtsm077i	???????? USER EXIT DISABLED Explanation: This user exit created a serious error. System action: The user exit is disabled. User action: Check the RC explanation, and restart.
dtsm078i	L1B2PCNX INITIALIZED Explanation: The connection user exit server is initialized. System action: The user exit was called. User action: None.
dtsm079i	L1B2PCNX TERMINATED Explanation: The connection user exit driver is terminated. System action: The user exit was called. User action: None.
dtsm080E	L1B2PCNX DISABLED Explanation: The connection user exit driver has been disabled. System action: The CNX exit is disabled. User action: Check the RC explanation, and restart.
dtsm081E	L1B2P040 ERROR : ??????? ??? rc:?? Explanation: A major task failed. System action: Connect:Express is terminating. User action: Check the explanation and restart. Keep corresponding information and call programming support.
dtsm082E	SMF ERROR REQUEST ????/????? rc:?? Explanation: SMF task failed. System action: Connect:Express is terminating. User action: Check the explanation and restart. Keep corresponding information and call programming support.
dtsm083I	ANM INITIALIZATION COMPLETE Explanation: ANM Successfully started. System action: None. User action: None.

The 'dtsm' prefix is replaced by your 'C:X' subsystem name.

WTO Messages Issued by the Monitor (continued)

Code	Description
dtsm084I	APM ?? INITIALIZATION COMPLETE Explanation: APM Successfully started. System action: None. User action: None.
dtsm085I	ANM ?? ENDED CMP: S???/U??? Explanation: ANM ended. System action: None. User action: Check for no abend.
dtsm086I	APM ?? ENDED CMP: S???/U??? Explanation: APM ended. System action: None. User action: Check for no abend.
dtsm087W	ANM HANDLER TERMINATED SRC=??? TRC=??? Explanation: ANM ended. System action: This handler is no more available. User action: Check the reason of termination.
dtsm088W	APM ?? CANCELLED BY OPERATOR Explanation: APM ended. System action: This APM is no more available. User action: Check the reason of cancel termination.
dtsm089E	XPOST ERROR FOR ADDRESS SPACE ???????? Explanation: One asid is in error. System action: Connect:Express will terminate. User action: Check the reason of error, and call programming support.
dtsm099I	REQUEST COUNT RESTARTED ???????? Explanation: The request number has reached the 99999999 maximum value. System action: Connect:Express restart the count. User action: None.
dtsm100I	TOM IS STARTING ????????????? Explanation: The initialization process is running. System action: The SYSMSG file is opened. User action: None.
dtsm101I	EVENT MEMBER RELOAD REQUESTED BY OPERATOR Explanation: The EVT command was entered. System action: The event is refreshed. User action: None.
dtsm102I	???????? DEL/RELOAD REQUESTED BY OPERATOR Explanation: A refresh command was entered. System action: The member is refreshed. User action: None.
dtsm111E	xxxxxxxx INITIALIZATION FAILED RC=rrrr Explanation: The initialization of the task specified by xxxxxxx failed. System action: The task is terminated. User action: Check complementary messages.
dtsm112E	xxxxxxxx ERROR ON yyyyyyyy RC=rrrr Explanation: The task xxxxxxxx has detected an error when processing the file yyyyyyyy. System action: The task function is ended. User action: Check complementary messages.

The 'dtsm' prefix is replaced by your 'C:X' subsystem name.

WTO Messages Issued by the Monitor (continued)

Code	Description
dtsm112E	L1B2P043 ERROR ON SYSJCL RC=rrrr Explanation: The task L1B2P043 has detected an error when processing the file SYSJCL. System action: The UNLOAD/RELOAD function is ended. User action: Check complementary messages. If RC=0148, check that the JOBNAME format of SYSJCL is 'xxxxxx$$'.
dtsm113E	xxxxxxxx FUNCTION (z) FAILED REQ: rrrrrrrr Explanation: The task xxxxxxxx has detected an error during the function (Unload, Reload or Send_file) for the request rrrrrrr. System action: The task function is ended; the request is set disable. User action: Check complementary messages, apply corrections and enable the request.
dtsm114E	xxxxxxxx MBR yyyyyyyy NOT FOUND rrrrrrrr Explanation: The task xxxxxxxx has not found the member yyyyyyyy into the library rrrrrrrr. System action: The task function is ended; the request is set disable. User action: Check complementary messages, apply corrections and enable the request.
dtsm115E	xxxxxxxx HAS BEEN SET DISABLE Explanation: The task xxxxxxxx has found a sever error and cannot perform any kind of service. System action: The task is set disable. User action: If the task is "L1B2P043" all incoming requests with "RELOAD" will be rejected, all outgoing requests with "UNLOAD" will not be scheduled. To set the task enable: apply necessary corrections and issue the command "REF SYSJCL".
dtsm116E	xxxxxxxx REFRESH SYSJCL yyyyyyyy rc=rrrr Explanation: The refresh command for the task xxxxxxxx has failed or complete successfully. System action: None if refresh successful, else the task is set disable. User action: Check and process according to their messages if any. If RC=0106 (partial), some special unload/reload skeletons are missing in SYSJCL, create dummy members as named in dtsm114E to avoid message.
dtsm121I	???????? ???????? FILE DELETED Explanation: A request was purged, or workfile no more necessary at completion. System action: Delete of DSN done. User action: None.
dtsm123I	????????.????????.????????.???????? Explanation: DSN of dtsm121I. System action: None. User action: None.
dtsm126E	L1B2P045 ERROR RC=??? Explanation: The journal utility has detected an error. System action: The task is terminated. User action: Check installation and customization, then call programming support.
dtsm127E	L1B2P055 ERROR (ATTACH) Explanation: The journal exit driver has detected an error. System action: The task is terminated. User action: Check installation and customization for your STEPLIB/SYSLIB/LINKLIST journal user exit(s).
dtsm128E	L1B2P055 ERROR RC= S???/U??? Explanation: The journal exit driver has detected an error. System action: The task is terminated. User action: Check the process of your journal user exit(s).

The 'dtsm' prefix is replaced by your 'C:X' subsystem name.

WTO Messages Issued by the Monitor (continued)

Code	Description
dtsm129E	L1B2P055 ERROR : ???????? IS INACTIVE Explanation: The journal exit driver has detected a user exit error. System action: The task is terminated. User action: Check the process of your journal user exit(s).
dtsm130I	END OF TOM IS REPORTED TO APPLICATION Explanation: Application was notified. System action: Notification done. User action: None.
dtsm131I	REQ: ???????? WITH ???????? ERROR: ???? Explanation: An error was detected. System action: Request not processed. User action: Check for message and error code.
dtsm132I	REQUEST NUMBER CHANGED + xxxxxxx -> xxxxxxxx Explanation: Operator command changed request number start. System action: Requests will start from this number. User action: None.
dtsm133I	T1B2PX25 RELOADED BY OPERATOR Explanation: Operator command refreshed the table. System action: The table is refreshed totally/partially. User action: Check for rc=00, else see error messages then correct and retry.
dtsm134I	T1B2PSNA RELOADED BY OPERATOR Explanation: Operator command refreshed the table. System action: The table is refreshed totally/partially. User action: Check for rc=00, else see error messages then correct and retry.
dtsm135I	EXTERNAL APPLICATION TABLE CREATE Explanation: At start up, APLNUM was requested. System action: The table T1B2PAPL was created. User action: None.
dtsm136E	L1B2P048 ERROR RC= S???/U??? Explanation: The RSA utility has detected an error. System action: The task is terminated. User action: Check installation and customization, then call programming support.
dtsm137E	L1B2P058 ERROR (ATTACH) Explanation: The RSA utility has detected an error. System action: The task is terminated. User action: Check installation and customization, then call programming support.
dtsm138E	L1B2P058 ERROR RC= S???/U??? Explanation: The RSA utility has detected an error. System action: The task is terminated. User action: Check installation and customization, then call programming support.
dtsm139E	L1B2P058 ERROR : RSA-EXIT IS INACTIVE Explanation: The RSA utility has detected a user exit error. System action: The task is terminated. User action: Check installation and customization, then call programming support.
dtsm141E	ACB GENERATION FAILED Explanation: The ANM has detected a serious error. System action: The ANM is terminated. User action: Check installation and customization, then call programming support.

The 'dtsm' prefix is replaced by your 'C:X' subsystem name.

WTO Messages Issued by the Monitor (continued)

Code	Description
dtsm142E	RPL GENERATION FAILED Explanation: The ANM has detected a serious error. System action: The ANM is terminated. User action: Check installation and customization, then call programming support.
dtsm143E	??: OPEN ACB EF: ?? Explanation: The ANM has detected a serious error. System action: The ANM is terminated. User action: Check installation and customization, then call programming support.
dtsm144E	LU: ???????? ??: ???????? RC: ?? FB2: ?? Explanation: The ANM has detected a serious error. System action: The ANM is terminated. User action: Check installation and customization, then call programming support.
dtsm145E	LU: ???????? EX: LOSTERM RS: ?? Explanation: The ANM has detected a serious error. System action: The ANM is terminated. User action: Check installation and customization, syslogs, then call programming support.
dtsm146E	REQUEST MANAGER IS BUSY; TRY LATER Explanation: This asid could not perform process. System action: The process is terminated. User action: Check installation, syslogs, if problem persists then call programming support.
dtsm147E	LU: ???????? TIME-OUT DETECTED Explanation: The ANM has detected an error. System action: The process is terminated. User action: Check installation and customization, syslogs, then call programming support.
dtsm161I	SESSION ENDED WITH PARTNER : ???????? Explanation: The session ended. System action: The connection is terminated. User action: None.
dtsm162I	REJECT COMMAND IS RECEIVED Explanation: The session ended. System action: The connection is terminated. User action: Check syslogs.
dtsm163I	???? INVALID COMMAND STATE ? ?? Explanation: An invalid command was received. System action: The process is terminated. User action: Check syslogs, if problem persists then call programming support.
dtsm164I	LUIDT SESSION REJECTED RC= ? Explanation: An invalid command was received. System action: The process is terminated. User action: Check syslogs, if problem persists then call programming support.
dtsm165I	??????? PARTNER ERROR RC = ? Explanation: An error was encountered with this partner. System action: The process is terminated. User action: Check syslogs, if problem persists then call programming support.
dtsm166E	PC-PT: ???????? RC: ?? FDB: ?? Explanation: An critical error was encountered between asids. System action: The asid is terminated. User action: Call your PROGRAMMING SUPPORT.

The 'dtsm' prefix is replaced by your 'C:X' subsystem name.

WTO Messages Issued by the Monitor (continued)

Code	Description
dtsm191I	ATM IS NOT IN A VALID STATE (????????) Explanation: The operator trace command is not possible with the current ATM status shown between parenthesis . System action: No change done. User action: None.
dtsm192I	ATM PROCEDURE IS INITIALIZING ASID=????? Explanation: The operator activation command has been accepted. System action: The trace server is initializing in the ASID as indicated. User action: None.
dtsm193I	ATM INITIALIZATION FAILED RC=.. Explanation: The ATM could not initialize. System action: ATM status is inactive. User action: Check the code, fix and try again..
dtsm194I	ATM STARTED ASID=????? TRACE IS ACTIVE Explanation: The ATM successfully initialized. System action: ATM status is active, waiting for trace requests. User action: The operator can request traces.
dtsm195I	ATM ABNORMAL TERMINATION DETECTED Explanation: The ATM detected an error and stopped. System action: ATM status is inactive. User action: : check message dtsm196I.
dtsm196I	ATM ENDED CMP: SXXX/UXXX Explanation: The ATM terminated after a comand from the operator or an abnormal end. System action: ATM status is inactive. User action: In case of abnormal end, check the codes (System abend XXX or User abend XXX) and call support.
dtsm197W	ATM TRACE REQUEST FAILED RC=???? Explanation: Connect:Express detected an error while sending a trace request to ATM. System action: None. User action: check the codes and call support..
dtsm198I	ATM TRACE REQUEST ACCEPTED ID=???????? Explanation: The ATM recorded a trace request and identified it as indicated in the message. System action: None. User action: None.
dtsm199W	ATM TRACE REQUEST REJECTED : ???????????? Explanation: The ATM rejected a trace request for the reason indicated in the message. For example: duplicate request, partner is unknown. System action: None. User action: Refer to ATM SYSLOG, change trace request parameters and retry.
Dtsm200I	ATM TRACE REQUEST DELETED ID=???????? Explanation: The ATM deleted a trace request identified as indicated in the message. System action: None. User action: None.
dtsm201E	OLD SYSIN VALUE FOR ???????? WAS ???? Explanation: The parameter specified in the following message changed. System action: Change done. User action: None.

The 'dtsm' prefix is replaced by your 'C:X' subsystem name.

WTO Messages Issued by the Monitor (continued)

Code	Description
dtsm202E	RUN=H DENIED, RESTORE SYSIN VALUE FOR: Explanation: The parameter specified in the following message must have the same value for cold and hot start. System action: The task is terminated. User action: If that value must be changed a cold start must be specified.
dtsm204i	'MODIFY COMMAND' Explanation: The Command displayed was entered by the operator. System action: This message is followed by another message stating the result of the action. User action: None.
dtsm205E	xxxxxxxx PARTNER LIST MEMBER TOO BIG Explanation: The partner list specified contains more than 128 partners. System action: The partner list is not processed. User action: Modify the partner list and issue the command "ACF=xxxxxxxx"
dtsm206E	HOT START: SYSRCY DSNAME HAS BEEN CHANGED Explanation: The SYSRCY file contains requests that has been issued when Connect:Express was not started. The SYSRCY dsname is stored in memory after a $LOAD$ command has been issued. This DSNAME must be the same as the one specified in monitor JCL. System action: The task is terminated. User action: In order to change the dsname of SYSRCY a $LOAD$ command must be issued before cold start the monitor.
dtsm207E	xxxxxxxx ERROR DURING yyyyyyyy FUNCTION Explanation: The function specified by yyyyyyyy has failed within program xxxxxxxx. System action: See other messages. User action: Call your PROGRAMMING SUPPORT. If hot start SYSCHK/SYSRCY 'READ' error or SYSLOG 'WRITE' error after file has been migrated/freed, refer to the Installation Guide and complete step 13, Create the Monitor Environment. You only have to do this for the file that has the problem.
dtsm208I	SYSIN SUCCESSFULLY PROCESSED Explanation: The SYSIN is correct. System action: None. User action: None.
dtsm209I	??????? DUPLICATE ENTRY (IGNORED) Explanation: This entry is already present. System action: None. User action: Correct the name (file/partner).
dtsm210I	RACF USER/GROUP/O: ????????/????????/? Explanation: This information shows the RACF user/group assigned to this TOM STC/JOB and where from (Ascb to Tcb). System action: None. User action: Check if correct.
dtsm211W	REQ xxxxxxxx MARKED UNUSABLE (PURGE ONLY) Explanation: During hot start the monitor has detected an invalid request. System action: Request marked unusable. User action: Check the reasons. The request cannot be processed, only purge is authorized.
dtsm212W	uuu xxxxxxxx MARKED UNUSABLE Explanation: During hot or cold start, the monitor has detected an invalid file or partner entry. System action: The entry is marked unusable. User action: Check the reasons, modify the entry in order to change its status.

The 'dtsm' prefix is replaced by your 'C:X' subsystem name.

WTO Messages Issued by the Monitor (continued)

Code	Description
dtsm213W	PCT xxxxxxxx INVALID SLD ssssssss Explanation: During hot or cold start the monitor has detected an invalid partner definition. The SLD specified does not exist. System action: The entry is marked unusable. User action: Check the reasons, modify the entry in order to change its status. To refresh the SLD table the monitor must be restarted.
dtsm214W	PCT xxxxxxxx INVALID ALT.PART. ssssssss Explanation: The partner specified by ssssssss does not exist. System action: The entry is marked unusable. User action: Apply corrections and modify the entry in order to change its status.
dtsm215W	REQ xxxxxxxx PARTNER NOT FOUND ssssssss Explanation: The partner specified by ssssssss has been deleted, Connect:Express was not started. System action: The request is marked unusable. User action: Check the reasons and purge the request.
dtsm216W	REQ xxxxxxxx FILE NOT FOUND ssssssss Explanation: The file specified by ssssssss has been deleted, Connect:Express was not started. System action: The request is marked unusable. User action: Check the reasons and purge the request.
dtsm217W	REQ xxxxxxxx INVALID TQE CHAIN ssssssss Explanation: The request and its TQE are incompatible. System action: The request is marked unusable. User action: Purge the request.
dtsm218W	REQ xxxxxxxx SLD (d)/= PART ssssssss Explanation: The partner definition (sld) has been modified, Connect:Express was not started. System action: The request is marked unusable. User action: Purge the request.
dtsm219W	REQ xxxxxxxx PART_TYP (t)/= PART ssssssss Explanation: The partner definition (partner type) has been modified, Connect:Express was not started. System action: The request is marked unusable. User action: Purge the request.
dtsm220W	REQ xxxxxxxx INVALID SST ADDRESS Explanation: The request has an invalid pointer. System action: The request is marked unusable. User action: Purge the request.
dtsm221W	REQ xxxxxxxx INVALID SST PARTNER ADDRESS Explanation: The request has an invalid pointer. System action: The request is marked unusable. User action: Purge the request.
dtsm222W	REQ xxxxxxxx INVALID SST TYPE OF CALL Explanation: The request has an invalid status. System action: The request is marked unusable. User action: Purge the request.

The 'dtsm' prefix is replaced by your 'C:X' subsystem name.

WTO Messages Issued by the Monitor (continued)

Code	Description
dtsm223W	REQ xxxxxxxx MAY BE UNUSABLE Explanation: The monitor detected the request status is unexpected. This can occur because an UNLOAD or RELOAD JOB has been issued while Connect:Express was ending and the JOB is not finished when Connect:Express restarts. System action: The request is marked almost unusable. User action: Check the reasons, if a JOB has been issued the request will be processed at its end, otherwise purge the request.
dtsm224W	SYSxxx RECORD rrrrrrrr IS INVALID (iiii) Explanation: The monitor has detected an invalid record within SYSSNA or SYSX25 file: "rrrrrrr" indicates the record number and "iiii" the reason. System action: The record is not processed. User action: Modify the specified record and issue a refresh command.
dtsm225W	PCT xxxxxxxx INVALID PROTOCOL / LINK Explanation: The specified partner has an invalid link specified according to its protocol. System action: The entry is marked unusable. User action: Modify the entry in order to change its status.
dtsm226W	REQ xxxxxxxx LINK (l) FORCED TO (y) Explanation: The partner of the request has been modified (link definition), Connect:Express was not started. System action: The "link" (l) specified in the request has been changed into "link" (y) (first link of the partner). User action: Check the reasons.
dtsm227W	FCT xxxxxxxx INVALID PARTN. LIST yyyyyyyy Explanation: The partner list specified in the definition of the file has not been found. System action: The file is marked unusable. User action: Apply the corrections, modify the file entry and issue the command "ACF=xxxxxxxx"
dtsm228W	??? xxxxxxxx INVALID CHARACTER IN NAME Explanation: The entry specified has an invalid character as ' ' or '-' in position one, or not EBCDIC printable char. System action: Entry is marked unusable. User action: Delete it, then type correct name.
dtsm230W	REQUEST COUNT HAS BEEN ADJUSTED Explanation: Depending on how the monitor has been stopped the request count can be false. System action: The request count is adjusted. User action: None
dtsm231E	SYS??? PROCESS ERROR RC=?? Explanation: During SNA/X25/TCP extension process, an error has bee detected. System action: Connect:Express stops. User action: Correct the related file, and restart.
dtsm232I	???????? NOT USED Explanation: The mentioned DD card was not used in the procedure. System action: None User action: None
dtsm233I	T1B2PTCP RELOADED BY OPERATOR Explanation: Operator command refreshed the table. System action: The table is refreshed totally/partially. User action: Check for rc=00, else see error messages then correct and retry.

The 'dtsm' prefix is replaced by your 'C:X' subsystem name.

WTO Messages Issued by the Monitor (continued)

Code	Description
dtsm240I	CXPLEX SUCCESSFULLY PROCESSED Explanation: The CXPLEX is correct System action: None User action: None
dtsm241E†	ERROR DETECTED IN THE CXPLEX FILE Explanation: An error had been detected while processing the CXPLEX file. System action: Connect:Express stops initialization. User action: See next message.
dtsm242E†	?????? INVALID CXPLEX CARD VALUE L=??? ?? Explanation: An invalid CXPLEX card value has been detected. System action: Connect:Express stops initialization. User action: Correct this value about line = ????
† Refer to the Administration Guide for a description of the CXPLEX file.	
dtsm243E†	?????? INVALID CXPLEX KEYWORD L=??? Explanation: An invalid CXPLEX card value has been detected. System action: Connect:Express stops initialization. User action: Correct this value about line = ????
dtsm244E†	?????? TOO MANY/INV.SEQ.CXPLEX CARD L=??? Explanation: An invalid CXPLEX card value has been detected. System action: Connect:Express stops initialization. User action: Correct this value about line = ????
dtsm245E†	?????? ?????? MISSING CXPLEX CARD Explanation: An invalid CXPLEX card value has been detected. System action: Connect:Express stops initialization. User action: Add the missing card.
dtsm251E‡	?????? INVALID SYSIN CARD VALUE (RUN=H) Explanation: An invalid sysin card value has been detected. System action: Connect:Express stops initialization. User action: Correct this value for a warm start (H), or perform a cold start (C).
dtsm252E‡	?????? INVALID SYSIN CARD VALUE L=??? Explanation: An invalid sysin card value has been detected. System action: Connect:Express stops initialization. User action: Correct this value about line ???.
dtsm253E‡	?????? INVALID SYSIN CARD (KEYWORD) L=??? Explanation: An invalid sysin card keyword has been detected. System action: Connect:Express stops initialization. User action: Correct this keyword about line ???.
dtsm254E‡	?????? TOO MANY/INV.SEQ. SYSIN CARD L=??? Explanation: An invalid sysin condition has been detected. System action: Connect:Express stops initialization. User action: Correct this condition about line ???, whether too many cards, whether invalid sequence.
dtsm255E‡	?????? MISSING SYSIN CARD Explanation: An invalid sysin condition has been detected. System action: Connect:Express stops initialization. User action: Add the missing card.

The 'dtsm' prefix is replaced by your 'C:X' subsystem name.

WTO Messages Issued by the Monitor (continued)

Code	Description
dtsm260I	AFMCLS=(X/XXX/XXX/XXX): COMMAND REJECTED Explanation: Changing AFM class parameter for FTP transfers is rejected. System action: None User action: check the syntax of the command and retry.
dtsm261I	AFMCLS=(X/XXX/XXX/XXX):COMMAND ACCEPTED Explanation: Changing AFM class parameter for FTP transfers accepted. System action: The FTP scheduling process is changed rules. User action: None
dtsm262I	AFM START COMMAND FAILED RC=???? Explanation: The AFM initialization process failed. System action**:** The task is terminated. User action**:** Check the JCL of the AFM and the AFM parameter file.
†	Refer to the Administration Guide for a description of the CXPLEX file.
‡	Refer to Appendix D of the Administration Guide, *Initialization Parameters*, for a description of the SYSIN file.
dtsm263I	AFM SUCCESSFULLY STARTED ASID=???? Explanation**:** The FTP manager has been successfully started. System action: FTP service is enabled. User action: None
dtsm264I	TOM LAST UPDATE DATE-TIME : 010701-0808 Explanation: This is the last update of the product. System action: None. User action: This information may be checked by support.
dtsm265E	FREEMAIN ERROR DETECTED FOR TABLE: ??? Explanation: CSA storage could not be freed System action: Termination processed. User action: Keep information and call programming support.
dtsm270E	JOIN XCF FAILED RC=?? Explanation: The standby could not Join the XCFGROUP. System action: The process is stopped. Recovery functions are disabled. User action: Check return code and environment.
dtsm271I	COMMUNICATION OPENED WITH MANAGER ??????? Explanation: a SYSPLEX server connected to the manager. System action: the API is active on the corresponding system. User action: None
dtsm272I	MEMBER $MGTOM1$ $$SYSB$$ CONNECTED Explanation: Manager TOM1, executes on the z/OS image SYSB, and joined the group System action: Manager TOM1 is activated, message 317 is issued. User action: None.
dtsm273I	STANDBY ???????? CONNECTED ???????? Explanation: The standby process connected to the XCF group. System action: Extended Recovery is active for the related component User action: None
dtsm274I	MEMBER $MGTOM1$ TERMINATED Explanation: Member $MGTOM1$ left the group. System action: None. User action: Check the reasons for the member to leave.

The 'dtsm' prefix is replaced by your 'C:X' subsystem name.

WTO Messages Issued by the Monitor (continued)

Code	Description
dtsm275E	SERVER ???????? ABNORMALY TERMINATED Explanation: A SYSPLEX server disconnected from the manager. System action: Service is disabled for the corresponding system. User action: Check the server SYSOUT.
dtsm276I	SERVER ACKNOWLEDGES STOP COMMAND Explanation: A SYSPLEX server is notified to STOP. System action: This server terminates. User action: None
dtsm277I	STANDBY INITIALIZATION COMPLETE Explanation: A standby process initialized successfully. System action: Extended Recovery is active for the related component User action: None
dtsm278I	STANDBY ACKNOWLEDGES RECOVERY COMMAND Explanation: The Standby is taking over after the related component abnormally terminated. System action: The component is reinitialized and new standby started. User action: Check the component abnormal termination.
dtsm279I	STANDBY ACKNOWLEDGES STOP COMMAND Explanation: The standby is notified to stop. System action: Process is normally stopped. User action: None
dtsm280I	STANDBY ENDED CMP: S???/U??? Explanation: The standby termination is complete. System action: None User action: None
dtsm281E	STANDBY ABNORMALY TERMINATED Explanation: The standby process was terminated abnormally. System action: The Extended Recovery is disabled for the corresponding component. User action: Check the abnormal termination.
dtsm282I	START SERVER ???????? ???????? Explanation: A SYSPLEX server is started. System action: The start procedure is launched. User action: None
dtsm283I	START STANDBY ???????? ???????? Explanation: The standby is started System action: The start procedure is launched. User action: None
dtsm284E	START MANAGER TOMJOB1 SYSB Explanation : The supervisor started procedure TOMJOB1 on the z/OS image SYSB from its configuration file - MANAGR=(SYSB,TOMJOB1),SSN=TOM1. System action: The manager initializes and join the group. User action: None.
dtsm285E	START MEMBER TOMJOB1 SYSB FAILED Explanation: The supervisor or a manager failed to start a member on the z/OS image SYSB. The name of the start command is TOMJOB1. System action : None. User action: Check messages in the system log, Contact Sterling Commerce Customer Support.

The 'dtsm' prefix is replaced by your 'C:X' subsystem name.

WTO Messages Issued by the Monitor (continued)

Code	Description
dtsm286I	START STANDBY TOMSTBY SYSA FAILED Explanation : The supervisor or a manager failed to start the standby on the z/OS image SYSA. The name of the start command is TOMSTB1. System action : None. User action: Check messages in the system log, contact Support.
dtsm287I	SEND COMMAND TO TOMJOB1 SYSB Explanation: The supervisor transmitted an MVS command to manager TOMJOB1 that is execuitng on the z/OS image SYSB. System action: The command is routed to SYSB and passed to TOMJOB1. User action: None.
dtsm288E	$SPTOMX$ REJECTED, SSN CHANGED TOMP->TOMX Explanation: The manager received a join indication from a supervisor different from the one that initialized it. System action: The supervisor is abended (Abend 00C). User action: Check the configuration.
dtsm289E	UNKNOWN MANAGER JOINING $MGTOMX$ $$SYSB$$ Explanation: The supervisor received a join indication from a manager that is not defined in its configuration file. System action: The manager is abended (abend 00C). User action: Check the configuration.
dtsm290E	SUPERVISOR ALREADY UP: $SPTOMX$ ABENDED Explanation: A supervisor joined a group that is already under supervision . System action: The second supervisor is abended (Abend 00C). User action: Check the configuration.
dtsm291I	AP DETECTED UNAUTH PROGRAM MODIFICATIONS Explanation: The AP key file has been modified System action: Initialization is terminated User action: Get valid AP key file from Sterling Commerce
dtsm292I	AP KEY NOT FOR THIS SYSTEM, ???????? Explanation: The AP key file is not for OS/390 System action: Initialization is terminated User action: Get valid AP key file from Sterling Commerce
dtsm293W	AP KEY FILE NOT VALID FOR ?? DAY(S) Explanation: The AP key file will not be valid for ?? days System action: Initialization is terminated User action: Get valid AP key file from Sterling Commerce
dtsm294W	AP KEY OPTION HAS EXPIRED, ???????? Explanation: The specified option has expired System action: Processing continues User action: Get valid AP key file from Sterling Commerce
dtsm295W	WARNING AP OPTION ???????? HAS EXPIRED Explanation: The specified option has expired System action: Processing continues User action: Get valid AP key file from Sterling Commerce
dtsm296W	AP OPTION EXPIRES IN ?? DAY(S) ???????? Explanation: The specified option is about to expire System action: Processing continues User action: Contact Sterling Commerce to update your AP key file

The 'dtsm' prefix is replaced by your 'C:X' subsystem name.

WTO Messages Issued by the Monitor (continued)

Code	Description
dtsm297W	AP KEY IS NOT VALID Explanation: The AP key file is corrupt System action: Initialization is terminated User action: Get valid AP key file from Sterling Commerce
dtsm298W	AP KEY HAS EXPIRED Explanation: The AP key file has expired System action: Initialization is terminated User action: Contact Sterling Commerce for a valid AP key file
dtsm299W	WARNING AP KEY HAS EXPIRED Explanation: The AP key file has expired System action: Processing continues User action: Contact Sterling Commerce for a valid AP key file
Dtsm300W	AP KEY EXPIRES IN ?? DAY(S) ???????? Explanation: The AP key file is about to expire System action: Processing continues User action: Contact Sterling Commerce for a valid AP key file
Dtsm301I	UNABLE TO INITIALIZE AP SYSTEM Explanation: The Asset Protection (AP) system could not be initialized System action: Processing terminates User action: Contact Sterling Commerce Customer Support
Dtsm302W	AP KEY NOT VALID Explanation: The AP key file is corrupt System action: Initialization is terminated User action: Get valid AP key file from Sterling Commerce
Dtsm303I	INITIALIZED USING EMERGENCY KEY Explanation: A temporary emergency key has been found System action: Processing continues User action: Contact Sterling Commerce for a valid AP key file
Dtsm304I	REFRESHING ASSET PROTECTION VALUES Explanation: The asset protection values are being refreshed System action: Processing continues User action: None
Dtsm305I	ASSET PROTECTION FAILURE Explanation: The asset protection (AP) system has failed System action: Processing is terminated User action: Contact Sterling Commerce Customer Support
Dtsm306I	NUMBER OF LICENSED COPIES EXCEEDED Explanation: The number of licensed copies has been exceeded System action: Processing is terminated User action: Get a valid AP key file from Sterling Commerce
Dtsm307I	AP_STRG ALLOCATION FAILURE Explanation: The asset protection (AP) system had a storage failure System action: Processing is terminated User action: Contact Sterling Commerce Customer Support
Dtsm308I	AP SYSTEM STORAGE ALLOCATION FAILURE Explanation: The asset protection (AP) system had a storage failure System action: Processing is terminated User action: Contact Sterling Commerce Customer Support

The 'dtsm' prefix is replaced by your 'C:X' subsystem name.

WTO Messages Issued by the Monitor (continued)

Code	Description
Dtsm309I	AP KEY FILE ALLOCATION ERROR Explanation: The AP key file could not be allocated System action: Processing is terminated User action: Check initialization parameters for valid AP key file name
Dtsm310I	NUMBER OF SERVERS EXCEEDED Explanation: The number of TOM servers has been exceeded System action: Processing is terminated User action: Get valid AP key file from Sterling Commerce
Dtsm311W	AP ERROR, ERRFUNC ???????? RC=???????? Explanation: An internal error occurred in asset protection processing System action: Processing is terminated User action: Contact Sterling Commerce Customer Support
Dtsm312W	ERROR TXT=???????????????????????????????? Explanation: An internal error occurred in asset protection processing System action: Processing is terminated User action: Contact Sterling Commerce Customer Support
Dtsm315I	TOM ACKNOWLEDGES SHUTDOWN COMMAND Explanation: The supervisor received a SHUTDOWN command, either for all or for a specific manager. System action: The supervisor stops all managers and terminates or stops the specified manager. User action: None.
Dtsm316W	MANAGER TOM1 IS NOT ACTIVE Explanation: The supervisor received the command SHUTDOWN=TOM1 and manager TOM1 is inactive or unknown. System action: The supervisor rejects the command. User action: Check the command or the status of sub system TOM1.
Dtsm317I	REQUEST NUMBERS FROM 00000001 TO 33333333 Explanation: This message follows message 326 at the end of initialization of a manager or the supervisor, or message 264-320 at the end of initialization of the supervisor when it participates to transfers. This message follows message 272 each time the supervisor accepts the connection of a manager. The sub system associated to this message uses this request range. System action: Each componant uses the request range that the supervisor allocated. User action: None.
Dtsm318E	NO MANAGERS DEFINED FOR THIS SUPERVISOR Explanation: While initializing the supervisor found no MANAGR= card in its CXPLEX file. System action: The supervisor terminates. User action: Check the configuration, correct, and retry
Dtsm319I	FILE TRANSFER FUNCTIONS ARE INACTIVE Explanation: XFRTYP=NO in the supervisor CXPLEX file. System action: Network resources (ANM) and transfer resources (APM et AFM) are inactive. The supervisor rejects all local transfer requests. User action: None.
Dtsm320I	ONLY HOLD REQUESTS ARE SUPPORTED Explanation: XFRTYP=HOLD in the supervisor CXPLEX file. System action: The supervisor accepts only local requests of type Hold, and records them in the HCT. User action: None.

The 'dtsm' prefix is replaced by your 'C:X' subsystem name.

WTO Messages Issued by the Monitor (continued)

Code	Description
Dtsm321I	???????? IXCQUERY ERROR RC/RS=????/???? Explanation: An IXCQUERY error occurred System action: Processing is terminated User action: Contact Sterling Commerce Customer Support
Dtsm322I	???????? IXCJOIN ERROR RC/RS=????/???? Explanation: An IXCJOIN error occurred System action: Processing is terminated User action: Contact Sterling Commerce Customer Support
Dtsm323I	???????? IXCLEAVE ERROR RC/RS=????/???? Explanation: An IXCLEAVE error occurred System action: Processing is terminated User action: Contact Sterling Commerce Customer Support
Dtsm324I	GROUPP SUPERVISOR STANDBY INITIATED Explanation: The standby of the supervisor of group GROUPP is initialized. System action: Processing continues. User action: None.
Dtsm325I	GROUPP $SVTOMP$ STANDBY INITIATED Explanation: The standby of member $SVTOMP$ in the group GROUPP is initialized. System action: Processing continues. User Action: None
Dtsm326I	COMMUNICATION OPENED WITH SUPER $SPTOMP$ Explanation: Communication is established with supervisor TOMP. System action: Manager status is set to active. User action: None.
Dtsm327W	COMMUNICATION CLOSED WITH SUPER $SPTOMP$ Explanation: Supervisor TOMP terminated. System action: The manager becomes a standalone manager. User action: Restart the supervisor.
Dtsm328I	SUPERVISOR $SPTOMP$ IS UP: INIT. STOPPED Explanation: While initializing the supervisor detects that another supervisor is already started. System action: The supervisor terminates. User action: Check the configuration.
Dtsm329E	JOIN FAILED WITH $MGTOM1$ - DTSM288E Explanation: Manager $MGTOM1$ rejected the first contact with the supervisor. The reason is specified by the message number, DTSM288E in this example. System action: The member that is joining the group terminates. User action: Check the configuration.
Dtsm330I	PAUSE MEMBER JOBTOM1 SYSB Explanation: The supervisor received a command to shutdown a manager. The name of the procedure is JOBTOM1 for this manager and it is executing on z/OS image SYSB. System action: The pause command is routed to the z/OS image SYSB. User action: None.
Dtsm331E	LOST CONTACT WITH MANAGER, GROUP ???????? Explanation: A TOM server has lost contact with the TOM manager System action: The TOM server is terminated User action: Contact Sterling Commerce Customer Support

The 'dtsm' prefix is replaced by your 'C:X' subsystem name.

WTO Messages Issued by the Monitor (continued)

Code	Description
Dtsm332E	GROUPP MEMBER LOST: $MGTOM1$$$SYSB$$ Explanation: Member TOM1 of GROUPP lost communication. System action: Member $MGTOM1$ is set to inactive. User action: Contact Sterling Commerce Customer Support.
Dtsm333E	LOST CONTACT WITH $MGTOM1$ MANAGER Explanation: A TOM server has lost contact with the TOM manager. System action: The server terminates. User action: Contact Sterling Commerce Customer Support.
Dtsm334I	SEVERE XCF COMMUNICATIONS ERROR Explanation**:** A severe XCF error occurred System action: Processing is terminated User action**:** Contact Sterling Commerce Customer Support
Dtsm335I	REQUEST ABENDED ON TARGET Explanation**:** A request has abended on the receiving side System action: The request is terminated User action: Retry the request. Contact Sterling Commerce Customer Support
Dtsm336I	SERVER LEFT XCF GROUP Explanation**:** A TOM server has been terminated System action: Processing continues User action: None
Dtsm337I	XCF REQUEST TIMED-OUT Explanation**:** An attempt to communicate using XCF has timed-out System action: Message Dtsm331E or Dtsm332I is issued User action: See appropriate message
Dtsm338I	XCF MCIA AREA CLEARED TO ZERO Explanation**:** A null MCIA area was received System action: Processing is terminated User action: Contact Sterling Commerce Customer Support
Dtsm339I	IXCMSGO ERROR Explanation**:** An error occurred when trying to communicate using XCF System action: Processing is terminated User action: Contact Sterling Commerce Customer Support
Dtsm340I	M1B2XCF$ MACRO SUCCESSFUL Explanation**:** Informational only System action: Processing continues User action: None
Dtsm341I	NUMBER-SERVERS EXCEEDED Explanation: The number of TOM servers has been exceeded System action: The TOM server is terminated User action: Get a valid AP key file from Sterling Commerce
Dtsm342I	NO EMPTY REQUEST QUEUE FOUND Explanation: This is an internal error System action: Processing is terminated User action: Contact Sterling Commerce Customer Support
Dtsm343I	CANNOT FIND MEMBER D1B2PXCT Explanation**:** This is an internal error System action: Processing is terminated User action: Contact Sterling Commerce Customer Support

The 'dtsm' prefix is replaced by your 'C:X' subsystem name.

WTO Messages Issued by the Monitor (continued)

Code	Description
Dtsm344I	TERMINATING STANDBY Explanation: TOM manager or server is terminating its standby System action: Processing continues User action: None
Dtsm345I	TERMINATING. EXPECT ABEND U1025 00000000 Explanation**:** The XCF member has been terminated System action: Processing is terminated with ABEND U1025 User action: None. Look for other messages for cause.
Dtsm346I	TERMINATING. EXPECT ABEND U1027 00000000 Explanation**:** The XCF member has been terminated System action: Processing is terminated with ABEND U1027 User action: None. Look for other message for cause
Dtsm347I	INSUFFICIENT STORAGE FOR ???????? Explanation: Insufficient storage is available for a request System action: Processing is terminated User action: Increase region size and retry
Dtsm350E	SUPERVISOR XCT ENTRY NOT FOUND Explanation: The manager did not find the supervisor XCT entry. System action: The manager rejects the supervisor request. User action: Contact Sterling Commerce Customer Support.
Dtsm351E	$SPTOMP$ REJECTED, REQUEST RANGE CHANGED Explanation: The manager detects that the request range proposed by the supervisor during initialization is different than the current request range. System action: The manager terminates. User action: Check the configuration of the supervisor.
Dtsm352E	HOT RESTART: CXPLEX CONFIG CHANGED Explanation: The supervisor is hot started with a new configuration. System action: Message 011 is issued, the supervisor terminates. User action: Cold start the supervisor or change back the configuration.
Dtsm354E	15000000 INVALID INIT REQUEST NUMBER Explanation : The monitor (manager or supervisor) is initializing with the EXEC parm REQ=15000000, and this value is out of the request range. System action: The monitor termminates. User action: Change parameter REQ=.
Dtsm355I	STARTING REQUEST NUMBER AT 01500000 Explanation: The monitor (manager or supervisor) is initializing with the EXEC parm REQ=15000000. System action: The first request number will be this value. User action: None.
Dtsm356W	MANAGER TOMP IS ACTIVE Explanation: The supervisor received the command LAUNCH=TOMP, but TOMP is already active. System action: The command is ignored. User action: Check the reason for sending this command.
Dtsm357W	MANAGER TOMP IS UNKNOWN Explication: The supervisor received the command LAUNCH=TOMP, but TOMP is not defined in the CXPLEX file. System action: The command is ignored. User action: Check the reason for sending this command.

The 'dtsm' prefix is replaced by your 'C:X' subsystem name.

WTO Messages Issued by the Monitor (continued)

Code	Description
Dtsm358W	DTSM358W PCT UPDATE REJECTED BY $MGTOM2$ TRC=2030 Explanation: Manger TOM2 rejected the partner update. System action: The supervisor updates the partner definition locally. User action: Check the TRC code and update again.
Dtsm389E	INVALID CONFIGURATION FILES Explanation: During initialization, the manager detects that the configuration files are different from those of the supervisor. System action: The manager terminates. User action: Check configuration files SYSFIL, SYSPAR, SYSSNA, SYSX25 et SYSTCP.

The 'dtsm' prefix is replaced by your 'C:X' subsystem name.

SYSLOG Messages Issued by the Monitor

SYSLOG messages are text messages and have no 'ID.' They are listed in this section by product/function. The table below lists symbols that may appear in SYSLOG messages:

Symbol(s)	Substitution
?	return code
c	class
cccc	CPU-ID
d	direction
dddddd	date
dsn	data-set-name
ffffffff	symbolic file name
gggggggg	RACF group
hhhhhh	hour
jjjjjjjj	job name or user name
l	link type
lid	type, main, secondary link (x,mch,cv/...)
llllllll	lu-name
mmmm	C:X subsys
nn	number
nnnnnnnn nnnnnnnn	NRC
o	option
pp	percent
pppp	PRC
pppppppp	symbolic partner name
rrrrrrrr	request number

Symbol(s)	Substitution
S???	system abend
ssss	SRC
tttt	TRC (L/R)
U???	user abend
uuuuuuuu	userid
xxxxxxxx	function

The following table lists SYSLOG messages issued by the Monitor by product/function.

SYSLOG Messages Issued by the Monitor

Message	Comment
START MESSAGES	

`TOM IS STARTING (xxxxxxxx)`
`START OF A.N.M. IN ASID: nnnnn RC=?`
`START OF A.P.M. nn IN ASID: nnnnn RC=?`
`START OF AF.M REQUESTED BY TOM`
`TOM INITIALIZATION COMPLETE V410-PTF2-5 dddddd-hhhh RUN=H`
	The update level information may be requested by support.

`AFM INITIALIZATION COMPLETE ASID=nnnnn PORT=nnnnn SESSION (nnnn1/nnnn2)`	The AFM is listening incoming calls on PORT=nnnnn, nnn1 sessions input and nnnn2 sessions output are available.
`PARAMETERS OF JOB: jjjjjjjj RUN=o CPU=xxxx` `PLEX PARAMETERS OF JOB: jjjjjjjj`	The SYSIN and CXPLEX files have already been listed.
`RESTART PROCEEDING` `RESTART SUCCESSFUL`	Connect:Express processed the Recovery file in which file transfer requests are recorded while the monitor was not up.
Connection ~~With TOM Local	

`CONNECTION ESTABLISHED WITH LOCAL: llllllll`
`CONNECTION CLOSED WITH LOCAL: llllllll`
`CONNECTION FAILED WITH LOCAL: llllllll NRC=nnnnnn nnnnnnn`
	Check the SNA link between Local and Global (TOMACB= parameter in the SYSIN file).

REQUEST MESSAGES	
Normal Request	

SYSLOG Messages Issued by the Monitor (continued)

Message	Comment
REQUEST rrrrrrrr <- ffffffff ppppppppp (d) SRC=0000 TRC=0000 PRC=0000 REQUEST rrrrrrrr <- jjjjjjjj ACCEPTED (o) AFM INCOMING FTP SESSION (nn/nn) OPENED WITH ppppppppp (xxx.xxx.xxx.xxx) AFM OUTGOING FTP SESSION (nn/nn) OPENED WITH ppppppppp (xxx.xxx.xxx.xxx) REQUEST rrrrrrrr COMMUNICATION OPENED (d) WITH ppppppppp lid REQUEST rrrrrrrr ffffffff TRANSFER ACCEPTED APM nn EFF nn REQUEST rrrrrrrr HSM AUTOMATIC RECALL ffffffff SRC=ssss REQUEST rrrrrrrr HSM AUTOMATIC RECALL PROCESSING FOR DSN dsn REQUEST rrrrrrrr/ffffffff/ppppppppp JOBnnnnn UNLOAD STARTED REQUEST rrrrrrrr/ffffffff/ppppppppp JOBnnnnn UNLOAD ENDED REQUEST rrrrrrrr ffffffff TRANSFER STARTED APM nn EFF nn REQUEST rrrrrrrr ffffffff TRANSFER RESTARTING APM nn EFF nn REQUEST rrrrrrrr TRANSMITTING -> ppppppppp FILE ffffffff RECORDS nnnnnnn REQUEST rrrrrrrr RECEIVING <- ppppppppp FILE ffffffff RECORDS nnnnnnn REQUEST rrrrrrrr ffffffff ppppppppp SRC=ssss TRC=tttt PRC=pppp REQUEST rrrrrrrr ffffffff TRANSFER ENDED TRC=0000L APM nn EFF nn REQUEST rrrrrrrr/ffffffff/ppppppppp JOBnnnnn RELOAD STARTED REQUEST rrrrrrrr/ffffffff/ppppppppp JOBnnnnn RELOAD ENDED REQUEST rrrrrrrr JOB jjjjjjjj SUBMITTED	This is the normal sequence of messages, depending on the type of file. PU, VU, SU, UU files are processed with Unload job before transfer and Reload job after transfer.
End Transfer	
TRANSFER DURATION hhmmss, RESTART NUMBER: nnn ,NUMBER OF S/R: nnnnnnn COMMUNICATION CLOSED (d) WITH: ppppppppp APM nn EFF nn AFM INCOMING FTP SESSION (nn/nn) CLOSED WITH ppppppppp(xxx.xxx.xxx.xxx) TRC=???? AFM OUTGOING FTP SESSION (nn/nn) CLOSED WITH ppppppppp(xxx.xxx.xxx.xxx) TRC=???? SE 'ffffffff REQ rrrrrrrr TRANSFER ENDED',USER=(uuuuuuu),LOGON	The TSO user is notified only if the transfer is done
End Request	
REQUEST rrrrrrrr PURGED BY uuuuuuuu REQUEST rrrrrrrr <- uuuuuuuu COMPLETED REQUEST rrrrrrrr <- uuuuuuuu PURGED	A request can be purged either automatically at end of successful transfer or by the operator.
Abnormal Request	
INCOMING REQUEST REJECTED rrrrrrrr SRC=ssss TRC=tttt PRC=pppp	This message is generally issued when an input call is not resolved. For example an X25 call packet has been received and session interrupted.
TRACE ppppppppp : nnnnnnnn nn T=x G=xxxxx U=xxxxxx "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"	See Administration guide Chapter 9 (Tracking Incoming Connection errors): the trace facility has been activated. An incoming call was received for partner ppppppppp, X25 address nnnnnnnn, sub address nn ….and so on.

SYSLOG Messages Issued by the Monitor (continued)

Message	Comment
`REQUEST rrrrrrrr INVALID DESTINATION pppppppp`	The local Connect:Express is not well identified by the remote partner. The remote partner must change its definition or aliases must be used.

`REQUEST rrrrrrrr COMMUNICATION POSTPONED BY ppppppp,RETRY IN nn MIN lid`
`REQUEST rrrrrrrr COMMUNICATION NOT OBTAINED ppppppp,RETRY IN nn MIN lid`
`REQUEST rrrrrrrr RETRY WITH PARTNER pppppppp, NEW LINK : l`
`AFM OUTGOING FTP SESSION (nn/nn) POSTPONED WITH pppppppp (xxx.xxx.xxx.xxx)`
`RC=????? TRC=???? PRC=????`
`AFM OUTGOING FTP SESSION (nn/nn) NOT OBTAINED WITH pppppppp`
`(xxx.xxx.xxx.xxx) RC=????? TRC=???? PRC=????`
	The automatic retry process is running from STIMEV= and STIMOC= parameters declared in the SYSIN file. When all possible retries that are possible have been done (all alternate addresses failed for the current link), next link type is selected.

`REQUEST rrrrrrrr COMMUNICATION REJECTED BY pppppppp PRC=pppp lid`
`AFM INCOMING FTP SESSION (nn/nn) REJECTED WITH pppppppp`
`(xxx.xxx.xxx.xxx) TRC=????`
`AFM OUTGOING FTP SESSION (nn/nn) REJECTED WITH pppppppp (xxx.xxx.xxx.xxx)`
`RC=????? TRC=???? PRC=????`
`REQUEST rrrrrrrr COMMUNICATION ABORTED WITH pppppppp, PARTNER HELD`
	No automatic retry process is running because it has never been activated or the automatic retry process is stopped at the end of the possible links. if needed add the PRC received in T1B2PCOD, and ACR=YES in partner definition.

`REQUEST rrrrrrrr ABORT <- pppppppp SRC=ssss TRC=tttt PRC=pppp`
`REQUEST rrrrrrrr SESSION ERROR : AAAAAAAA NRC=nnnnnn nnnnnn`
`REQUEST rrrrrrrr SESSION DISCONNECTED SRC=ssss TRC=tttt PRC=pppp`
	These are different ways of cutting a session. A protocol message may abort the session or a network anomaly can occur (check the NRC codes).

`REQUEST rrrrrrrr NOT xxxxxxx SRC=ssss TRC=tttt PRC=pppp`	You probably passed a command that is not allowed by the status of this request (for example restarting a not interrupted request, purging a request during negotiation phase are not allowed).

`REQUEST rrrrrrrr <- ffffffff ppppppppp (d) SRC=ssss TRC=tttt PRC=pppp`
`REQUEST rrrrrrrr <- jjjjjjjj REJECTED`
`REQUEST rrrrrrrr WITH PARTNER pppppppp UNSUPPORTE D LINK : I`
`REQUEST rrrrrrrr ffffffff ERROR DURING SELECTION TRC=ttttL PRC=pppp`
`REQUEST rrrrrrrr REJECTED SRC=ssss TRC=ttttL PRC=pppp`
`REQUEST rrrrrrrr ffffffff SUSPENDED TRC=ttttL PRC=pppp`
`REQUEST rrrrrrrr ffffffff INTERPT./PURGED BY SERVER TRC=ttttL PRC=pppp`
`REQUEST rrrrrrrr ffffffff INTERPT./PURGED BY CALLER TRC=ttttL PRC=pppp`
`REQUEST rrrrrrrr ffffffff WILL BE RESTARTED TRC=ttttL PRC=pppp`
`REQUEST rrrrrrrr/ffffffff/pppppppp JOBnnnn TRANSFER INTERRUPTED (?)`
	If the trc is null, the problem has probably been detected on the remote side. If the TRC=ttttL this is a local error, if the TRC=ttttR this is a remote error.

`REQUEST rrrrrrrr NO AUTOMATIC RESTART BY T1B2PCOD, PRC=ppp 'protocol id'`	This message is issued if a connection or file selection error. If needed add this PRC in T1B2PCOD, and ACR=YES in the partner definition.

SYSLOG Messages Issued by the Monitor (continued)

Message	Comment
REQUEST rrrrrrrr API-CREATE: api-data	The API-CREATE field enables checking of API data sent/received if fail, see *MACLIB* D1B2RAPI for format (Origin – Destination – file name – date/time).
REQUEST rrrrrrrr/ffffffff/pppppppp JOBnnnnn ABEND S???/U??? RC=? REQUEST xxxx/rrrrrrrr/ffffffff(nnnn) PARTNER ABEND S???/U??? PHASE f REQUEST rrrrrrrr SESSION ABORTED APM nn ABENDED REQUEST rrrrrrrr PARTNER pppppppp FUNCTION xxxxxxxx FAILED TRC=tttt REQUEST rrrrrrrr pppppppp TRANSFER ABORTED SRC=ssss TRC=tttt PRC=pppp REQUEST rrrrrrrr AUTOMATIC COMMAND FORBIDDEN, UPRFCT=o	The command associated with file end/start transfer is forbidden by the UPRFCT option set in the SYSIN file.
AD HOC Incoming Notification	
SECURITY-ADHOC STAT RACFCN=o ADHOCN=o ffff DSN=dsn U=uuuuuuu G=ggggg	None
RACF Control Refused	
SECURITY xxxxx o r15/r0 uuuuuuuu gggggggg ffffffff dsn SECURITY-ADHOC INIT o r15/r0 uuuuuuuu gggggggg	This DSN was not found or found in error or mismatching
Allocation Control	
REQUEST rrrrrrrr ffffffff xxxxxxxx FAILED DSN=dsn REQUEST rrrrrrrr ffffffff LOC/CHECK FAILED DSN=dsn	This DSN was not found or found in error or mismatching
REQUEST rrrrrrrr ffffffff ALLOC FAILED DSN=dsn E..I..E..I..S...	E/I = DYNALLOC error/info primary+secondary codes, S = SMS error code
REQUEST rrrrrrrr ffffffff xxxxxxxx ADVISED DSN=dsn	An option is advised for this file, in file directory
REQUEST rrrrrrrr ffffffff DATA SET ENQUEUED BY uuuuuuuu cccc	Enqueue detected for the requested dsn
Auxiliary Protocol	
RJE RESTART --> $AJnnnn BSC APM: nn LINE: nn PROCESSING FAILED SRC=ssss TRC=tttt PRC=pppp REQUEST rrrrrrrr TED INTERRUPTION ?	None
Change	
rrrrrrrr ACTIVATED BY uuuuuuuu rrrrrrrr DEACTIVATED BY uuuuuuuu xCT: xxxxxxxx xxxxxxxx BY uuuuuuuu REQUEST rrrrrrrr PRTY c TO c CHANGED BY uuuuuuuu REQUEST rrrrrrrr SITDATE UPDATED uuuuuuuu J-2 yymmdd/J-1 yymmdd/J+0 yymmdd REQUEST rrrrrrrr INTERRUPTED BY uuuuuuuu INTERRUPT REQUEST TRANSMIT TO APM nn/n REQ: rrrrrrrr PURGE REQUEST TRANSMIT TO APM nn/n REQ: rrrrrrrr PARTNER pppppppp TRACE ON BY uuuuuuuu REQUEST rrrrrrrr NOT FOUND	None

SYSLOG Messages Issued by the Monitor (continued)

Message	Comment
Anomaly	
`REQUEST rrrrrrrr AUTOMATIC COMMAND NOT RESOLVED`	Check the start or end command defined for the file.
`LOCAL REQUEST QUEUE RCT pp % FULL` `EXTERNAL REQUEST QUEUE RCT pp % FULL`	The RCT size is define by the RQEMAX= parameter of the SYSIN file.
`ANM HANDLER ABNORMALY TERMINATED SRC=ssss TRC=tttt PRC=pppp`	Check the return codes. Check the network resource involved, and restart the handler when the problem is fixed.
`AFM INITIALIZATION FAILED  ASID=nnnnn  RC=nnnnn`	Check the return code. Check the configuration (Started task, TCP/IP, RACF …)
`STOP OF A.P.M nn REQUESTED BY TOM` `ABEND IN L1B2Pnnn S???/U???`	Keep the dump and call support.
End Connect:Express	

`TOM (GLOBAL) TERMINATION COMPLETE V410-PTF2 RUN=(o)`
`TOM IS ENDING NORMALLY (xxxxxxxx) ----------`
`STOP  OF A.F.M.  REQUESTED BY TOM`
`AFM TERMINATION  COMPLETE ASID=xxxxx  RC=?  ?`
	None

Local Connect:Express Messages	

`TOM (LOCAL) INITIALIZATION COMPLETE V410-PTF2-5`
`CONNECTION ESTABLISHED WITH GLOBAL: llllllll`
`L1B2P046 INITIALIZATION FAILED NRC=nnnnnnnn`
`CONNECTION CLOSED WITH GLOBAL: llllllll`
`CONNECTION FAILED WITH GLOBAL: llllllll`
`rrrrrrrr FROM uuuuuuuu TRANSMITTED TO GLOBAL mmmm,WAITING FOR RESP.`
`rrrrrrrr FROM uuuuuuuu NOT TRANSMITTED TO GLOBAL mmmm TRC=tttt`
`x xxxxxxxx COMMAND EXECUTED`
`x xxxxxxxx COMMAND NOT EXECUTED`
`REQUEST nnnnnnnn CLAS c TO c CHANGED BY uuuuuuuu`
`TOM (LOCAL) TERMINATION COMPLETE V410-PTF2`
	None

WTO Messages Issued by the APM

WTO Messages Issued by the APM

Code	Description
APM001I	CVT ALLOCATION ERROR RC=?? Explanation: The APM initialization cannot complete successfully. System action: rc="rc", the task is terminated after message: APM005I ERROR DURING INITIALIZATION -> STOP User action: Call support.

WTO Messages Issued by the APM (continued)

Code	Description
APM002I	ERROR DETECTED IN PARM OF EXEC Explanation: The APM initialization cannot complete successfully. System action: rc="rc", the task is terminated after message: APM005I ERROR DURING INITIALIZATION -> STOP User action: Check the PARM of EXEC of the failing APM, modify and start again.
APM003I	..... DD STATEMENT MISSING Explanation: The APM needs this work file. System action: The task is terminated after message: APM005I ERROR DURING INITIALIZATION -> STOP User action: Check the JCL of the failing APM, modify, start again.
APM005I	ERROR DURING INITIALIZATION -> STOP Explanation: The APM initialization cannot complete successfully. System action: The task is terminated. User action: Check previous message.
APM006I	SSCVT CHAIN IS INVALID Explanation: The APM initialization cannot complete successfully. System action: The task is terminated after message: APM005I ERROR DURING INITIALIZATION -> STOP User action: Call support.
APM007I	ssss NOT DEFINED AS A SUB-SYSTEM Explanation: The APM initialization cannot complete successfully. System action: The task is terminated after message: APM005I ERROR DURING INITIALIZATION -> STOP User action: Call support.
APM008I	xxxx NOT INITIALIZED Explanation: The APM initialization cannot complete successfully. System action: The task is terminated after message: APM005I ERROR DURING INITIALIZATION -> STOP User action: Call support.
APM010I	GETMAIN ERROR RC=xxx Explanation: The APM initialization cannot complete successfully. System action: rc="rc", the task is terminated after message: APM005I ERROR DURING INITIALIZATION -> STOP User action: Increase region size and if not ok call support.
APM012I	ERROR DURING ATTACH EFF. :.. RC=.. Explanation: The APM cannot attach an EFFECTOR. System action: rc="rc", the transfer is interrupted. User action: Call support.
APM016I	APM xx INITIALIZATION COMPLETE RC=00 Explanation: APM number xx successfully initialized. System action: None. User action: None.
APM017I	EFFECTOR: xx INITIALIZATION COMPLETE Explanation: Effector number xx successfully initialized. System action: None. User action: None.

WTO Messages Issued by the APM (continued)

Code	Description
APM018I	EFFECTOR: xx REQ. rrrrrrrr Ssss/Uuuu Explanation: An abend SYSTEM/USER occurred during transfer. System action: rc="S.../U...", the transfer is interrupted. The transfer resource is restored. User action: Check the abend code. Be aware that if any user exit is activated and an abend occurs in it, this message is issued.
APM019I	EFFECTOR: xx TERMINATION COMPLETE Explanation: Effector number xx successfully terminated. System action: None. User action: None.
APM020I	SP:xx/EF: xx REQ: rrrrrrrr Ssss/Uuuu Explanation: An abend occurred during transfer; session protocol is SP, effector is EF, request number is REQ. System action: rc="S.../U...", the transfer is interrupted. The transfer resource is restored. User action: Check the abend code. Be aware that if any user exit is activated and an abend occurs in it, this message is issued.
APM021I	EFFECTOR: xx PROTOCOL ETEBAC1/2 STARTED Explanation: Effector ETEBAC number xx successfully initialized. System action: None. User action: None.
APM022I	EFFECTOR: xx REQ: rrrrrrrr,PR pp STARTED Explanation: The request number REQ is allocated the effector xx, working with session protocol pp. System action: None. User action: None.
APM024I	EFFECTOR: xx REQ: rrrrrrrr,PR pp ENDED Explanation: Effector number xx successfully terminated. System action: None. User action: None.
APM024I	EFFECTOR: xx PROTOCOL ETEBAC1/2 ENDED Explanation: Effector ETEBAC number xx successfully terminated. System action: None. User action: None.
APM026I	EFFECTOR: xx ----> ABNORMAL TERMINATION Explanation: Effector stopped because of abend. System action: Effector becomes unavailable. User action: Check corresponding SYSUDUMP and correct. If all effector are in abend, stop and restart this APM
APM026I	COMMUNICATION OPENED WITH OPERATOR Explanation: APM available for command. System action: None. User action: None.
APM027I	INVALID COMMAND Explanation: APM received an invalid command. System action: None. User action: Correct, and retry.
APM028I	APM xx ALREADY IN THIS STATUS Explanation: APM received an already processed command. System action: None. User action: None.

WTO Messages Issued by the APM (continued)

Code	Description
APM029I	APM xx IS IN MODE : STOP Explanation: APM ETEBAC1/2 received a STOP command. If any transfer is active, a two minutes timer is activated and the active effector will be detached when expired. System action: None. User action: None.
APM031I	APM xx ACKNOWLEDGES IMMEDIAT STOP Explanation: APM received a stop command. System action: The APM will stop. User action: None.
APM036I	OPTION NOT AUTHORIZED RC:2-ttt Explanation: An executing transfer needs a C:X OPTION module the customer is not authorized to use. System action: TRC given indicates the option needed. The transfer is interrupted. User action: Check the TRC code. Contact the support for AUTHORIZATION key.
APM037I	APM xx INIT REPORT REJECTED BY C:X :2-xxx Explanation: A BSC APM is rejected by Connect:Express at initialization. System action: TRC given indicates the reason. The task is terminated after message: APM005I ERROR DURING INITIALIZATION -> STOP User action: Check the TRC code. Check DAPMxx= parameters in C:X SYSIN and APM= in PARM of EXEC of BSC APM JCL.
APM038I	APM xx END REPORT REJECTED BY C:X :2-ttt Explanation: The APM cannot advise Connect:Express of its stop. System action: 2-ttt="trc", the stop is incomplete. User action: Call support.
APM039I	APM xx ABNORMAL TERMINATION Ssss/Uuuu Explanation: The APM cannot stop normally. System action: Ssss ="system-abend", Uuuu ="user-abend". User action: Call support.
APM040I	APM xx TERMINATION COMPLETE Explanation: APM number xx successfully ended. System action: None. User action: None.
APM042I	uuuuuuuu USER EXIT INVOKED Explanation: The user exit server was invoked successfully and the parameter created using the T1APMSRT table. System action: None. User action: None.
APM043I	uuuuuuuu USER EXIT ERROR RC:.... Explanation: The 'uuuuuuuu' user exit server returned R15=RC. System action: RC is reported in TRC given by Connect:Express. The transfer is interrupted. User action: Check the TRC and PRC codes. Check transfer specifications.
APM044I	uuuuuuuu USER EXIT DISABLED Explanation: The 'uuuuuuuu' user application server returned error at initialization or abended. System action: The exit will be no more invoked. User action: Check the exit, modify, stop APM and start it again.

WTO Messages Issued by the APM (continued)

Code	Description
APM045I	L1APMSRV INITIALIZED Explanation: User exit application driver l1APMSRV is initialized. System action: None. User action: None.
APM046I	L1APMSRV TERMINATED Explanation: User exit application driver L1APMSRV is terminated. System action: None. User action: None.
APM047I	L1APMSRV DISABLED Explanation: User exit application driver L1APMSRV is disabled. System action: None. User action: Check for exit trouble, and correct.
APM051I	SMS ???????? ???????? Explanation: Tells if CHKMODEL is SMS managed or not. If SMS managed, no unit/volser will be used for checkpoints. System action: None. User action: None.

SYSLOG File Messages Issued by the APM

SYSLOG Messages from the APM

Description
SPRyy STARTED/PROCESSING/ENDED REQ: zzzzzzzz Explanation: The effector xx, session protocol yy, is starting/processing/ending with request zzzzzzzz. System action: None. User action: None.
DDDDDDDD ALLOCATION ERROR SRC .... INF .... RC .. Explanation: An error occurred when allocating DDDDDDDD. System action: SRC is reported to Connect:Express, INF is SVC99 info code which indicates the allocation parmlist field involved in the error. User action: Check the SRC and INFO codes in IBM documentation.
SESSION ERROR : AAAAAAA NRC .... Explanation: An error occurred during AAAAAAA access to network. System action: NRC is reported to Connect:Express. User action: Check the NRC.
uuuuuuuu TERM USER EXIT ERROR RC:.... Explanation: The 'uuuuuuuu' user exit returned R15=RC. System action: RC is reported in TRC given by Connect:Express. The transfer is interrupted. User action: Check the TRC and PRC codes. Check transfer specifications.
DDDDDDDD ETEBAC3 PARAMETER CARD DEFAULT USED Explanation: No specific ETEBAC3 card was given by user neither by PARMETB3 file nor API= parameter. System action: This is for information purpose. User action: Check if normal or not.
RRRRRRRR I/O ERROR DDD,VOLSER,CM,EC,STAT,SENSES Explanation: An I/O error occurred on sequential file. System action: System message and codes given. User action: Check codes and messages.

SYSLOG Messages from the APM (continued)

Description
RRRRRRRR EEEEEEEE SERVER INVOKED OPTION NOT AUTHORIZED Explanation: An executing transfer needs a Connect:Express OPTION module the customer is not authorized to use. System action: TRC given indicates the option needed. The transfer is interrupted. User action: Check the TRC code. Contact the support for asset protection file.
RRRRRRRR EEEEEEEE SERVER RC:.... Explanation: The 'rrrrrrrr' user application server returned R15=RC. System action: RC is reported in TRC given by Connect:Express. The transfer is interrupted. User action: Check the TRC and PRC codes. Check transfer specifications.
RRRRRRRR EEEEEEEE SERVER DISABLED Explanation: The 'uuuuuuuu' user application server returned error at initialization or abended. System action: The exit will be no more invoked. User action: Check the exit, modify, stop APM and start it again.

WTO Messages Issued by the ANM

The ANM may issue some warnings, especially if you are using X25 connections. These messages are found in the SYSOUT file of the ANM. The information provided can help you to troubleshoot Network problems.

ANM Messages in the SYSOUT File

Code	Description
ANM0001	MCH ..... XIO TO NETWORK FAILED Session has been lost
ANM0002	MCH ..... PHYSICAL LINK FAILED Session has been lost
ANM0003	MCH ..... DIAG RCVD; CD=.. This can occur when the MCH definition declares more SVC's than the subscription does.
ANM0004	MCH ..... STRANGE DATA RCVD Unexpected data received, session with MCH is closed.
ANM0005	MCH .....OPNDST PENDING; CHECK WHY? At initialization an MCH could not be activated and an Opendest is pending. The operator must VARY INACT the MCH FORCE and VARY ACT it. Then he must activate the MCH and pass the ACT TOM command.
ANM0006	APPL ..... GENCB FAILED RC:.. At initialization GENACB for application APPL failed, RC = register 15.. X25 resources initialization procedure is stopped.
ANM0007	GENRPL FAILED RC:.. At initialization GENRPL failed, RC = register 15. X25 resources initialization procedure is stopped.
ANM0008	APPL ..... OPEN FAILED RC:..:.. At initialization OPEN ACB for application APPL failed, RC = register 15: ACB error flag. X25 resources initialization procedure is stopped.
ANM0009	SETLOGON FAILED RC:..:..:..:.. At initialization SETLOGON, RC = register 0:register 15:RPL rtncd:RPL feedback 2. ACBs are closed, X25 resources initialization procedure is stopped.

ANM Messages in the SYSOUT File (continued)

Code	Description
ANM0010	MCH ..... MODCB FAILED RC:.. At initialization MODCB failed for the MCH LU, RC = register 15. The current MCH is set to inactive, next MCH is processed.
ANM0011	MCH ..... OPDEST REJECT RJ:.. At initialization OPEN DEST was not accepted for the MCH, RC = register 15. The current MCH is set to inactive, next MCH is processed.
ANM0012	MCH ..... OPDEST RC:..:..:..:.. At initialization OPEN DEST failed for the MCH, RC = register 0:register15:RPL rtncd:RPL fdbk2. The current MCH is set to inactive, next MCH is processed.
ANM0013	MCH ..... MODCB/RPL RC:..:..:.. At initialization after OPEN DEST successfull MODCB failed for the MCH, RC register15:RPLrtncd:RPLfdbk2. The current MCH is set to inactive, next MCH is processed.
ANM0014	MCH ..... SESSIONC RJ:..:..:..' At initialization, after successfull OPEN DEST, SESSIONC was rejected for the MCH, RC register15:RPLrtncd:RPLfdbk2. The current MCH is set to inactive, next MCH is processed.
ANM0015	MCH ..... SESSIONC RC:..:..:..:.. At initialization after OPEN DEST successfull SESSIONC failed for the MCH, RC register15:register0:RPLrtncd:RPLfdbk2. The current MCH is set to inactive, next MCH is processed.
ANM0016	MCH ..... SENDFB REJECTED:..:..:.. At initialization of a date MCH, the restart procedure failed, send FB request was rejected. RC register15:RPLrtncd:RPLfdbk2. The current MCH is set to inactive, next MCH is processed.
ANM0017	MCH ..... SENDFB ERROR:..:..:..:.. At initialization of a date MCH, the restart procedure failed, send FB packet was in error. RC register15:register0:RPLrtncd:RPLfdbk2. The current MCH is set to inactive, next MCH is processed.
ANM0018	MCH ..... RECEIVE REJECT :..:..:.. At initialization of a date MCH, the restart procedure failed, receive request was rejected. RC register15:RPLrtncd:RPLfdbk2. The current MCH is set to inactive, next MCH is processed.
ANM0019	MCH ..... REST/CONF RC:..:..:..:.. At initialization of a date MCH, the restart procedure failed, receive request was in error. RC register15:register0:RPLrtncd:RPLfdbk2. The current MCH is set to inactive, next MCH is processed.
ANM0020	MCH ..... SENDFF ERROR:..:..:..:.. At initialization of a date MCH, the restart procedure failed, send FF request was rejected. RC:register15:RPLrtncd:RPLfdbk2. The current MCH is set to inactive, next MCH is processed.
ANM0021	MCH ..... ERROR INFO RECEIVED At initialization of a date MCH, the restart procedure failed, error info was received twice.
ANM0022	MCH ..... LINE LOST :..:..:..:.. While listening to the MCH session an error occured. The line is probably lost. RC:register15:Register0:RPLrtncd:RPLfdbk2. The current MCH is disabled. Automatic retry is activated.

ANM Messages in the SYSOUT File (continued)

Code	Description
DTSANM11	NXHSRC42 PCNE/PAD EVENT WITH FLG=0 Unexpected event when session is being cleared, ignored,.
MSHXPR70	ALL XRB BUSY This may occur when more SVC'S are available than used by ANM : incoming calls maximum number is then higher than the maximum control blocks managed by ANM. The number of ANM control blocks is equal to the total number of APM EFFECTORS declared at initialization: XRB = APM*EFF1 + APM2*EFF2+ ….
NXHWTO00	MCH ..... xxxxxxxxxxxxxxx LVL3 DEF ERROR More than ten RESET packets received on this MCH,connected with this remote dial number : probably the gate MCH definition doesn't match the subscription.
NXHWTO22	SVC WILL BE CANCELLED An out of sequence event was received and the SVC is closed. This may happen with gate MCH after re-init phase.
X25WTO00	MCH ..... xxxxxxxxxxxxxxx LVL3 DEF ERROR More than ten RESET packets received on this MCH,connected with this remote dial number: probably the date MCH definition doesn't match the subscription.

Appendix C

Monitor Commands

This appendix lists and describes monitor commands that you can use to activate or deactivate resources.

Commands and Descriptions

The OS/390 monitor commands can be sent to Connect:Express in two ways. First, you can use the Modify command. For example, F PROCNAME,XXX where PROCNAME is the name of the job or the Connect:Express PROC defined by the CMDPRE field. You can also send a command to a subsystem, +XXX. For example, +XXXX where + is the command prefix character in the Connect:Express SYSIN CMDPRE= parameter.

Note: Only $LOAD$ and SNAP Commands are processed by a PLEX server.

The following tables list actions and monitor commands for each resource type.

Commands for the TOM Resource

Action	Command and Description
Stop Monitor activity	**F PSRTOM3,STOP or +STOP** Only use this command when requested by technical support. The monitor stops without interrupting current transfers. Current transfers continue, and others are refused. We advise you to use an initialization procedure where you declare the UPRINI= parameters in the SYSIN file. This procedure executes the P1B2ZIPL utility which changes the Connect:Express status back to START.
Restart Monitor activity	**F PSRTOM3,START or +START** The monitor restarts and begins executing transfers.
Reinitialize the subsystem environment	**F PSRTOM3,$LOAD$ or +$LOAD$** This command makes it possible to force re-loading of tables/modules residing in the CSA and communication areas of the subsystem interface. This is useful to reload the Connect:Express APL table (CICS/...). The Monitor must then be stopped, and this operation will be executed during the next start up. NOTE: The reload of other CSA modules/tables is automatically performed only if some change is detected. You can also use the P1B2P9RL program, with PARM='SSN=TOM ?'.
Refresh the MSGINI= message by Connect:Express	**F PSRTOM3,MSG=stop-1-hour-on-95/01/22-at-12:00** **or** **+MSG=stop-1-hour-on-95/01/22-at-12:00** This command requests the monitor to update the MSGINI= operator message temporarily until the next start-up. This message is displayed for batch/TSO users. No control is done, and a message can be up to 64 characters with no blanks.
Stop the monitor	**P PSRTOM3 or +TERM** This command requests the monitor to stop at once and to interrupt current transfers through a protocol. These transfers can resume later, if Connect:Express is restarted with RUN=H.
Refresh the Asset Protection File	**F PSRTOM3,AUTHDS='APdsn' or +AUTHDS='APdsn'** This command prompts the monitor to process a new temporary Asset Protection File. This file will be available until the next refresh or start of the monitor.

Commands for Troubleshooting

Action	Command and Description
Execute a SYS1.DUMP by Connect:Express	**F PSRTOM3,DUMP or +DUMP** This command requests the monitor to execute a complete SYS1.DUMP of Connect:Express addresses spaces (MONITOR+ANM+APM's+APMBSC). The purpose is to have a full core image of Connect:Express if some anomaly is detected. This assumes, a free sys1.dump and at least 60 cylinders of space. The sys1.dump can be sent unformatted to Sterling Commerce (as PSRDUMPI) for analysis. Connect:Express does not stop while OS/390 executes the dump, but a sys1.dump locks OS/390 for a short time.
SNAP Connect:Express data	**F PSRTOM3,SNAP=E or +SNAP=E (enable snap)** **F PSRTOM3,SNAP=H or +SNAP=H (disable snap)** This command requests the monitor to take snap data and is used only for debugging. The 'SNAP=ON,??' complementary command is given by the support center.
Activate Trace	**F PSRTOM3,TRACE=E or +TRACE=E (enable trace)** **F PSRTOM3,TRACE=H or +TRACE=H (disable trace)** This command enables/disables the incoming connection error tracking facility. All undetermined incoming calls are issued a message in the Connect:Express SYSLOG file. This command must be sent before enabling the trace feature for one Partner.

Action	Command and Description
Start or stop the ATM	**F PSRTOM3,S ATM or +S ATM** **F PSRTOM3,P ATM or +P ATM** This command starts/stops the trace manager. When the ATM is initialized, you can request to trace protocol sessions with one partner or a group of partners.
Trace request to ATM	**F PSRTOM3,TRACEPAR=(Partner,Type) or + TRACEPAR=(Partner,Type)** This command prepares for tracing any inbound or outbound protocol sessions with the Partner. The partner parameter can be either a partner name (Note: the partner rmust exist in the Partners directory) or a root such as 'PAR*' for all partners the name of which begins with 'PAR'. The type parameter indicates if the trace must include both negotiations and data or only negotiations.

Commands for a Partner Resource

Action	Command and Description
Disable the Partner table	**F PSRTOM3,PCT=H or +PCT=H** The Partner Control Table resource is disabled. Connect:Express no longer submits outgoing requests and denies all incoming requests. Current transfers continue.
Enable the Partner table	**F PSRTOM3,PCT=E or +PCT=E** The Partner Control Table resource is enabled. Connect:Express can submit and accept any requests.
Refresh the SYSSNA file by Connect:Express	**F PSRTOM3,REF SYSSNA** This command requests the monitor to read the SYSSNA edit file and refresh it. Controls are carried out and invalid records are disabled.
Refresh the SYSX25 file by Connect:Express	**F PSRTOM3,REF SYSX25** This command requests the monitor to read the SYSX25 edit file and refresh it. Controls are carried out and invalid records are disabled.
Refresh the SYSTCP file by Connect:Express	**F PSRTOM3,REF SYSTCP** This command requests the monitor to read the SYSTCP edit file and refresh it. Controls are carried out and invalid records are disabled.

Commands for a File Resource

Action	Command and Description
Disable the File table	**F PSRTOM3,FCT=H or +FCT=H** The File Control Table resource is disabled. Connect:Express no longer schedules outgoing requests and denies all incoming requests. Current transfers continue.
Enable the File table	**F PSRTOM3,FCT=E or +FCT=E** The File Control Table resource is enabled.
Reactivate a Partners list	**F PSRTOM3,ACF=#LIST or +ACF=#LIST** This command requests the Monitor to reinitialize the Partner table associated with a symbolic File name after modifying the Partner list in *SYSPRM* (#LIST).
Refresh the SYSJCL file by Connect:Express	**F PSRTOM3,REF SYSJCL** This command requests the monitor to read the SYSJCL library and refresh all the files in it. This can be used if a user error was detected in any of the SYSJCL skeletons.

Commands for a Request Resource

Action	Command and Description
Disable the request table	**F PSRTOM3,RCT=H or +RCT=H** The REQUESTS global resource is disabled. Connect:Express no longer schedules outgoing requests and denies all incoming requests. Current transfers continue.
Enable the request table	**F PSRTOM3,RCT=E or +RCT=E** The REQUESTS global resource is enabled. Connect:Express can submit and accept all requests.

Action	Command and Description
Increment the value of request numbers	**F PSRTOM3,CHG REQ=+xxxxxxx or +CHG REQ=+xxxxxxx** This command requests the Monitor to modify request numbering by adding the value indicated to the current number. The value of xxxxxxx can be between 1 and 9999999.

Commands for a Transfer Resource

Action	Command and Description
Stop an APM	**F PSRTOM3,P APM=xx or +P APM=xx** where *xx* is the number of the APM to stop (01 to 08). APM xx was previously started by Connect:Express.
Start an APM	**F PSRTOM3,S APM=xx or +M APM=xx** where *xx* is the number of the APM to start (01 to 08). APM xx was previously stopped using a STOP command, or it was described as DISABLED in the SYSIN when starting Connect:Express. (DAPMxx=(H,16,A).
Modify the classes for an APM	**F PSRTOM3,M APMxx=xyz or +S APMxx=xyz** where *xx* defines the number of the APM chosen. In this example, the APM's classes were changed to xyz from the previous values.
Stop the AFM	**F PSRTOM3,P AFM** AFM was previously started by Connect:Express.
Start the AFM	**F PSRTOM3,S AFM** AFM was previously stopped using a STOP command, or it was described as DISABLED in the SYSIN when starting Connect:Express. (AFMPRC=afmproc,H).

Commands for a Network Resource

Action	Command and Description
Activate an MCH by Connect:Express	**F PSRTOM3,ACT,MCH=xxxxxxxx or +ACT,MCH=xxxxxxxx** This command requests the monitor to require a reinitialization of MCH xxxxxxxx from the ANM.
Activate an ANM component by Connect:Express	**F PSRTOM3,ACT,DRV=x or +ACT,DRV=x** This command requests the monitor to require the reinitialization of the following components from the ANM: • SNA-LU0 component if x=S • SNA-LU2 component if x=T • SNA-LU6.2 component if x=C • X.25 component if x=X • TCP-IP component if x=I
Activate SSL	**F PSRTOM3,SSL=ON** This command requests the monitor to require the reinitialization of the SSL handler by the ANM.
Disable SSL	**F PSRTOM3,SSL=OFF** This command requests the monitor to require the termination of the SSL handler by the ANM.

Appendix D

Initialization Parameters

This appendix identifies the SYSIN parameters for Connect:Express and describes the parameters that are required for Connect:Express to work as a stand-alone, a Plex manager, or a Plex server, and the optional parameters that you can use to increase resource productivity and to take advantage of advanced features.

SYSIN Parameters

Connect:Express can be started as a stand-alone, a Plex manager, a Plex server, or a local TOM, but only the stand-alone and the Plex Manager are in charge of monitoring transfers. This is called a global TOM as opposed to a local TOM.

You set Connect:Express configuration parameters in the SYSIN file. Then, you must define the ANM and at least one APM to manage transfers. In a SYSPLEX configuration, the Plex manager processes the SYSIN file as a stand-alone monitor, and the Plex server only uses the parameters it needs.

SYSIN Parameters Required for Initialization

When initializing Connect:Express, you must configure the required SYSIN parameters shown in this section, the general ANM parameters, and the parameters for the link type you are using. Optional SYSIN parameters enable you to use more than one APM to start the AFM or to activate additional functions. The following table lists the SYSIN parameters required for Connect:Express initialization in alphabetical order.

SYSIN Parameters Required for Connect:Express Initialization

Field	Length	Description
ACTEXT	4 alpha characters	The ACTEXT parameter of CONT makes it possible to ignore a return code from a user exit (UEXJNL=) other than 0. The Monitor deactivates the exit. The ACTEXT parameter of STOP makes it possible to stop Connect:Express if the user exit (UEXJNL=) ends with a return code other than zero. If this return code is between 64 and 128, Connect:Express places an 80-character message in the SYSLOG file that is returned by the exit through the communication area. Examples: CONT, STOP

SYSIN Parameters Required for Connect:Express Initialization (continued)

Field	Length	Description
ADHOCN	Y, N, or U	If ADHOCN is set to **Y**, the TSO AD HOC request is allowed, with mandatory RACF user and password to be transmitted with the request to the remote Partner. This RACF facility is available for protocol number 5 (PeSIT-E) only, and with another Connect:Express. If ADHOCN is set to N, the TSO AD HOC request is not permitted for incoming or outgoing transmissions. This is the default. If ADHOCN is set to **U**, the TSO AD HOC request is allowed, with optional RACF user and password to be transmitted with the request to the remote Partner. This is the UNSAFE mode. For incoming calls you can use the RACFUD= field as a default RACF user.
ANMPRC	6 to 8 alphanumeric characters	ANMPRC is the name of the start procedure for the auxiliary network manager. You can use characters 1 to 8 to generate a PROCSTEP name. Example: ANMPRC=TOM3ANM Results in: S TOM3ANM.TOM3ANM
APLNUM	1 to 2 numeric character	This parameter can be described in two forms. In the first form, APLNUM indicates the number of applications or address spaces which can be connected to the Monitor. Each address space receives the end of transfer notifications. The maximum authorized value is 64. Examples: 16, 4
	- or -	
	2 + 2 numeric characters	This second form enables you to indicate two values separated by a comma and placed between brackets. The first parameter indicates the number of applications (1 to 64) as described in the first form. The second parameter indicates the number of entries stacked (1 to 32) for each application. If the previous form is used, the default is 8. Each entry uses 512 bytes of CSA. The application table and the stack are initialized only after an OS/390 IPL, or if "$LOAD$" has been issued during the previous session of the CONNECT:Express Monitor. Example: (2,10)
APMPRC	6 to 8 alphanumeric characters	APMPRC is the name of the started procedure for an auxiliary protocol manager. Characters 1 to 6 are used with the APM number to generate a different PROCSTEP name per task. Example: APMPRC=TOM3APM Results in: S TOM3APM.TOM3AP01 for APM number one...
AUTHDS=	1 TO 44 alphanumeric characters	AUTHDS= is the data set name of the Asset Protection File.
CMDPRE	1 alphanumeric characters	CMDPRE is the unique prefix character of the subsystem command. All commands intended for the Monitor can be transmitted by MODIFY or through the subsystem feature. Examples: F TOMJOB,P APM=01 or +P APM=01 See Appendix C for a list of monitor commands.

SYSIN Parameters Required for Connect:Express Initialization (continued)

Field	Length	Description
DAPM01	1 to 14 alphanumeric characters	DAPMxx=(x/nn/xyz) DAPMxx defines the auxiliary protocol manager resource where *xx* can assume values 01 to 08. The standard license enables two APMs. The first one is mandatory. It is possible to increase the number of APMs. If you want to use more than two APMs, you have to modify the license contract. The first field, *x*, defines the submission of the APM and can assume the value E, H, or O. For DAPM01, only E and H are valid. E – APM must be started during the initialization of Connect:Express. H – APM must not be started during initialization, but can later be submitted by command. O – APM is not used and cannot be submitted. The second field, *nn* defines the number of protocol servers or effectors which can be initialized in this APM. It is stated as two numeric bytes (01 to 16). Note: If you have huge user exits, you may have to specify less than 16 effectors to avoid ABEND80A/ABEND106 in the APM region. The third field, *xyz*, defines transfer classes. In the following example, the APM is started by Connect:Express at initialization with 16 servers for ABCDEFG class requests. DAPM01=(E/16/ABCDEFG)
DPCPSW	8 alphanumeric characters	DPCPSW is the default symbolic password of your Connect:Express. Examples: ZEUS, HERMES, PI31416
DPCSID	8 alphanumeric characters	DPCSID is the symbolic name given to the Connect:Express Monitor. Partners use this name for identification when the session begins. This is your Partner name. Examples: NEPTUNE, BACKUP01, PSRTOM3
FCTADD	2 numeric characters	If FCTADD is *equal to zero*, no update of the file directory is dynamically transmitted to the Monitor tables. Updates are only taken into account for the next hot or cold start of the Monitor. If FCTADD is *not equal to zero*, all updates of the File directory under TSO/ISPF are dynamically transmitted to the file table (FCT) in the Connect:Express Monitor and are available for use. The following conditions are necessary. • Connect:Express is active. • There is no request for this file. A message is displayed if the UPDATE cannot be sent to the FCT. It is recorded in the File directory and is taken into account at the next hot or cold start of the Monitor. The value of FCTADD represents the maximum number of new entries in the File directory which can be sent dynamically to Connect:Express. Examples: 00, 33, 99
MAJIND	17 alphanumeric characters	MAJIND is the major index of temporary files allocated by Connect:Express when unloading and loading partitioned or VSAM-type files. Examples: PSR.PSR, PSR
MAXSRQ	2 numeric characters	MAXSRQ is the maximum number (up to 64) of IEFSSREQ issued simultaneously by TSO users or batch JOBs using "L0B2Z20." Examples: 08, 12, 16
MSGINI	1 to 64 characters	MSGINI indicates a message to be sent to an operator when you initialize a TSO/ISPF session or activate a Connect:Express function.
MSGPRX	4 alphanumeric characters	MSGPRX represents the four prefix characters used in messages transmitted by the Monitor. Examples: TOM1, TOM2, TOMP, and so on.

SYSIN Parameters Required for Connect:Express Initialization (continued)

Field	Length	Description
PCTADD	2 numeric characters	If PCTADD is *equal to zero*, no update of the Partner directory is sent to the Monitor tables. Updates will be taken into account at the next hot or cold start of the Monitor. If PCTADD is *not equal to zero*, all the UPDATES of the Partner directory under TSO/ISPF are dynamically sent to the Partners Control Table (PCT). The following conditions must be present. • Connect:Express must be active • There is no request for this Partner. A message is displayed if the UPDATE cannot be sent to the PCT. It is recorded in the Partner directory and is taken into account at the hot or cold start of the Monitor. The value for PCTADD represents the maximum number of new entries in the Partner directory which can be dynamically sent to Connect:Express. Examples: 00, 22, 99
RACFCN	Y, N, or S	If RACFCN is set to S or Y, Connect:Express controls monitor and local/remote user authorization to access the transfered DSN before opening it. The monitor and the TSO user, JOB, or remote Partner issuing a file transfer request must have RACF authorization to read it (Transmit) or to create/write it (Receive). The remote partner RACF userid/group can be set in the Partners directory. A default userid can also be set in the *SYSIN* file (See RACFUD field). Connect:Express is authorized to access files based on the RACF definition of its started task (STC). Authorization is performed with the RACROUTE-SAF (compatibile with any security products). NOTE: The AD HOC facility enables the remote Partner to be identified by its own userid and password. (See ADHOCN field.) If RACFCN is set to N, Connect:Express does not check the RACF origin of the request. Warning: This may cause the monitor to end abnormally with an Abend913 if it is opening a restricted DSN.
RMFLOG	Y or N	If RMFLOG is set to **Y**, Connect:Express sends RMF records. This measures the Connect:Express system utilization. If RMFLOG is set to **N**, Connect:Express does not produce RMF records.
RQEMAX	4 numeric characters	RQEMAX represents the maximum number of requests in the Request Control Table (RCT). This value includes the possible number of requests pending, those in progress, and those which have been interrupted. The requests which have been successfully executed are deleted from the RCT. The maximum number of requests cannot be greater than 1024. Examples: 0512, 1024
SMFREC	3 numeric characters	SMFREC is the user SMF record number assigned to Connect:Express. Enter 000 if you do not want these records. Examples: 240, 250, 000
SMSSDB	Y or N	SMSSDB tells the Monitor which method to use when computing the blocksize. If "Y" is specified, the blocksize is computed by SMS, otherwise Connect:Express computes the blocksize, based on the volume where the received file will reside. Allocation in bytes, Kilo bytes or Mega bytes can be used only if SMSSDB=Y.
STIMEV	2 + 2 numeric characters	This parameter includes two mandatory fields separated by a comma and placed between brackets. The first field is the number of minutes between two call attempts. Examples: (06,..) or (15,..) The second parameter is the number of minutes after which Connect:Express restarts a local request that was interrupted following a network incident, or a local request that was rejected by the Partner with an error code defined in the T1B2PCOD table. Examples: (..,05) or (..,00)

SYSIN Parameters Required for Connect:Express Initialization (continued)

Field	Length	Description
STIMOC	2 numeric characters	This parameter is used with the STIMEV parameter and defines the maximum number of attempts for calling a Partner or restarting a request. STIMOC represents the maximum number of call attempts on one link to another Monitor. If the call is not successful after a time equal to the first parameter of STIMEV * STIMOC, then one of the following will occur: • A new link is processed • The Partner is disabled (All alternate links have been processed and ACR=no in the Partner definition.) If ACR=**NO**, in the Partner definition, the TRC code is set to 2076 and processing stops. If ACR=**YES**, the retry and restart processes continue until the transfer is successful. STIMOC represents the maximum number of attempts for a transfer request. If the transfer is not accepted after a time equal to the second parameter of STIMEV * STIMOC, then the request is disabled with a status of ERROR SELECTION. Examples: 06, 04, 02
TOMACB	8 alphanumeric characters prefix of APLPFX (6) + 00	TOMACB is the name of the VTAM application allocated to the Monitor for SNA sessions with local Monitors of the same cross-domain. These Monitors ensure transmission of requests from one machine to the central monitor by a defined VTAM link. This parameter is only used in a local/global Connect:Express environment. If Connect:Express is operating alone, TOMACB must be coded with the value NONE. Examples: TOMAPLID, MASTLTOM, ANMAPP00
TOMLCL	2 numeric characters	TOMLCL indicates the number of simultaneous SNA sessions (1 per local Connect:Express) with a limit of 16. This parameter is used only in a local/global Connect:Express environment. If Connect:Express is operating alone (TOMACB = NONE), TOMLCL must be coded with the value 00. Examples: 00 16
UEXJNL	1 to 8 alphanumeric characters	UEXJNL is the name of the user exit routine which receives control for each end of transfer. Control is also given at INIT and TERM of Monitor. Sample user exits can be found in the Connect:Express SAMPLIB. L1B2PDIX in the *SAMPLIB* EX#DIX is a driver which receives a SYSIN file (SYSINEXT) with a list of user exits. You must place the module to be run in the library defined by the SYSLIB card of the CONNECT:Express Monitor. The setting is NONE if this exit is not implemented. Examples: L1B2PDIX, NONE, or MYJNL
UPRCPI	8 alphanumeric characters	UPRCPI is the name of a user procedure to be started after correct initialization of Connect:Express. This parameter must be coded with the value NONE, if it is not used. Examples: UPRCPI=USERCPI
UPREND	8 alphanumeric characters	UPREND is the name of a user procedure to be started after correct termination of Connect:Express. This parameter must be coded with the value NONE, if it is not used. Examples: UPREND=USEREND
WRKUNT	8 alphanumeric characters	WRKUNT is the unit name Connect:Express uses to allocate temporary files created when unloading and loading partitioned or VSAM-type files. Example: SYSALLDA

SYSIN Parameters for the ANM

ANM parameters describe the characteristics of network resources assigned to the ANM. The following sections describe general ANM parameters, X.25 parameters, and TCP/IP parameters.

General ANM Parameters

The following table describes general ANM parameters.

Field	Length	Description
APLPFX	6 alphanumeric characters	This required parameter is the 6-character prefix of the VTAM application name used by the ANM. The complete name comes with a 2-digit suffix number, from 01 to 07. See Appendix E*, Definition of VTAM Resources*. Examples: APLPFX=ANMAPP: ANMAPP01 SNA LU0 X-DOMAIN ANMAPP02 X.25NPSI DATE handler ANMAPP03 X.25NPSI PCNE ANMAPP05 3270 ANMAPP06 X.25NPSI GATE ANMAPP07 LU6.2
MAXEXT	3+3 numeric characters	This parameter has two mandatory fields separated by a comma and placed between brackets, and describes the rules for extending buffers. The first value is the number of possible simultaneous extensions. The second value is the maximum buffer length. The extension of a buffer is acquired by the ANM during initialization. The APM requests the extended buffer from the ANM when protocol negotiations result in a message length more than the STDMSG parameter. This must be related to the MESSAGE SIZE fields in the T1B2PSxx session tables. Example: MAXEXT=(016,128)
MCHNBR	2 numeric characters	This parameter is the number of multi-channel lines handled by the ANM. The maximum is 32.
STDMSG	2 numeric characters	This parameter is the standard network buffer length to be acquired by the ANM at initialization. Example: STDMSG=04

X.25 Parameters

X.25 parameters are mandatory if MCHNBR is greater than zero. Each MCH must be defined with the subparameters described in the following table.

Field	Length	Description
MCHLUN = luname	1 to 8 alphanumeric characters	Required. Specifies the LU name for each MCH.
MCHMSC	1 alphanumeric character	Optional. This parameter is the X.25 network selection code (MCH). This parameter permits you to identify the MCHs which access the same X.25 network. It is referred to in the Partner directory. Leased or switched point-to-point lines with X.25 protocol must be considered as a distinct X.25 network. Example: MCHMSC=A
MCHNAM	1 to 8 alphanumeric characters	Required. This parameter corresponds to the MCH macro name in the NCP.
MCHPKS†	4 numeric characters	Required. This value specifies the size of the X.25 packets for this subscription.

Field	Length	Description
MCHRTR	2+2 numeric characters	This optional parameter has two mandatory fields separated by a comma and placed between brackets. Indicates the occurrences and time (in 30 seconds) of the MCH reactivation which is performed automatically if the MCH is lost. Example: MCHRTR=(05,02)
MCHTYP	1 alpha character	Required. Indicates the type of MCH (D=DATE, G=GATE).
MCHVCN†	3 numeric characters	Required. Specifies the number of generated switched virtual circuits for this MCH.
MCHWDS†	2 numeric characters	Required. Indicates the size of the X.25 packet window for this subscription (=MCH).
MCHXLA	1-15 characters	Optional. This is the local X.25 address of a specific MCH. The Partner local address, if it exists in the directory definition, is added at the end.

† These parameters must match the subscription parameters.

TCP/IP Parameters

The TCP/IP parameters are shown in the following table.

Field	Length	Description
TCPORG	4+8 alphanumeric characters	The TCPORG parameter defines the TCP/IP stack and access method. Two parameters are specified between brackets and the first parameter is fixed. ‘HPNS’ is for the IBM standard socket interface. 'SOE' is for the IBM Open Edition interface. The second parameter indicates the job name of the TCP/IP stack running on the system. This is valid for both IBM or Computer Associates software. Example: (HPNS,TCPIPJOB), (SOE,) NOTE: Previous definitions are still supported (ITL31,ACSS,4096) for TCP ACCESS and (IBM3.1,TCPIPJOB).
TCPPRT	5 numeric characters	The TCPPRT parameter defines the port number on which the ANM listens for PeSIT and ETEBAC3 incoming calls.
TCPPRO	5 numeric characters	The TCPPRO parameter defines the port number on which the ANM listens for ODETTE-FTP incoming calls.

SSL Parameters

The SSL parameters are shown in the following table.

Field	Length	Description
SSLOPT	1 alpha character	N/Y: Determines if SSL is used or not, No is the default. 'Y' requires a minimum number of parameters among the SSL parameters shown below.
SSLKRG	1 to 44 alphanumeric characters	Name of the RACF Keyring associated with the ANM. This parameter excludes SSLDTB and SSLPSW parameters.
SSLDTB	1 to 44 alphanumeric characters	Name of the HFS database in which certificates are stored. This parameter requires SSLPSW and excludes SSLKRG..

Field	Length	Description
SSLPSW	8 alphanumeric characters	Password to access the HFS database in which certificates are stored.
SSLCER	1 to 34 alphanumeric characters	Label of the local certificate defined in he HFS database or the RACF keyring. It can include blanks. This parameter is optional, the default of the database or keyring is used.
SSLPRT	From 1 to 65535 numeric	TCP/IP port number for inbound PeSIT SSL calls.
SSLUDF	2 to 16 hexadecimal characters	X25 user data expected from PeSIT SSL Clients. The number of characters must be even. Example : SSLUDF=AB02
SSLSAD	1 to 5 numeric characters	X25 subaddress expected from PeSIT SSL Clients.
SSLPRO	From 1 to 65535 numeric	TCP/IP port number for inbound Odette SSL calls.
SSLUDO	2 to 16 hexadecimal characters	X25 user data expected from Odette SSL Clients. The number of characters must be even. Example : SSLUDF=AB02
SSLSAO	1 to 5 numeric characters	X25 sub address expected from Odette SSL Clients.
SSLTRC	1 numeric character	0/1 - '0' is the default. '1' activates the environment trace of the SSL handler. The trace is written in a SYSPRINT file of the ANM.
SSLTIM	1 to 6 numeric characters	Number of seconds during which the SSL session identifier is kept. The default is 86400 seconds.
SSLLEV	2 numeric characters	Minimum SSL version supported. The default is 30, for SSLV3 and TLSV1. 20 for SSLV2 to TLSV1, 31 for TLSV1 only.
SSLAUT	1 alpha character	Indicates if client authentication is required. N is the default.
SSLCIP	2 to 32 hexadecimal characters	Cipher suite: indicates preferences from options supported by z/OS SSL services. The number of charaters must be even. The default is the z/OS SSL services default : 050435363738392F303132330A1613100D0915120F0C0306020100 Example: SSLCIP=09060504

SYSIN Parameters Processed by a Plex Server

The following parameters are processed by the Plex Server without using any transfer resources.

Field	Length	Description
ACTEXT	4 alpha characters	The ACTEXT parameter of CONT makes it possible to ignore a return code from a user exit (UEXJNL=) other than 0. The Monitor deactivates the exit. The ACTEXT parameter of STOP makes it possible to stop Connect:Express if the user exit (UEXJNL=) ends with a return code other than zero. If this return code is between 64 and 128, Connect:Express places an 80-character message in the SYSLOG file from the exit through the communication area. Examples: CONT, STOP

Field	Length	Description
APLNUM	1 Numeric character	This parameter can have two forms. In the first form, APLNUM indicates the number of applications (address spaces) connected to the Server. This address space receives the end of transfer notifications. The maximum authorized value is 8. Examples: 4, 8 NOTE: Applications are connected to the sub-system. You can connect as many applications to each server as to the manager. For example, if 4 applications can be connected to each server or to the manager and 4 servers are active, 20 applications can be connected to the monitor through the SYSPLEX.
	- or -	
	1+2 Numeric character(s)	In the second form, APLNUM lets you indicate the number of applications and the number of stacked entries (1 to 32) for each application. If the previous form is used, the default is 8. Each entry uses 512 bytes of CSA. The application table and the stack are initialized only after an OS/390 IPL or if "$LOAD$" has been issued during the previous session of the Connect:Express Monitor. Examples: (4, 8) or (24,16)
CMDPRE	1 alphanumeric character	CMDPRE is the unique prefix character of the subsystem command. All commands intended for the Monitor can be transmitted by MODIFY or through the subsystem feature. Examples: F TOMJOB,P APM=01 or +P APM=01 Note: Only some commands are processed by the server, like $LOAD$ and SNAP. See Appendix C for a list of monitor commands.
MAXSRQ	2 Numeric characters	MAXSRQ is the maximum number (64) of IEFSSREQ issued simultaneously by TSO users or batch JOBs using "L0B2Z20". Examples: 08, 12, 16 Note: This parameter applies to the local subsystem image.
MSGPRX	4 alphanumeric characters	MSGPRX represents the four prefix characters used in messages transmitted by the Monitor. Examples: TOM1, TOM2, TOMP, and so on.

Local TOM SYSIN Parameters

Field	Length	Description
TOMACB	8 alphanumeric characters prefix of APLFX (6) + 00	TOMACB is the name of the VTAM application allocated to the Monitor for SNA sessions with local Monitors of the same cross-domain. These Monitors ensure transmission of requests from one machine to the central monitor by a defined VTAM link. (See Appendix E, *Definition of VTAM Resources.*) This parameter is only used in a local/global Connect:Express environment. If Connect:Express is operating alone, TOMACB must be coded with the value NONE. Examples: TOMAPLID, MASTLTOM, ANMAPP00
TOMLCL	2 numeric characters	TOMLCL indicates the number of simultaneous SNA sessions (1 per local Connect:Express) with a limit of 16. This parameter is only used in a local/global Connect:Express environment. If Connect:Express is operating alone (TOMACB = NONE), TOMLCL must be coded with the value 00. If TOMACB=NONE _TOMLCL=00 Examples: 00 16

Optional SYSIN Parameters

These optional parameters enable you to configure Connect:Express for FTP transfers, increase the number of simultaneous transfers, or activate optional features.

APM SYSIN Parameters

The following table describes the optional APM parameters for CONNECT:Express. Changing the APM SYSIN parameters is discussed in *APM Initialization State* on page D-12. The number of APMs that can be started depends on the AUTHDS authorization file .

Field	Length	Description
DAPM02	1 to 14 alphanumeric characters	Standard Example: DAPM02=(H/16/Z) In this example, the APM is not started by Connect:Express at initialization, but can be submitted later by Connect:Express through an operator command with 16 servers for class Z requests.
DAPM03	1 to 14 alphanumeric characters	Needs a license. Example: DAPM03=(O/00/*) In this example, the APM number 03 will be marked not used in the transfer resource list. This is equivalent to not giving a DAPM03 card.
DAPM04	1 to 14 alphanumeric characters	Needs a license.
DAPM05	1 to 14 alphanumeric characters	Needs a license.
DAPM06	1 to 14 alphanumeric characters	Needs a license.
DAPM07	1 to 14 alphanumeric characters	Needs a license.
DAPM08	1 to 14 alphanumeric characters	Needs a license.

ATM SYSIN Parameter

To activate the protocol trace manager, you need to indicate the ATM procedure name.

Field	Length	Description
ATMPRC	6 to 8 Alphanumeric characters	ATMPRC is the name of the started procedure for the auxiliary trace manager. Example: ATMPRC=TOM3ATM Results in: S TOM3ATM.TOM3ATM

AFM SYSIN Parameters

The AFM parameters are combined with the SYSPARM file of the AFM. This parameter file is described in the FTP Guide.The AFM SYSIN parameters are listed below.

Field	Length	Description
AFMALL	1 alphanumeric character	This parameter is used to manage the LIST and NLIST commands received from FTP clients. If AFMALL=N, the response will consist of all file definitions that are available to the current FTP Client, excluding definitions available for ALL Partners. The file definitions (screen 2/5) which has the Client name in the 'TRANSMITTING PARTNER' or 'RECEIVING PARTNER' fields will be selected. The file definition that has a Partner list in one of these two fileds is selected if the client is in the list. If AFMALL=Y, the response will consist of all files definitions that are available to the current FTP Client, including definitions available for ALL partners ($$ALL$$ or * in the Transmitting Partner or Receiving Partner fields). Example: AFMALL=Y
AFMPRC	6 to 8 alphanumeric characters	AFMPRC is the name of the started procedure for the auxiliary FTP manager. Example: AFMPRC=TOM3AFM Results in: S TOM3AFM.TOM3AFM
AFMPRF	8 alphanumeric characters	The AFMPRF parameter indicates the general file profile used for incoming FTP transfers.This symbolic name must correspond to a file directory entry. If an FTP client request doesn't specify the file profile, the default for this Partner is used. If no default profile is defined in the FTP Partner entry, this general default name is used. Example: FTPDEF
AFMCLS	15 alphanumeric characters (1/3/3/3)	This parameter is composed of 4 subparameters, separated by a slash and placed between brackets. The first subparameter is a transfer request class identifier (A to Z). The second one is the total number of simultaneous sessions processed within this class with a maximum of 128. The next 2 parameters indicate the number of ongoing and outgoing simultaneous sessions. The first example means that 30 ongoing calls can be accepted from FTP clients, and 2 outgoing calls to FTP servers can be processed simultaneously for class A transfers. Examples: AFMCLS=(A/032/030/002) AFMCLS=(B/032/000/000)
HTTPLST	3 fields (6,5,5)	This parameter is used with the Connect:Express HTTP option. It is composed of 3 subparameters, separated by a comma, and placed between parenthesis. They provide the default options for sending the list to the http user. The first subparameter indicates if a list must be sent (LIST) or (NOLIST). The second subparameter indicates if the list contains the files control table (FCT) definitions or not (NOFCT), the third subparameter indicates if the list contains hold requests (RCT) or not (NORCT). The second and third subparameters are omitted if the first subparameter is NOLIST. See the *HTTP Option Guide* for more information about implementing http list rights. Examples: (NOLIST,,), (LIST,NOFCT,RCT)

General SYSIN Parameters

The following table describes the optional CONNECT:Express parameters that you can add.

Field	Length	Description
RACFUD	'userid'	This optional RACFUD field is the racf-user by default. Programming note: it is advised to always have it coded in SYSIN. If RACFUD is set to an racf-user-name, this user is used for logical DSN access control in two cases: • If there is no racf-user in the Partner definition • If there is no racf-user in ADHOC incoming calls with UNSAFE mode, ADHOCN=U.
UEXFTS UEXFTE UEXFPS UEXFPE	1 alphanumeric character	These optional UEXF.. fields are the default names of beginning and end of transfer exits. UEXFTS is for transmit-start-exit, UEXFTE is for transmit-end-exit, UEXFRS is for receive-start-exit, and UEXFRE is for receive-end-exit. If any UEXF.. is specified, and no name is set in the corresponding field of the current File Entry definition, then this exit is used. If NONE is specified in the File Directory definition, NO exit is used not even the UEXF.. default exit. For example, you can use these parameters for installing the Utilities option L1GFIUE1 general exit and make it available for most files.
UPRFCT	1 alphanumeric character	UPRFCT can be set to Y to allow any OS/390/CLIST command to be executed from command fields of a file directory entry. If UPRFCT is set to N, no command is executed, and an informational message is issued. This parameter is optional, and the default value is Y.
UPRABE	8 alphanumeric characters	UPRABE is the name of a user procedure to be started after an abnormal/ABENDED termination of Connect:Express. Example: UPRABE=USERABE
LRECLT	1 alphanumeric character	The LRECLT parameter provides extra control for the file allocation parameter received from a remote transmitter. The Record Length parameter received from a sender may or may not be checked. If LRECLT=Y, the LRECL defined in the file entry for creating the file, or the LRECL of a pre-allocated file, must match the LRECL parameter sent by the transmitter. If LRECLT=N, no verification is performed. This means that by combining LRECLT and RECFMT parameters, you can receive a fixed record length file into a variable record length file or receive a variable record length file with your own maximum record length that can be different from the transmitter's maximum record length. The verification is done during transfer of the file, and the length of each record received must be consistent with the allocation LRECL .
RECFMT	1 alphanumeric character	The RECFMT parameter provides extra control for the file allocation parameter received from a remote transmitter. The Record format parameter received from the sender may or may not be checked. If RECFMT=Y, the RECFM defined in the file entry for creating the file or the RECFM of a pre-allocated file, must match the RECFM parameter sent by the transmitter. If RECFMT=N, no verification is performed. This means that by combining LRECLT and RECFMT parameters, you can receive a fixed record length file into a variable record length file or receive a variable record length file with your own maximum record length that can be different from the transmitter's maximum record length. The verification is done during transfer of the file, and the length of each record received must be consistent with the allocation LRECL.

Field	Length	Description
ODTUDF	1 to 4 alphanumeric characters	This is the X25 User Data Field expected for ODETTE FTP incoming calls. When Connect:Express receives this string in the X25 Call packet, it assumes that the caller is an ODETTE FTP Partner. The ODTUDF parameter is processed before the ODTSAD described below. Note: The first character must be é (hexadecimal "C0"). Example: éODT
ODTSAD	1 to 8 numeric characters	This is the X25 sub-address expected for ODETTE FTP incoming calls. When Connect:Express receives this string in the X25 Call packet, it assumes that the caller is an ODETTE FTP Partner. The ODTSAD parameter is processed after the ODTUDF described above. Examples: 9 , 45

Changing SYSIN Parameters

When you change some SYSIN parameters, you must cold start Connect:Express to make those changes available. For example, if you change the APM parameters, you must cold start Connect:Express for the changes to become effective.

If you change any of the following parameters, you cannot hot start Connect:Express:

- DPCSID
- DPCPSW
- RQEMAX
- MAJIND
- CMDPRE
- DAPMxx (adding or deleting cards)
- APLNUM

APM Initialization State

The state of the APM is different depending on if the Connect:Express Monitor is hot-started or cold-started. During a cold start, only the SYSIN file is processed. During a hot start, some information is restored from the previous situation recorded in the monitor checkpoint file (SYSCHK). The following Modify commands sent to the Connect:Express monitor affect a hot start:

- "P APM=nn" stops APMnn, and disables it for the next hot start.
- "S APM=nn" starts APMnn, and enables it for the next hot start.
- M APMxx=xyz changes the APM classes. The new classes are kept during the next hot start.

You can change the status of an APM in the SYSIN file, the classes it serves, and the number of effectors it runs. When the Connect:Express Monitor is cold started, the classes and the number of effectors are taken from the contents of the card "DAPMxx=(s/nn/abcdefg)" in the Monitor’s SYSIN file. When the Connect:Express Monitor is hot started, the classes and the number of effectors are taken from the contents of the card "DAPMxx=(s/nn/abcdefg)" in the Monitor’s SYSIN file, or from the information stored in the CHECKPOINT file (SYSCHK) using the rules described in the table below.

The first column (SYSCHK) indicates the APM status found in the checkpoint file. The second column (SYSIN) indicates the APM status found in the SYSIN file. The next columns give the resulting STATUS, and the files from which the CLASS parameter and the EFFECTORS parameter are taken.

SYSCHK	SYSIN	STATUS	CLASS	EFFECTORS
E	E	E	SYSCHK	SYSIN
D	E	D	SYSCHK	SYSIN
O	E	E	SYSIN	SYSIN
E	D	E	SYSCHK	SYSIN
D	D	D	SYSCHK	SYSIN
O	D	D	SYSIN	SYSIN
E	O	O	SYSIN if any	SYSIN
D	O	O	SYSIN if any	SYSIN
O	O	O	SYSIN if any	SYSIN

Appendix E

Definition of VTAM Resources

This appendix provides definitions of the VTAM resources for Connect:Express. It includes definitions of an application major node, a logmode, a switched major node for X.25, an interpret table for transparent PAD, and an X25NPSI definition for X.25 links.

Definition of an Application Major Node

The VTAM applications listed below are required to start the ANM with all its handlers. The contents are in the *INSTLIB* in the member VTAPPTOM. The major node must be added to SYS1.VTAMLST. No ANMAPP04 application is present.

Application	Description
ANMAPP	This is the value of the APLPFX= Connect:Express parameter that you set. All the applications used by ANM have the same 6-character prefix coded in the APLPFX initialization parameter of Connect:Express. It is user-definable.
ANMAPP00	This is an optional application name. It is only needed if Connect:Express global-local functions are in use and if TOMLCL does not equal zero. To use ANMAPP00, the TOMACB parameter of a global Connect:Express must refer to the application. This application is used by the Connect:Express address space.
ANMAPP01	This is the ACB name of the ANM SNA application-to-application handler for transfers through an SNA network.
ANMAPP02	This is the ACB name of the ANM DATE CTCP which handles switched virtual circuits on X.25 links dedicated to the ANM.
ANMAPP03	This is the ACB name of the ANM PAD/PCNE handler which manages data transfer on X.25 links dedicated to the ANM.
ANMAPP05	This is the ACB name of the ANM 3270 handler which manages sessions with TOMPC running on PCs using a 3270 emulation card.
ANMAPP06	This is the ACB name of the ANM GATE CTCP which handles switched virtual circuits and manages data transfer on X.25 links, shared with other applications.
ANMAPP07	This is the optional ACB name of the ANM LU6.2 handler. You must estimate the EAS, AUTOSESS, DSESLIM, DMINWL, and DMINWR parameters based on your environment.

Connect:Express Logmodes

The logmode must be specified for LU6.2 Partners and can be specified for 3270 TEDPC and SNA application-to-application Partners in the COS table. LOGTOM is the logmode table example with standard ANM logmodes. The screen below shows examples of standard ANM logmodes.

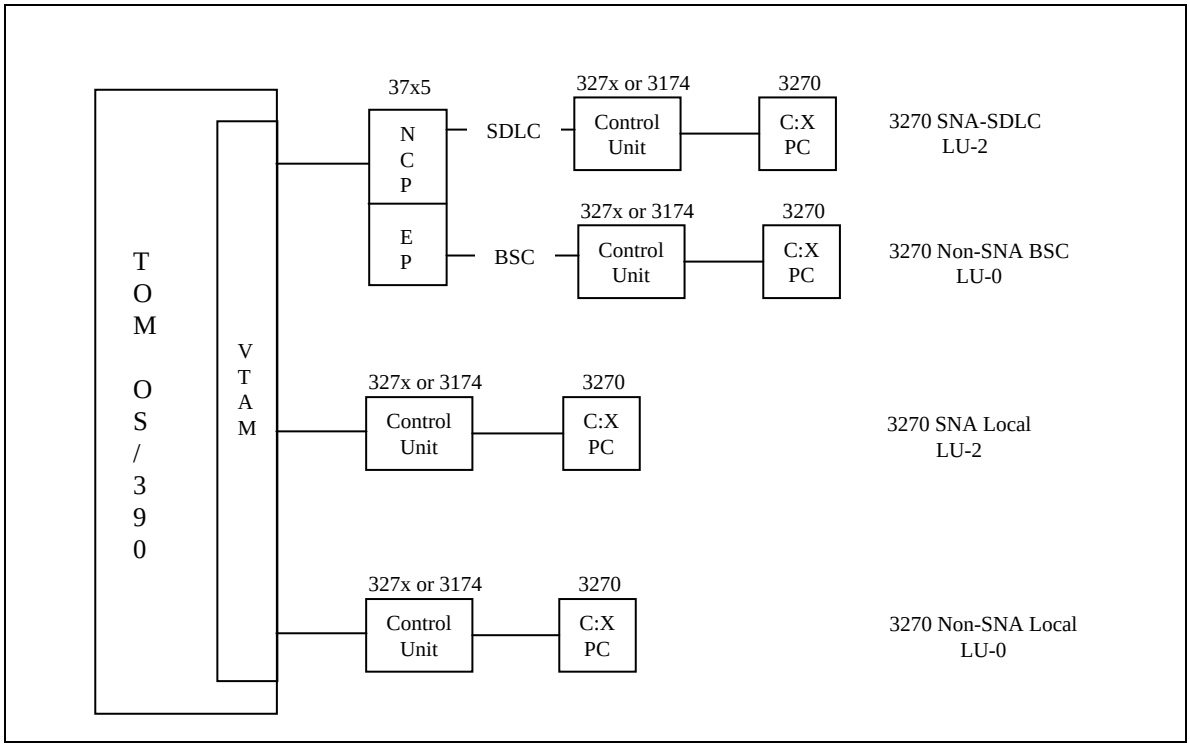
```
LOGTOM    MODETAB
*************************************************************************
*              ANM SNA APPLICATION-TO-APPLICATION LOGMODE               *
SNAHND01 MODEENT LOGMODE=SNAHND01,                                      X
               FMPROF=X'03',                                            X
               TSPROF=X'03',                                            X
               PRIPROT=X'B0',                                           X
               SECPROT=X'B0',                                           X
               SRCVPAC=X'02',                                           X
               SSNDPAC=X'02',                                           X
               PSNDPAC=X'02',                                           X
               COMPROT=X'0040'
*************************************************************************
*              ANM LU6.2 HANDLER LOGMODE                                *
*              FOR INDEPENDENT LU                                       *
LU6P2I01  MODEENT LOGMODE=LU6P2I01,                                     X
               FMPROF=X'13',                                            X
               TSPROF=X'07',                                            X
               PRIPROT=X'B0',                                           X
               SECPROT=X'B0',                                           X
               COMPROT=X'50B1',                                         X
               RUSIZES=X'8585',                                         X
               TYPE=X'00',                                              X
               PSERVIC=X'060200000000000000000300',                     X
               ENCR=B'0000'
```

```
*********************************************************************
*              ANM LU6.2 HANDLER LOGMODE                            *
*              FOR INDEPENDENT LU                                   *
LU6P2I02  MODEENT LOGMODE=LU6P2I02,                                  X
              FMPROF=X'13',                                          X
              TSPROF=X'07',                                          X
              PRIPROT=X'B0',                                         X
              SECPROT=X'B0',                                         X
              COMPROT=X'50B1',                                       X
              RUSIZES=X'8989',                                       X
              TYPE=X'00',                                            X
              PSERVIC=X'060200000000000000000300',                   X
              ENCR=B'0000'
*********************************************************************
*              ANM LU6.2 HANDLER LOGMODE                            *
*              FOR DEPENDENT LU                                     *
LU6P2D01  MODEENT LOGMODE=LU6P2D01,                                  X
              FMPROF=X'13',                                          X
              TSPROF=X'07',                                          X
              PRIPROT=X'B0',                                         X
              SECPROT=X'B0',                                         X
              COMPROT=X'50B1',                                       X
              RUSIZES=X'8585',                                       X
              TYPE=X'00',                                            X
              PSERVIC=X'060200000000000000000000',                   X
              ENCR=B'0000'
*********************************************************************
*              ANM LU6.2 HANDLER LOGMODE                            *
*              FOR DEPENDENT LU                                     *
LU6P2D02  MODEENT LOGMODE=LU6P2D02,                                  X
              FMPROF=X'13',                                          X
              TSPROF=X'07',                                          X
              PRIPROT=X'B0',                                         X
              SECPROT=X'B0',                                         X
              COMPROT=X'50B1',                                       X
              RUSIZES=X'8989',                                       X
              TYPE=X'00',                                            X
              PSERVIC=X'060200000000000000000000',                   X
              ENCR=B'0000'
          MODEEND
          END
```

3270 Terminal Connections for Connect:Express PC

The following information is needed by users who have installed both Connect:Express OS/390 and Connect:Express /TED PC. All of the DLOGMODs are standard IBM logmodes and are set in ISTINCLM.

The term name is the terminal LU name for SSCP. You set it in the Connect:Express Partner directory when you define the PC 3270 Partner. The figure below depicts the PC connection types and displays the 3270 terminal connections supported by Connect:Express.



The control unit shown in the PC Connection Type can be real, PC-emulated, or a black box such as a protocol converter. The OS/390 Connect:Express expects a 3270 Extended Data Stream from the PC connection.

Terminal Type Definitions

The 3270 SNA-SDLC and 3270 BSC non-SNA terminal types are defined by the NCP/EP generation input stream. The 3270 SNA local and 3270 non-SNA local terminal types are defined in the VTAMLST major nodes.

Alternate screen terminal usage can improve file transfer throughput. Each terminal type is described in the following table.

Terminal Type	Description
3270 SNA-SDLC	Defined by the NCP/EP generation input stream. including:
	LINE ...
	PU
(*termname*)	LU ...,DLOGMOD=D4C3278x,...
3270 Non-SNA BSC	Defined by the NCP/EP generation input stream, including:
	LINE...
	CLUSTER...
(*termname*)	TERMINAL ...,DLOGMOD=D4B3278x,...
3270 SNA-SDLC Local	Defined in a VTAMLST major node and the node is defined as:
	VBUILD TYPE=LOCAL (cluster with terminal)
	PU ...
(*termname*)	LU ...,DLOGMOD=D4A3278x,...
3270 Non-SNA Local	Defined in a VTAMLST major node and the node is defined as:
	LBUILD ... (direct with terminal)
(*termname*)	LOCAL ...,DLOGMOD=D4B3278x,...

Definition of a Switched Major Node

The following example illustrates an X.25 switched major node definition.

```
VMCH130  VBUILD TYPE=SWNET
C13X192  PU    ADDR=C1,PUTYPE=1,DISCNT=YES,                          *
               MAXDATA=4096,IDBLK=003,IDNUM=03192
V13X1920 LU    LOCADDR=0,SSCPFM=FSS,PACING=(1,1),LOGTAB=INTPSRTM
C13X194  PU    ADDR=C1,PUTYPE=1,DISCNT=YES,                          *
               MAXDATA=4096,IDBLK=003,IDNUM=03194
V13X1940 LU    LOCADDR=0,SSCPFM=FSS,PACING=(1,1),LOGTAB=INTPSRTM
C13X196  PU    ADDR=C1,PUTYPE=1,DISCNT=YES,                          *
               MAXDATA=4096,IDBLK=003,IDNUM=03196
V13X1960 LU    LOCADDR=0,SSCPFM=FSS,PACING=(1,1),LOGTAB=INTPSRTM
C13X198  PU    ADDR=C1,PUTYPE=1,DISCNT=YES,                          *
               MAXDATA=4096,IDBLK=003,IDNUM=03198
V13X1980 LU    LOCADDR=0,SSCPFM=FSS,PACING=(1,1),LOGTAB=INTPSRTM
C13X19A  PU    ADDR=C1,PUTYPE=1,DISCNT=YES,                          *
               MAXDATA=4096,IDBLK=003,IDNUM=0319A
V13X19A0 LU    LOCADDR=0,SSCPFM=FSS,PACING=(1,1),LOGTAB=INTPSRTM
C13X19C  PU    ADDR=C1,PUTYPE=1,DISCNT=YES,                          *
               MAXDATA=4096,IDBLK=003,IDNUM=0319C
V13X19C0 LU    LOCADDR=0,SSCPFM=FSS,PACING=(1,1),LOGTAB=INTPSRTM
C13X19E  PU    ADDR=C1,PUTYPE=1,DISCNT=YES,                          *
               MAXDATA=4096,IDBLK=003,IDNUM=0319E
V13X19E0 LU    LOCADDR=0,SSCPFM=FSS,PACING=(1,1),LOGTAB=INTPSRTM
C13X1A0  PU    ADDR=C1,PUTYPE=1,DISCNT=YES,                          *
               MAXDATA=4096,IDBLK=003,IDNUM=031A0
V13X1A00 LU    LOCADDR=0,SSCPFM=FSS,PACING=(1,1),LOGTAB=INTPSRTM
```

The switched major node must have as many PU/LU pairs as the number of SVCs on the corresponding X.25 link. The IDNUM values are the default values computed during generation of the NCP.

Note: The LOGTAB parameter specifying the Interpret table name is necessary and only valid for PAD links on MCH in GATE=GENERAL or GATE=NO.

Definition of the Interpret Table

An interpret table is used only when a PAD Partner calls Connect:Express on an X.25 link, not a DATE link. The following example shows the Interpret Table Definition.

```
INTPSRTM INTAB
              LOGCHAR APPLID=(APPLICID,ANMAPP01),SEQNCE=' PESIT'
              LOGCHAR APPLID=(APPLICID,ANMAPP01),SEQNCE=' A   '
              LOGCHAR APPLID=(APPLICID,ANMAPP01),SEQNCE=' R   '
              ENDINTAB
              END
```

The first byte of the SEQNCE parameter value must be binary 0 (hex on). The PAD Partner can call on any X.25 link, even one not seen by the ANM.

Definition of X25NPSI

An X25NPSI definition is shown in the following example. In the definition of MCH130, the parameter PAD=TRANSP enables PAD links on this X.25 link (GATE=GENERAL). MCH131 is dedicated to the ANM. Connect:Express supports PCNE and PAD LLC types.

```
Example:
*************************************************************************
*                                                                       *
*---------------------------------------------------------------------- *
* ADR  LINE    LU MCH   SPEED CVC CVP NUMBER     USER        SWITCHED  *
*---------------------------------------------------------------------- *
* 022  MCH130  XU13022   9600   8  1  192081379  PSR GATE              *
* 023  MCH131  XU13023   9600   8  1  192081926  PSR DATE              *
*                                                                       *
*************************************************************************
      X25BUILD MCHCNT=9,          NUMBER OF MCHS                      *
               IDNUMH=03,         IDNUM FOR PAD = 00XXX               *
               SNAP=NO,           NO INTERNAL TRACE                   *
               TYPSYS=OS,         SYSTEM OS/VS                        *
               MODEL=3725,        COMM. CONTROLLER NAME               *
               MAXPIU=4K,                                             *
               VERSION=V4
      X25NET   NETTYPE=1,         TRANSPAC NETWORK                    *
               CPHINDX=4,         4 MACRO X25VCCPT                    *
               OUHINDX=2,         2 MACRO X25OUFT                     *
               DM=YES                LAPB
      X25VCCPT INDEX=1,           DEFINE VCCPTIT ENTRY                *
               MAXPKTL=128,       MAX PACKET LENGTH                   *
               VWINDOW=3          TRANSMIT/RECEIVE WINDOW
               .
               .
      X25OUFT  INDEX=1            DEFINE OUFTIT ENTRY
               .
               .
```

Continued

```
*-------------------------------------------------------------------*
*                                                                   *
*                       MCH130-ADDRESS 022     -                    *
*                                                                   *
*     CHARACTERISTICS:  SVC NUMBER.........    8                   *
*                       RATE................ 9600                  *
*                       WINDOW (FRAME)......    7                   *
*                       DELAY............... 1600                  *
*                       WINDOW (PACKET).....    3                   *
*                       PACKET SIZE.........  128                   *
*                       TAXATION RULE.......  NON                   *
*                                                                   *
*-------------------------------------------------------------------*
MCH130 X25MCH ADDRESS=022,        LINE INTERFACE ADDRESSES          *
               LCGDEF=0(7),       LOGICAL CHANNEL GROUP IS 0 TO 7  *** *
               FRMLGTH=131,       FRAME LENGTH  (128 + 3)           *
               MWINDOW=7,         FRAME WINDOW                      *
               LLCLIST=(LLC0,LLC2,LLC3,LLC4,LLC5), PCNE,PSH,GATE  PAD  *
               GATE=GENERAL,      GATE FUNCTION SUPPORTED           *
               PAD=TRANSP,        TERMINALS SUPPORTED VIA THE PAD   *
               SUBADDR=YES,       SUBADDRESSING IS USED ON THIS LINK  *
               LLC0=0,            PCNE                              *
               LLC2=2,            PSH                               *
               LLC3=3,            QLLC                              *
               LLC4=4,            GATE                              *
               LLC5=5,            PAD                               *
               TPTIMER=2.5,       X25 T1 TIMER IN SECONDS           *
               TDTIMER=2,         TIMER TO WAIT FOR ND RETRIES      *
               NPRETRY=10,        NUMBER OF RETRIES WHEN TPTIMER ELAPSE*
               NDRETRY=2,         NUMBER OF (TP*NP) RETRIES         *
               DBIT=NO,           D BIT NOT USED IN PCNE            *
               LUNAME=XU13022,    MCH LU NAME                       *
               PUNAME=XP13022,    MCH PU NAME                       *
               LCN0=USED,         LOGIC PATH   0 USED WITH XPAC     *
               NCPGRP=G13X250,                                      *
               PKTMODL=8,         SEQUENCE PACKETS:    MODULO 8     *
               PROTCOL=LAPB,      LAPB RECOMMEND.                   *
               STATION=DTE,       THIS PHYSICAL LINK IS A DTE       *
               ANS=CONT,                                            *
               OWNER=HOST02,                                        *
               TRAN=NO            NO ASCII TRANSLATION
LCG0    X25LCG LCGN=0             LCG NUMBER FOR SUBSEQUENT VC
**********************************************************************
*                                                                    *
*    LOGICAL CHANNEL 0-07  CVC                                       *
*                                                                    *
**********************************************************************
*
L13X250 X25VC  LCN=(0,7),          LOGICAL CHANNEL          **      *
               VCCINDX=1,          ENTRY IN VCCPTIT TABLE           *
               IDNUMT=A02E,        IDNUM ATTRIBUTION                *
               TYPE=SWITCHED,      SVCs                             *
               CALL=INOUT,         INCOMING AND OUTCOMING CALLS     *
               MAXLU=1,            MAX LU NUMBER                    *
               NCPGRP=G13CVC0,     CVC GROUP NAME                   *
               OWNER=HOST02,                                        *
               OUFINDX=1           ENTRY IN OUFTIT TABLE
               .
```

Continued

```
*--------------------------------------------------------------------*
*                                                                    *
*                       MCH131-ADDRESS 023     -                     *
*                                                                    *
*     CHARACTERISTICS:   SVC NUMBER.........    8                    *
*                        RATE................ 9600                   *
*                        WINDOW (FRAME)......    7                    *
*                        DELAY............... 1600                   *
*                        WINDOW (PACKET).....    3                    *
*                        PACKET SIZE.........  128                    *
*                        TAXATION RULE.......  YES                    *
*                                                                    *
*--------------------------------------------------------------------*
MCH131 X25MCH  ADDRESS=023,       LINE INTERFACE ADDRESSES             *
               LCGDEF=0(7),       LOGICAL CHANNEL GROUP IS 0 TO 7      *
               FRMLGTH=131,       FRAME LENGTH  (128 + 3)              *
               MWINDOW=7,         FRAME WINDOW                         *
               LLCLIST=(LLC0,LLC5),      PCNE,PSH,PAD                  *
               GATE=DEDICAT,      DATE FUNCTION SUPPORTED              *
               PAD=TRANSP,        PAD CMNDS UP TO USER                 *
               TPTIMER=2.5,       X25 T1 TIMER IN SECONDS              *
               TDTIMER=2,         TIMER TO WAIT FOR ND RETRIES         *
               NPRETRY=10,        NUMBER OF RETRIES WHEN TPTIMER ELAPSE*
               NDRETRY=2,         NUMBER OF (TP*NP) RETRIES            *
               DBIT=NO,           D BIT NOT USED IN PCNE               *
               LUNAME=XU13023,    MCH LU NAME                          *
               PUNAME=XP13023,    MCH PU NAME                          *
               LCN0=USED,         LOGIC PATCH  0 USED WITH XPAC        *
               NCPGRP=G13X251,                                         *
               PKTMODL=8,         SEQUENCE PACKETS:    MODULO 8        *
               PROTCOL=LAPB,      LAPB RECOMMEND.                      *
               STATION=DTE,       THIS PHYSICAL LINK IS A DTE          *
               ANS=CONT,                                               *
               OWNER=HOST02,                                           *
               TRAN=NO            NO ASCII TRANSLATION
LCG1    X25LCG LCGN=0             LCG NUMBER FOR SUBSEQUENT VC
**********************************************************************
*    LOGICAL CHANNEL 0-07  CVC                                       *
**********************************************************************
L13X251 X25VC  LCN=(0,7),          LOGICAL CHANNEL          **       *
               VCCINDX=1,          ENTRY IN VCCPTIT TABLE           *
               IDNUMT=A01E,        IDNUM ATTRIBUTION                *
               TYPE=SWITCHED,      CVC                              *
               CALL=INOUT,         INCOMING AND OUTGOING CALL (DATE)*
               MAXLU=1,            MAX LU NUMBER                    *
               NCPGRP=G13CVC1,     CVC GROUP NAME                   *
               OWNER=HOST02,                                        *
               OUFINDX=1           ENTRY IN OUTFIT TABLE
               .
**********************************************************************
*    X25END MACRO                                                    *
**********************************************************************
       X25END  LSTUACB=YES,X25VTAM=YES
       END
```

Appendix F

JCL Files for the Connect:Express Monitor

This appendix describes the DD Names used by Connect:Express in different address spaces.

DDNAMES

The following tables describe the JCL files that Connect:Express uses for initialization, processing, and logging. The DDNAMES are listed for each type of address space.

DDNames for the TOM Address Space

DDname	Description
SYSIN	A card-type file which contains Connect:Express and ANM execution parameters. See Appendix D for detailed information about the SYSIN file.
CXPLEX	A card type file which contain SYSPLEX configuration parameters. See Chapter 1 of this guide for more information about the Sysplex Environment.
SYSFIL File directory	The File directory in the form of a VSAM KSDS file. It is initialized during installation of Connect:Express, and it contains characteristics about the data to be transferred.
SYSPAR Partner directory	A Partner directory in the form of a VSAM KSDS file. It is initialized during installation of Connect:Express and contains information about other Connect:Express hosts that your host can do transfers with.
SYSCHK	A checkpoint file in the form of a sequential file. It is initialized during the installation of Connect:Express, and is used by Connect:Express to write its internal tables each time they are modified. SYSCHK is read during a hot start to restore tables to their state before interruption. To increase integrity, this file can be duplicated by the DDNAME file *SYSCHK2* and allocated to another volume.
SYSLIB	An APF-authorized library of programs used by Connect:Express.
SYSLOG	A trace of transfer events in the form of a sequential file. It is initialized during installation of Connect:Express. You can use a wrap-around file or a SYSOUT file. The *SYSLOG* records all the events related to requests and transfers. See Appendix B for a list of SYSLOG messages issued by the monitor.
SYSMSG	A SYSOUT file that contains messages sent by Connect:Express. These messages can be generated during Connect:Express initialization, when it stops, or when an error occurs. See Appendix B for a list of WTO messages issued by the monitor.

DDNames for the TOM Address Space (Continued)

DDname	Description
SYSPRT	A SYSOUT file where the processed SYSIN cards are summarized.
SYSPRTX	A SYSOUT file where the processed CXPLEX cards are summarized.
SYSRCY	A sequential file that is initialized during Connect:Express installation. It records requests from local users when Connect:Express is not up. This file is called the Recovery file.
SYSPRM	A PDS file which contains members used for PDS unload using member selection and Partner lists.
SYSEVT	Describes a card-type file. It contains planned transfer requests and can be empty. See the member #EVENT in *PARMLIB* for an example.
SYSJCL	Contains model JCL members that are executed during specific types of transfers.
SYSJNL	A VSAM RRDS file, initialized at the installation of Connect:Express. It is used by Connect:Express to write a record for each successful or unsuccessful transfer. You can view *SYSJNL* through Connect:Express ISPF panels, or execute clists to interrogate the journal.
SYSPCH	A SYSOUT file (INTRDR) which is read by JES, and used by Connect:Express to submit jobs to JES for execution.
SYSANM	A card-type file containing the ANM initialization parameters. The parameters are read and controlled by the Connect:Express Monitor program, PIB2P000 from the Connect:Express SYSIN file and written in the *SYSANM* file.
SYSSNA (optional)	A card-type file which combines LU name and Partner name parameters. This option can be implemented if an alternate LU name is used to accept incoming calls from this Partner or sends outgoing calls to this Partner. If the LU name is replaced by the asterisk (*) character, no incoming control is performed. It is also possible to remove controls on calls by Connect:Express/PC 3270 by coding in the first parameter and keyword (LU2BYPAS). See member #SYSSNA in *PARMLIB* for an example.
SYSX25 (optional)	A card-type file which combines all X.25 addressing parameters and the Partner name. This option can be implemented when alternate X.25 addresses are used to accept incoming calls from this Partner or to send outgoing calls to this Partner. If the X.25 address is replaced by the asterisk (*) character, no control is done for an incoming call from this Partner. See member #SYSX25 in *PARMLIB* for an example.
SYSTCP (optional)	A card-type file which combines all TCP addressing parameters and the Partner name. This option can be implemented when alternate TCP/IP addresses are used to accept incoming calls from this Partner or to send outgoing calls to this Partner. If the TCP address/host is replaced by the asterisk (*) character, no control is done for an incoming call from this Partner. See member #SYSTCP in *PARMLIB* for an example.
SYSINEXT (optional)	A card-type file used by L1B2PDIX standard exits driver, UEXJNL=L1B2PDIX (see *SAMPLIB* EX#DIX).
SYSUE1 (nonstandard option)	A card-type file used by the L1GFIUE1 optional file exit, included in the PAC option.
SYSEXEC (nonstandard option)	A PDS-type file used for a REXX user exit.
SYSTSPRT (nonstandard option)	A SYSOUT-type file used for a REXX output.

DDNames for the ANM Address Space

DDname	Description
SYSIN	A card file containing the ANM initialization parameters. This file is written by the TOM Address (SYSANM file) space before starting the ANM.

DDNames for the APM Address Space

DDname	Description
CHKMODEL	This template provides information about the structure of the checkpoint file and where to create it. The request number is included in the data set name of the checkpoint file.
SYSMSG	A SYSOUT file that contains messages sent by the APM. These messages can be generated during Connect:Express initialization, when it stops, or when an error occurs. See Appendix B for a list of WTO messages issued by the APM.
SYSLOG	A trace of transfer events in the form of a sequential file that is initialized during the installation of Connect:Express. You can use a wrap-around file or a SYSOUT file. The *SYSLOG* records all the events related to requests and transfers. The information found there supplements the TOM AS SYSLOG information. See Appendix B for a list of SYSLOG messages issued by the APM.
SYSPRINT	This is required when using the IDCAMS functions with the L1GFIUE1 utility.
SYSPR20	This is required when using the API L0B2Z20 module through the L1GFIUE1 utility.
SYSEXEC	This is required with a CLIST implementation.
SYSPROC	This is required when implementing the job submission functions of the L1GFIUE1 utility.
SYSUE1	A card-type file used by the L1GFIUE1 optional file exit. The L1GFIUE1 is included in the Utilities option.

DDNames for the AFM Address Space

DDname	Description
SYSLOG	Shows the execution parameters that the AFM processed from its SYSPARM file.
AFMLOG	A trace of FTP events in the form of a SYSOUT file. Do not change the DSCB. This information supplements the TOM AS SYSLOG information. The AFMLOG messages are described in the FTP Guide.
SYSPARM	The AFM SYSIN file. This is a card-type file which contains the AFM execution parameters.
MIFPARM	A card-type file which contains parameters for communications with the monitor. Do not change this file.

DDNames for the EAS Address Space

DDname	Description
SYSPRINT	Required when using IDCAMS functions from the L1GFIUE1 utility.
PARMFTPL	Used to customize the FTP list format.
SYSPR20	Required when using the API L0B2Z20 module through the L1GFIUE1 utility.
SYSEXEC	Required with user CLIST implementation.
SYSPROC	Required when implementing the job submission functions of the L1GFIUE1 utility.
SYSUE1	A card-type file used by the L1GFIUE1 optional file exit. The L1GFIUE1 is included in the Utilities option.

Appendix G

Using Protocol Traces

This appendix describes how to use the Auxiliary Trace Manager services.

The Auxiliary Trace Manager

The Auxiliary Trace Manager (ATM) is an address space that can be activated and deactivated by a command to the Connect:Express monitor. The trace manager records trace requests that you pass to the monitor through commands. A trace request determines which partner or group of partners to trace and whether the trace must include both negotiation commands and file transfer data or only negotiation commands.

“Running the Trace Manager” on page 6-4 describes how to configure the ATM and how to activate traces. This appendix describes the ATM messages file and the trace files.

ATM Messages

ATM messages are prefixed by the string 'ATM' and the Monitor subsystem number. The following list explains the messages you may find in the ATM SYSLOG file.

Code	Description
ATMx001I ATMx002I	NO SYSLOG DD CARD FOUND, WILL SEND WTO SYSLOG FILE OPENED Explanation: If no SYSLOG DD card is found messages are sent to the operator. System action: None. User action: None.
ATMx003E	ERROR LOADING MODULE XXXXXXXX Explanation: The ATM detected a loading error during initialization. System action: Initialization aborts. User action: Check the module name in the STEPLIB and call Support.
ATMx004E	ERROR IN PARMS XXXXXXXXXXXXXXXXXXXXXXXXX Explanation: The ATM detected an error in the EXEC parm during initialization. System action: Initialization aborts. User action: Check the parm field and call Support.

Code	Description
ATMx005E	GETMAIN ERROR Explanation: The ATM detected a getmain error during initialization. System action: Initialization aborts. User action: Check the REGION parameter of the JCL and call Support.
ATMx006I	EXEC PARMS: SSN=XXXX REQ=XXX Explanation: The ATM EXEC parameters are the subsystem name and the number of simultaneous trace requests: 128 is the default. System action: Starts initialization. User action: None.
ATMx007E	COMMUNICATION ERROR WITH TOM AS Explanation: The ATM is not able to communicate with the monitor. System action: Initialization aborts. User action: Check the SSN= parameter.
ATMx008E	ATM ABENDING CODE=XXXX Explanation: The ATM is abending. System action: ATM status is inactive. User action: Keep the dump, check the abend code with Support.
ATMx009I	ATM INITIALIZATION COMPLETE Explanation: The ATM is successfully initialized: you can activate traces. System action: ATM status is active, waiting for trace requests. User action: Request traces.
ATMx010I	ATM TERMINATED Explanation: The ATM terminated, Connect:Express will reject all trace requests. System action: ATM status is inactive. User action: None.
ATMx015I ATMx016I ATMx017I	TRACE REQUEST ACCEPTED ID=XXXXXXXX TYPE : PART - ??? MASK : ???????? Explanation: The ATM accepted a new trace request: it is identified by number XXXXXXX. This is a partner trace; the trace type is dialog or all, for the partner or the group of partners shown by the mask. System action: None. User action: Note the trace request ID for further deletion.
ATMx018I	TRACE REQUEST DELETED ID=XXXXXXXX Explanation: Trace request number XXXXXXX is disabled. System action: None. User action: None.
ATMx021E	L1ATMTRC ATTACH ERROR RC=..... Explanation: The trace program initialization failed. System action: ATM initialization aborts. User action: Call Support..
ATMx022I	L1ATMTRC INITIALIZATION COMPLETE Explanation: The trace program is successfully initialized. System action: Completes initialization. User action: None.
ATMx021E	L1ATMTRC ABNORMAL TERMINATION ..... Explanation: The trace program stopped, probably after an abend. System action: ATM stops. User action: Check the return code, keep the dump, and call Support.

Code	Description
ATMx030I ATMx031I	L1ATMTRC IS INITIALIZING L1ATMTRC IS TERMINATING Explanation: The trace program is initializing or terminating. System action: Initializes or terminates ATM. User action: None.
ATMx032I ATMx033I	TRACE FILE OPENED ..... TRACE FILE CLOSED ..... Explanation: All trace file open and close is shown, with the file name. System action: Execute trace. User action: Note the name of the file for further analysis.
ATMx034E	GETMAIN ERROR ffffffff .... Explanation: The trace program detected a getmain error while opening the trace file 'ffffffff'. System action: The trace will not be created, and the trace program will probably not be able to work any longer. User action: Check the problem with Support.
ATMx035E ATMx035I	ALLOCATION ERROR ffffffff R15 = ... ERROR CODES C1 = ... C2 = ... C3 = ... Explanation: The trace program detected an allocation error while initializing the trace file 'ffffffff'. The problem may happen if two files are allocated at the same time, with a name built from the time stamp. This is very unlkikely to happen. System action: The trace is not created. User action: Check the problem with Support.
ATMx036E ATMx037w ATMx038e ATMx039e	OPEN ERROR ffffffff.... FREEMAIN ERROR ffffffff.... CLOSE ERROR ffffffff DCB=0 WRITE ERROR ffffffff DCB=0 Explanation: The trace program detected an error while processing the trace file 'ffffffff'. System action: The current trace process is aborted. User action: Check with Support any error that happens while processing a trace file.

Analysing Trace Files

When a protocol session opens, Connect:Express queries the ATM to know if the current partner requires a trace, and which type of trace (dialog or all). If such a trace request exists, the ATM opens a SYSPRINT file the name of which is computed from the request number (Rnnnnnnn) or from the current time (Ahmmsscc). The ATM SYSOUT file shows the trace files and the SYSLOG file.

```
    PSRTOM4  PSRTOM4  DEV$400  STC02840 PSRTOM4    NS  FB 1573
    PSRAN4   PSRAN4   DEV$400  STC02841 PSRAN4     NS  FB 2144
    PSRAP4   PSRAP401 CEE$400  STC02842 PSRAP4     LO  FF 1361
    PSRAP4   PSRAP402 CEE$400  STC02843 PSRAP4     LO  FF 1328
  ? PSRAT4   PSRAT4   DEV$400  STC07291 PSRAT4     NS  FB 2332
```

```
    JESJCLIN                    1 PSRAT4   X                        2
    JESMSGLG JES2               2 PSRAT4   X                        2
    JESJCL   JES2               3 PSRAT4   X                       52
    JESYSMSG JES2               4 PSRAT4   X                        2
  $ INTTEXT  JES2               5 PSRAT4   A                        0
    SYSLOG   PSRAT4           104 PSRAT4   V                        0
    R0042562 PSRAT4           110 PSRAT4   A LOCAL                 83
    A9323570 PSRAT4           111 PSRAT4   A LOCAL                 84
    R0042601 PSRAT4           112 PSRAT4   A LOCAL                191
    R0042602 PSRAT4           113 PSRAT4   A LOCAL                188
    R0042604 PSRAT4           114 PSRAT4   A LOCAL                191
    R0042605 PSRAT4           115 PSRAT4   A LOCAL                106
    R0042607 PSRAT4           116 PSRAT4   A LOCAL                191
    R0042608 PSRAT4           117 PSRAT4   A LOCAL                188
    R0042610 PSRAT4           118 PSRAT4   A LOCAL                191
    R0042611 PSRAT4           119 PSRAT4   A LOCAL                188
```

The trace files are written in XML format. The general structure respects the following rules:

- Inbound and outbound messages are identified by a number. For example: <IN_nnnnn> </IN_nnnnn> or <DATA_IN_nnnnn> </DATA_IN_nnnnn>. Messages are associated with a time stamp <TIME>hh:mm:ss:cc </TIME>.
- Negotiation and transfer synchronization messages are delimited by <IN_nnnnn> and <OU_nnnnn>; file data messages are delimited by <DATA_IN_nnnnn> and <DATA_OU_nnnnn>.
- Protocol commands are delimited by <CMD> </CMD: they are developed according to the protocol syntax (PeSIT and Odette).
- Data messages are delimited by <MSG> </MSG>.
- Data and negotiation messages exchanged are displayed in a SNAP format, with three columns: hexadecimal view, EBCDIC view, and ASCII view.
- Connect:Express fields that are used during file transfer refer to the table shown in “Connect:Express Fields” on page G-16. They are displayed after negotiation is ended for the field.

A trace is structured as shown below:

```
<TRACE ID=R0042562>

  <prot_SERVER>  (or <prot_CLIENT>)prot = FTP/PESIT/OFTP/ETEBAC3

  <REQN>Request number</REQN><PNAM>Partner</PNAM><TCPA> TCP/IP Adress </TCPA>
  . . .  Fields used by Connect:Express . . .
  <TCPH>TCP/IP Host name</TCPH>

    <IN_nnnn >   nnnn from 0001
      <TIME>hh:mm:ss:cc</TIME>
      <CMD>
  Protocol command as received, in a SNAP format
      </CMD>
      <Command 1>
        <param1>value</param1>
        <param2>value</param2>
        <param3>value</param3>
        <paramn>value</paramn>
      </Command 1>
    </IN_nnnn >

    <OU_nnnn >
      <TIME>hh:mm:ss:cc</TIME>
      <CMD>
  Protocol command as sent, in a SNAP format
      </CMD>
      <Commande 2>
        <param1>value</param1>
        <param2>value</param2>
        <param3>value</param3>
        <paramn>value</paramn>
      </Commande 2>
    </OU_nnnn >

  <FAPI>Applicative name of the file</FAPI><FSIZ>Size of the file, from the
sender</FSIZ>
  . . .  Fields used by Connect:Express . . .
  <LPHN>Physical file name</LPHN>
  Next data messages are included if type=all

    <DAT_OU_nnnn >
      <TIME>hh:mm:ss:cc</TIME>
      <MSG>
  Data message as sent, in a SNAP format
      </MSG>
    </DATA_OU_nnnn >

  End of protocol dialog

  </prot_SERVER>  (or </prot_CLIENT>)

</TRACE>
```

Next paragraphs show an example of a trace file for each protocol.

PeSIT Private (SIT) Protocol

Example of trace, type=dialog, client mode. Some lines are truncated. The record length of the trace file is 170 characters.

```
<TRACE ID=R0044683>
<PESIT CLIENT>
<REQN>00044683</REQN><PNAM>O1X01847</PNAM><PROT>PESIT</PROT><TYPL>TCP</TYPL>
<TCPA>127.000.000.001</TCPA><TCPH></TCPH>
<OU_0001><TIME>08:12:34:79</TIME><CMD>
00000000 001B4020 00110303 03073704 03010737   06010107 03002010 160100
</CMD>
<CONNECT>
<PI003 L=003>030737</PI003>
<PI004 L=003>010737</PI004>
<PI006 L=001>01</PI006>
<PI007 L=003>002010</PI007>
<PI022 L=001>00</PI022>
</CONNECT>
</OU_0001>
<IN_0002><TIME>08:12:34:88</TIME><CMD>
00000000 000E4021 11120601 01070300 2010
</CMD>
<ACONNECT>
<PI006 L=001>01</PI006>
<PI007 L=003>002010</PI007>
</ACONNECT>
</IN_0002>
<TDIR></TDIR><LFID>SIT00011</LFID><LPHN>PSR$REC.PS.F080.MEGA</LPHN><FSIZ>0000000
<OU_0003><TIME>08:12:34:98</TIME><CMD>
00000000 007DC011 12000914 03030307 37040301   07370B01 0C0C0532 32323232 0D02AE
00000020 10010211 01011902 0FD01E0D 20015025   08534954 30303031 3128042A 0204EE
00000040 0E330C39 37313230 39303935 38353963   2C404040 40404040 40404040 404040
00000060 40404040 40404040 40404040 40404040   40404040 40404040 40400000 00
</CMD>
<CREATE>
<PI009 L=020></PI009>
<PI003 L=003>030737</PI003>
<PI004 L=003>010737</PI004>
<PI011 L=001>0C</PI011>
<PI012 L=005>3232323232</PI012>
<PI013 L=002>AE8B</PI013>
<PI016 L=001>02</PI016>
<PI017 L=001>01</PI017>
<PI025 L=002>0FD0</PI025>
<PI030 L=013></PI030>
<PI032 L=001>50</PI032>
<PI037 L=008>SIT00011</PI037>
<PI040 L=004></PI040>
<PI042 L=002>04EE</PI042>
<PI050 L=014></PI050>
<PI051 L=012>971209095859</PI051>
<PI099 L=044>àààààààààààààààààààààààààààààààààààà   </PI099>
</CREATE>
</OU_0003>
<IN_0004><TIME>08:12:35:14</TIME><CMD>
00000000 000FC030 11000203 00000019 020FD0
</CMD>
<ACREATE>
<PI002 L=003>000000</PI002>
<PI025 L=002>0FD0</PI025>
```

Continued

```
</ACREATE>
</IN_0004>
<NMGS>04048</NMGS><SYNC>00032+16</SYNC><FAPI>010737030737000C22222971209095859</
<OU_0005><TIME>08:12:35:19</TIME><CMD>
00000000 0006C014 1200
</CMD>
<ORF>
</ORF>
</OU_0005>
<IN_0006><TIME>08:12:35:22</TIME><CMD>
00000000 000BC033 11000203 000000
</CMD>
<AORF>
<PI002 L=003>000000</PI002>
</AORF>
</IN_0006>
<OU_0007><TIME>08:12:35:22</TIME><CMD>
00000000 0006C002 1200
</CMD>
<WRITE>
</WRITE>
</OU_0007>
<IN_0008><TIME>08:12:35:22</TIME><CMD>
00000000 000EC036 11000203 00000012 0100
</CMD>
<AWRITE>
<PI002 L=003>000000</PI002>
<PI018 L=001>00</PI018>
</AWRITE>
</IN_0008>
<FBYT>000000001000</FBYT><FNRB>000000012798</FNRB>
<OU_0009><TIME>08:12:43:04</TIME><CMD>
00000000 0006C008 1200
</CMD>
<TRANSEND>
</TRANSEND>
</OU_0009>
<IN_0010><TIME>08:12:43:63</TIME><CMD>
00000000 000BC037 11000203 000000
</CMD>
<ATRANSEND>
<PI002 L=003>000000</PI002>
</ATRANSEND>
</IN_0010>
<OU_0011><TIME>08:12:43:63</TIME><CMD>
00000000 000BC015 12000203 000000
</CMD>
<CRF>
<PI002 L=003>000000</PI002>
</CRF>
</OU_0011>
<IN_0012><TIME>08:12:43:65</TIME><CMD>
00000000 000BC034 11000203 000000
</CMD>
<ACRF>
<PI002 L=003>000000</PI002>
</ACRF>
</IN_0012>
<OU_0013><TIME>08:12:43:67</TIME><CMD>
00000000 000BC013 12000203 000000
```

Continued

```
</CMD>
<DESELECT>
<PI002 L=003>000000</PI002>
</DESELECT>
</OU_0013>
<IN_0014><TIME>08:12:43:73</TIME><CMD>
00000000 000BC032 11000203 000000
</CMD>
<ADESELECT>
<PI002 L=003>000000</PI002>
</ADESELECT>
</IN_0014>
<OU_0015><TIME>08:12:43:91</TIME><CMD>
00000000 000B4023 12110203 000000
</CMD>
<RELEASE>
<PI002 L=003>000000</PI002>
</RELEASE>
</OU_0015>
<IN_0016><TIME>08:12:43:96</TIME><CMD>
00000000 00064024 1112
</CMD>
<RELCONF>
</RELCONF>
</IN_0016>
</PESIT CLIENT>
</TRACE>
```

Continued

PeSIT Public Protocol

Example of trace, type=dialog, client mode. Some lines are truncated, the record length of the trace file is 170 characters.

```
<TRACE ID=R0044687>
<PESIT CLIENT>
<REQN>00044687</REQN><LNAM>GFIPSR4</LNAM><TYPP>OTHER</TYPP><PNAM>GFIPSR4</PNAM><
<TCPA>010.020.129.003</TCPA><TCPH>MVSB.CSG.STERCOMM.COM</TCPH>
<OU_0001><TIME>08:17:22:29</TIME><CMD>
00000000 D7C5E2C9 E3404040 C7C6C9D7 E2D9F440   D7E2D940 40404040
</CMD>
</OU_0001>
<IN_0002><TIME>08:17:22:34</TIME><CMD>
00000000 C1C3D2F0
</CMD>
</IN_0002>
<OU_0003><TIME>08:17:22:34</TIME><CMD>
00000000 00304020 00110307 47464950 53523404   07474649 50535234 05085053 522020
00000020 20200601 02070300 40031601 02170101
</CMD>
<CONNECT>
<PI003 L=007>GFIPSR4</PI003>
<PI004 L=007>GFIPSR4</PI004>
<PI005 L=008>PSR     </PI005>
<PI006 L=001>02</PI006>
<PI007 L=003>004003</PI007>
<PI022 L=001>02</PI022>
<PI023 L=001>01</PI023>
</CONNECT>
</OU_0003>
<IN_0004><TIME>08:17:22:35</TIME><CMD>
00000000 001B4021 11120508 50535220 20202020   06010207 03004003 170101
</CMD>
<ACONNECT>
<PI005 L=008>PSR     </PI005>
<PI006 L=001>02</PI006>
<PI007 L=003>004003</PI007>
<PI023 L=001>01</PI023>
</ACONNECT>
</IN_0004>
<TDIR>T</TDIR><TYPD>EBCDIC</TYPD><LPHN>PSR$REC.PS.F080.SHORT</LPHN><FSIZ>0000000
<FLAB>PSR$REC.PS.F080.SHORT</FLAB>
<USDS>pi99 répertoire</USDS>
<OU_0005><TIME>08:17:22 58</TIME><CMD>
00000000 0183C011 1200091C 03074746 49505352   34040747 46495053 52340B01 000C05
00000020 3144454C 0D02AE8F 10010111 01011902   07FE1E1A 20015025 15505352 245245
00000040 2E50532E 46303830 2E53484F 52542803   2A013932 1C330C30 35313232 313038
00000060 37323234 0C303030 30303030 30303030   303D0747 46495053 52343E07 474649
00000080 53523463 FE706939 392072E9 70657274   6F697265 20202020 20202020 202020
000000A0 20202020 20202020 20202020 20202020   20202020 20202020 20202020 202020
000000C0 20202020 20202020 20202020 20202020   20202020 20202020 20202020 202020
000000E0 20202020 20202020 20202020 20202020   20202020 20202020 20202020 202020
00000100 20202020 20202020 20202020 20202020   20202020 20202020 20202020 202020
00000120 20202020 20202020 20202020 20202020   20202020 20202020 20202020 202020
00000140 20202020 20202020 20202020 20202020   20202020 20202020 20202020 202020
00000160 20202020 20202020 20202020 20202020   20202020 20202020 20202020 202020
00000180 202020
</CMD>
```

Continued

```
<CREATE>
<PI009 L=028></PI009>
<PI003 L=007>GFIPSR4</PI003>
<PI004 L=007>GFIPSR4</PI004>
<PI011 L=001>00</PI011>
<PI012 L=005>F1DEL</PI012>
<PI013 L=002>AE8F</PI013>
<PI016 L=001>01</PI016>
<PI017 L=001>01</PI017>
<PI025 L=002>07FE</PI025>
<PI030 L=026></PI030>
<PI032 L=001>50</PI032>
<PI037 L=021>PSR$REC.PS.F080.SHORT</PI037>
<PI040 L=003></PI040>
<PI042 L=001>39</PI042>
<PI050 L=028></PI050>
<PI051 L=012>051221081722</PI051>
<PI052 L=012>000000000000</PI052>
<PI061 L=007>GFIPSR4</PI061>
<PI062 L=007>GFIPSR4</PI062>
<PI099 L=254>pi99 répertoire
</CREATE>
</OU_0005>
<IN_0006><TIME>08:17:22 82</TIME><CMD>
00000000 0113C030 11000203 0000000D 02AE9019   0207FE63 FE706939 392072E9 706572
00000020 6F697265 20202020 20202020 20202020   20202020 20202020 20202020 202020
00000040 20202020 20202020 20202020 20202020   20202020 20202020 20202020 202020
00000060 20202020 20202020 20202020 20202020   20202020 20202020 20202020 202020
00000080 20202020 20202020 20202020 20202020   20202020 20202020 20202020 202020
000000A0 20202020 20202020 20202020 20202020   20202020 20202020 20202020 202020
000000C0 20202020 20202020 20202020 20202020   20202020 20202020 20202020 202020
000000E0 20202020 20202020 20202020 20202020   20202020 20202020 20202020 202020
00000100 20202020 20202020 20202020 20202020   202020
</CMD>
<ACREATE>
<PI002 L=003>000000</PI002>
<PI013 L=002>AE90</PI013>
<PI025 L=002>07FE</PI025>
<PI099 L=254>pi99 répertoire
</ACREATE>
</IN_0006>
<USDR>pi99 répertoire</USDR>
<FAPI>GFIPSR4                GFIPSR4                          F1DEL         0512
<NMGS>02046</NMGS><SYNC>00064+03</SYNC><FRLG>00080</FRLG>
<OU_0007><TIME>08:17:22 83</TIME><CMD>
00000000 0006C014 1200
</CMD>
<ORF>
</ORF>
</OU_0007>
<IN_0008><TIME>08:17:22 84</TIME><CMD>
00000000 000BC033 11000203 000000
</CMD>
<AORF>
<PI002 L=003>000000</PI002>
</AORF>
</IN_0008>
<OU_0009><TIME>08:17:22 84</TIME><CMD>
00000000 0006C002 1200
```

Continued

```
</CMD>
<WRITE>
</WRITE>
</OU_0009>
<IN_0010><TIME>08:17:22 84</TIME><CMD>
00000000 000EC036 11000203 00000012 0100
</CMD>
<AWRITE>
<PI002 L=003>000000</PI002>
<PI018 L=001>00</PI018>
</AWRITE>
</IN_0010>
<TYPC>01</TYPC><COMP>NOCOMP.</COMP>
<OU_0011><TIME>08:17:22 87</TIME><CMD>
00000000 000DC008 12001B02 06E01C01 16
</CMD>
<TRANSEND>
<PI027 L=002>06E0</PI027>
<PI028 L=001>16</PI028>
</TRANSEND>
</OU_0011>
<FBYT>000000000002</FBYT><FNRB>000000000022</FNRB>
<IN_0012><TIME>08:17:23 12</TIME><CMD>
00000000 0012C037 11000203 0000001B 0206E01C   0116
</CMD>
<ATRANSEND>
<PI002 L=003>000000</PI002>
<PI027 L=002>06E0</PI027>
<PI028 L=001>16</PI028>
</ATRANSEND>
</IN_0012>
<OU_0013><TIME>08:17:23 12</TIME><CMD>
00000000 000BC015 12000203 000000
</CMD>
<CRF>
<PI002 L=003>000000</PI002>
</CRF>
</OU_0013>
<IN_0014><TIME>08:17:23 14</TIME><CMD>
00000000 000BC034 11000203 000000
</CMD>
<ACRF>
<PI002 L=003>000000</PI002>
</ACRF>
</IN_0014>
<OU_0015><TIME>08:17:23 14</TIME><CMD>
00000000 000BC013 12000203 000000
</CMD>
<DESELECT>
<PI002 L=003>000000</PI002>
</DESELECT>
</OU_0015>
<IN_0016><TIME>08:17:23 27</TIME><CMD>
00000000 000BC032 11000203 000000
</CMD>
```

Continued

```
<ADESELECT>
<PI002 L=003>000000</PI002>
</ADESELECT>
</IN_0016>
<OU_0017><TIME>08:17:23:48</TIME><CMD>
00000000 000B4023 12110203 000000
</CMD>
<RELEASE>
<PI002 L=003>000000</PI002>
</RELEASE>
</OU_0017>
<IN_0018><TIME>08:17:23:53</TIME><CMD>
00000000 00064024 1112
</CMD>
<RELCONF>
</RELCONF>
</IN_0018>
</PESIT CLIENT>
</TRACE>
```

FTP Protocol

Example of trace, type=all, server mode. Some lines are truncated, the record length of the trace file is 170 characters.

The FTP list is considered both as part of the dialog (<Lnnnn> </Lnnnn>, and part of data transfer (<DATA_OU_nnnnn>) that is only included if type=all. <MODL> variable indicates the list structure that is used for building the list records if it was found in the PARMFTPL file.

```
<TRACE ID=A8580827>
<FTP SERVER>
<PNAM>FTP4</PNAM><TCPA>010.087.015.082</TCPA><TCPH></TCPH>
<IN_0001><TIME>08:58:08:28</TIME><CMD>USER FTP4  </CMD></IN_0001>
<OU_0002><TIME>08:58:08:28</TIME><CMD>331 FTP4     password please ?
<IN_0003><TIME>08:58:08:43</TIME><CMD>PASS ftp4psw </CMD></IN_0003>
<USER>FTP4</USER><DFID>FTPGCZ</DFID><DPHN>&EXTDSN</DPHN>
<OU_0004><TIME>08:58:09:17</TIME><CMD>230-FTP4     User logged on at Connect:Exp
<IN_0005><TIME>08:58:09:34</TIME><CMD>CWD FTPF </CMD></IN_0005>
<OU_0006><TIME>08:58:09:34</TIME><CMD>200 CWD  Command received.
<IN_0007><TIME>08:58:09:49</TIME><CMD>PWD </CMD></IN_0007>
<OU_0008><TIME>08:58:09:49</TIME><CMD>257 "FTPGCZ   " is current profile.
<IN_0009><TIME>08:58:09:65</TIME><CMD>HELP </CMD></IN_0009>
<OU_0010><TIME>08:58:09:65</TIME><CMD>214-The Connect:Express FTP commands are:
 PASS, PASV, PORT,  PWD, QUIT,*REIN,*REST, RETR, *RMD  214-*RNFR,*RNTO, SITE,*SM
implemented  214-  214-TYPE may be ASCII, EBCDIC, IMAGE  214-STRU may be RECORD,
PE, *LIST, *PWD, RETR, SITE, STOR, STOU, *TRC  214-  214-CONNECT:Express keyword
from monitor.  214-Profile = symbolic file name defined in the monitor directory
<IN_0011><TIME>08:58:11:43</TIME><CMD>TYPE A </CMD></IN_0011>
<OU_0012><TIME>08:58:11:43</TIME><CMD>200 Data type is ASCII , Format is NON PRI
<IN_0013><TIME>08:58:11:59</TIME><CMD>PORT 10,87,15,82,16,125 </CMD></IN_0013>
<OU_0014><TIME>08:58:11:59</TIME><CMD>200 PORT Command executed.
<IN_0015><TIME>08:58:11:74</TIME><CMD>LIST </CMD></IN_0015>
<PNAM>FTP4</PNAM><USER>FTP4</USER>
<MODL>&MBX.  &DDN.                        &NOT                                &FLG
<OU_0016><TIME>08:58:11:92</TIME><CMD>125 LIST Command accepted.
<L0015>FTP4       $AA                          FTPSEND!PSR$REC.PS.F080.SHORT
<L0016>FTP4       $ABMAB                       FTPSEND!PSR$REC.PS.F080.SHORT
<L0017>FTP4       $APPLIC                      FTPSEND!PSR$REC.PS.F080.SHORT
<L0018>FTP4       $BENCMPH                     FTPSEND!PSR$REC.PS.F080.SHORT
<L0019>FTP4       $BENCMPM                     FTPSEND!PSR$REC.PS.F080.SHORT
<L0020>FTP4       $BENCMPV                     FTPSEND!PSR$REC.PS.F080.SHORT
<L0021>FTP4       $BENCMP0                     FTPSEND!PSR$REC.PS.F080.SHORT
<L0022>FTP4       $BENS01                      FTPSEND!PSR$REC.PS.F080.SHORT
<L0023>FTP4       $BENS02                      FTPSEND!PSR$REC.PS.F080.SHORT
<DATA_OU_0017><TIME>08:58:11:99</TIME><MSG>
00000000 46545034 20202020 20202024 41412020   20202020 20202020 20202020 202020
00000020 20202020 20202020 20204654 5053454E   44215053 52245245 432E5053 2E4630
00000040 302E5348 4F525420 20202020 20202020   20202020 20202020 20202020 202020

00004720 20202020 20202020 20202020 20202020   20202020 20202020 20202020 202020
00004740 20202020 20202020 20202020 20202020   20202020 20202020 20202052 4E5420
00004760 20202020 20202020 20204654 50342020   20202020 20202020 20200D0A 465450
00004780 20202020 20202044 4F4E
</MSG></DATA_OU_0017>
<OU_0018><TIME>08:58:14:29</TIME><CMD>250 list completed successfully.        </CMD>
<IN_0019><TIME>08:58:35:06</TIME><CMD>TYPE I </CMD></IN_0019>
<OU_0020><TIME>08:58:35:06</TIME><CMD>200 Data type is IMAGE
<IN_0021><TIME>08:58:35:22</TIME><CMD>PORT 10,87,15,82,16,126 </CMD></IN_0021>
<OU_0022><TIME>08:58:35:22</TIME><CMD>200 PORT Command executed.
<IN_0023><TIME>08:58:35:38</TIME><CMD>STOR psr$tst.&parttnid.&reqnumb </CMD></IN
<RFID>FTPGCZ</RFID><RPHN>psr$tst.&parttnid.&reqnumb </RPHN>
<REQN>00044723</REQN><LFID>FTPGCZ</LFID><LPHN>PSR$TST.FTP4ARTT.A0044723</LPHN>
<OU_0024><TIME>08:58:35:67</TIME><CMD>125 STOR Command accepted, transfer starti
<DATA_IN_0025><TIME>08:58:35:91</TIME><MSG>
00000000 47494638 3961C800 C800C41B 00C6EFFF   CEEFFFCE F7FFD6EF FFC6E7FF BDE7FF
00000020 E7FFD6F7 FFADDEFF A5DEFFB5 DEFFA5D6   FFDEF7FF E7FFFFEF FFFF9CD6 FF94D6
00000040 94CEFF8C CEFF84C6 FF7BC6FF 84CEFFDE   FFFF7BBD FF73BDFF F7FFFFBD EFFFFF
```

Continued

```
00000500 86D1790B B1C6F318 2BCA4D77 A684992A   E315B30D B8F295AE CC000063 A280BA
00000520 2A4AE013 4AD08CE2 1818BE28 587FF950   308DA281 43B90E59 D4BA13FD B048B3
00000540 1D693C02 40DE4B2E E6BA5BB1 6B780498   49D39E26 9D4CDA8E
</MSG></DATA_IN_0025>
<DATA_IN_0026><TIME>08:58:36:04</TIME><MSG>
00000000 33F7B266 8B2CC980 3E329399 F27AE3BC   181093EC F8A85440 72965D74 F997EF
00000020 044381FB 1818A5D2 FFBB6EA9 AD6F13A3   8AFC063A CE6A416C 88713961 516E94
000004E0 473B0715 31A43353 1C727223 056DAF53   5BBDC375 644848DB 3351A2F4 574CF3


00000520 77E9C23D 88087678 FFD62B66 87126C16   4862D81A E2E62C62 756BAED5 3B1463
00000540 700550C8 81883D71 114DE638 BFA7278C   A51E2918 19FA625F
</MSG></DATA_IN_0026>

<OU_0036><TIME>08:58:36:51</TIME><CMD>226-Transfer completed successfully.    22
6>
<IN_0037><TIME>08:58:38:55</TIME><CMD>QUIT </CMD></IN_0037>
<OU_0038><TIME>08:58:38:55</TIME><CMD>221 QUIT Command accepted, Goodbye.
</FTP SERVER>
</TRACE>
```

Etebac3 Protocol

Example of trace, type=all, client mode. Some lines are truncated, the record length of the trace file is 170 characters:

```
<TRACE ID=R0044715>
<ETEBAC3 CLIENT>
<REQN>00044715</REQN><TYPP>-ETB-</TYPP><PNAM>GFIPSR41</PNAM><PROT>ETEBAC</PROT><
<TCPA>010.020.129.003</TCPA><TCPH>MVSB.CSG.STERCOMM.COM</TCPH>
<FAPI>A   0080    ETEBRECVETEBAC3 PSR</FAPI>
<TDIR>T</TDIR><LFID>ETEBEMIS</LFID><LPHN>PSR$REC.PS.F080.SHORT</LPHN><FRLG>00080
<OU_0001><TIME>08:35:50:17</TIME><CMD>A   0080    ETEBRECVETEBAC3 PSR</CMD></OU_
<IN_0002><TIME>08:35:50:35</TIME><CMD>OK</CMD></IN_0002>
<DATA_OU_0003><TIME>08:35:50:38</TIME><MSG>
00000000 5BF9F461 F0F261F0 F340F1F8 7AF4F17A   F3F840D7 E2D9F0F0 F0F44040 F8F3C2
00000020 40E3D6D4 F440D5D9 C1C3C540 406140F9   F461F0F2 61F0F340 F1F87AF4 F17AF3
00000040 40D7E2D9 F0F0F0F4 4040F8F3 C2C14040
</MSG></DATA_OU_0003>
<DATA_OU_0004><TIME>08:35:50:39</TIME><MSG>
00000000 5CC1C161 D4D461D1 D140C8C8 7AD4D47A   E2E240E4 E4E4E4E4 E4E4E440 C3C3C3
00000020 40E2E2E2 E240D6D6 D6D6D6D6 406140C1   C161D4D4 61D1D140 C8C87AD4 D47AE2
00000040 40E4E4E4 E4E4E4E4 E440C3C3 C3C340E2
</MSG></DATA_OU_0004>


<DATA_OU_0024><TIME>08:35:50:40</TIME><MSG>
00000000 5CC7E2D6 F1E7E7E7 E7E7E7E7 E7404040   40404040 40404040 40404040 404040
00000020 40404040 40404040 40404040 40404040   40404040 40404040 40404040 404040
00000040 40404040 40404040 40404040 40404040
</MSG></DATA_OU_0024>
<FBYT>000000000002</FBYT><FNRB>000000000022</FNRB>
<OU_0025><TIME>08:35:50:40</TIME><CMD>FF</CMD></OU_0025>
<IN_0026><TIME>08:35:50:67</TIME><CMD>OKF</CMD></IN_0026>
</ETEBAC3 CLIENT>
</TRACE>
```

Example of trace, type=dialog, server mode:

```
<TRACE ID=R0051008>
<ETEBAC3 SERVER>
<REQN>00051008</REQN><TYPP>-ETB-</TYPP><PNAM>GFIPSR41</PNAM><PROT>ETEBAC</PROT><
<TCPA>010.020.129.003</TCPA><TCPH>MVSB.CSG.STERCOMM.COM</TCPH>
<IN_0001><TIME>04:54:25:46</TIME><CMD>A   0080    ETEBRECVETEBAC3 PSR</CMD></IN_
<FAPI>A   0080    ETEBRECVETEBAC3 PSR</FAPI>
<TDIR>R</TDIR><LFID>ETEBRECV</LFID><LPHN>PSR$TST.ETEB.GFIPSR41.D051222.A0051008<
<OU_0002><TIME>04:54:25:57</TIME><CMD>OK</CMD></OU_0002>
<IN_0003><TIME>04:54:25:83</TIME><CMD>FF</CMD></IN_0003>
<FBYT>000000000002</FBYT><FNRB>000000000022</FNRB>
<OU_0004><TIME>04:54:25:84</TIME><CMD>OKF</CMD></OU_0004>
</ETEBAC3 SERVER>
</TRACE>
```

Odette Protocol

Example of trace, type=all, client mode. Some lines are truncated, the record length of the trace file is 170 characters:

```
<TRACE ID=R0044711>
<OFTP  CLIENT>
<REQN>00044711</REQN><LNAM>GFIPSR4O</LNAM><TYPP>OTHER</TYPP><PNAM>GFIPSR4O</PNAM
<TCPA>010.020.129.003</TCPA><TCPH>MVSB.CSG.STERCOMM.COM</TCPH>
<IN_0001><TIME>08:28:05:12</TIME><CMD>
00000000 494F4445 54544520 46545020 52454144   59200D
</CMD>
</IN_0001>
<OU_0002><TIME>08:28:05:12</TIME><CMD>
00000000 58314746 49505352 344F2020 20202020   20202020 20202020 20202050 535220
00000020 20202033 32373536 4259594E 30303520   20202020 00000000 00000000 0D
</CMD>
<SSID><LEV>1</LEV><CODE>GFIPSR4O</CODE><PSWD>PSR</PSWD><SDEB>32756</SDEB><SR>B</
SSID>
</OU_0002>
<IN_0003><TIME>08:28:05:12</TIME><CMD>
00000000 58314746 49505352 344F2020 20202020   20202020 20202020 20202050 535220
00000020 20202033 32373536 4259594E 30303520   20202020 00000000 00000000 0D
</CMD>
<SSID><LEV>1</LEV><CODE>GFIPSR4O</CODE><PSWD>PSR</PSWD><SDEB>32756</SDEB><SR>B</
</SSID>
</IN_0003>
<TDIR>T</TDIR><FAPI>GFIPSR4O                 GFIPSR4O                 FICTST
0080</FRLG><FSIZ>000000000057</FSIZ>
<OU_0004><TIME>08:28:05:29</TIME><CMD>
00000000 48464943 54535420 20202020 20202020   20202020 20202020 2020204E 202020
00000020 20202020 30353132 32313038 32383035   20202020 20202020 47464950 535234
00000040 20202020 20202020 20202020 20202020   20474649 50535234 4F202020 202020
00000060 20202020 20202020 20204630 30303830   30303030 30353730 30303030 303030
</CMD>
<SFID><DDN>FICTST</DDN><RSV1>N</RSV1><DATE>051221</DATE><TIME>082805</TIME><USER
0057</FSIZ><REST>000000000</REST></SFID>
</OU_0004>
<IN_0005><TIME>08:28:05:53</TIME><CMD>
00000000 32303030 30303030 3030
```

Continued

```
</CMD>
<SFPA><ACNT>000000000</ACNT></SFPA>
</IN_0005>
<TYPP>OTHER</TYPP><NMGS>32756</NMGS><SYNC>00005</SYNC><COMP>COMPRESS</COMP><TYPD
<DATA_OU_0006><TIME>08:28:05:56</TIME><MSG>
00000000 44165BF9 F461F0F2 61F0F340 F1F87AF4   F17AF3F8 40D7E2D9 43F001F4 42400F
00000020 F3C2C140 E3D6D4F4 40D5D9C1 C3C54240   176140F9 F461F0F2 61F0F340 F1F87A

00000200 40C4400D E3D6D45C C4D7C7C6 C9D7E2D9   F27F40C4 400EE3D6 D45CC8D9 F1F2F3
00000280 C1D3400D E3D6D4F2 C4D7C7C6 C9D7E2D9   F27F40C4 40055CC7 E2D6F148 E77F40
000002A0 40
</MSG></DATA_OU_0006>
<FBYT>000000001760</FBYT><FNRB>000000000022</FNRB>
<OU_0007><TIME>08:28:05:57</TIME><CMD>
00000000 54303030 30303030 32323030 30303030   30303137 3630
</CMD>
<EFID><RCNT>000000022</RCNT><UCNT>000000001760</UCNT></EFID>
</OU_0007>
<IN_0008><TIME>08:28:06:02</TIME><CMD>
00000000 344E
</CMD>
<EFPA><CD>N</CD></EFPA>
</IN_0008>
<OU_0009><TIME>08:28:06:14</TIME><CMD>
00000000 52
</CMD>
<CD></CD>
</OU_0009>
<IN_0010><TIME>08:28:06:28</TIME><CMD>
00000000 463030
</CMD>
<ESID><CODE>00</CODE></ESID>
</IN_0010>
</OFTP  CLIENT>
</TRACE>
```

Connect:Express Fields

The trace files show the values that Connect:Express uses during execution of the file transfer, after protocol negotiations and Connect:Express decisions, for example: identifications, network addresses, synchronization parameters, network message size, physical file name, and user fields.

These values are shown according to the list of delimiters below:

Keyword	Description
APPM	LU6.2 Mode name
APPT	U6.2 Transaction program
COMP	Type of compression (after negotiation)
CRCO	CRC option (Y / N)
FAPI	API field - Etebac3 card - Odette file identification
FBYT	Real file size in kbytes
FLAB	File Label

Keyword	Description
FNAM	File symbolique name
FNRD	Real file size in number of records
FRLG	Record length of the local file
FSIZ	File size provided by the sender before transfer, for allocation
LNAM	Local name or alias of the monitor
LPHN	Local physical name of the file
NMGS	Network message size (after negotiation)
PNAM	Partner symbolic name
PROT	Transfer protocol, PESITD, PESITE, ODETTE, ETEBAC3, FTP
REQN	Request number
SNAL	SNA Address
SYNC	Synchronizaion parameters
TCPA	TCP/IP Address
TCPH	TCP/IP Host name
TDIR	Transfer direction (R/T)
TDST	Transfer destination
TORG	Transfer origin
TYPC	Local compression type (Presentation table number)
TYPD	Type of Data: ASCII, EBCDIC, BINARY
TYPL	Type of Link: L62, X25, TCP, SNA
TYPP	Type of Partner, TOM, OTHER APPLI
USDR	Pi99 PeSIT received
USDS	Pi99 PeSIT sent
X25A	X25 remote ddress
X25F	X25 facilities
X25L	X25 local address
X25P	MCH Identification
X25U	X25 user data field

Index

A

C

D

E

F

H

I

J

L

M

N

O

P

R

S

T

U

V

W

X