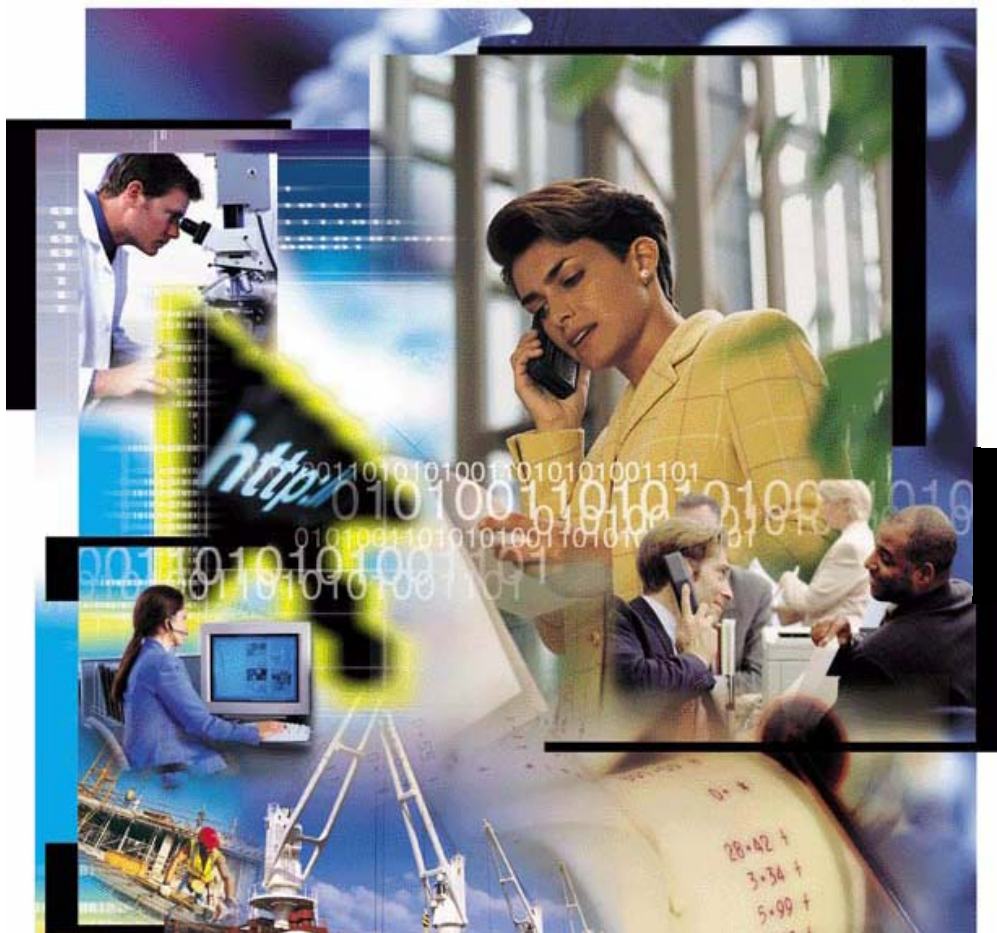


GXS EDI Services



# Expedite Base/400 Programming Guide

*Version 4 Release 6*



**Fourth Edition (November 2005)**

This edition replaces document number GC34-2254-02.

**© Copyright GXS, Inc. 1998, 2005. All rights reserved.**

Government Users Restricted Rights - Use, duplication, or disclosure restricted.



# Contents

---

- To the reader ..... ix
  - Who should read this book ..... ix
  - How to use this book ..... ix
  - Related books ..... x
  
- Chapter 1. Introducing Expedite Base/400 ..... 1
  - Information Exchange ..... 1
  - Understanding the Expedite Base/400 operating environment ..... 2
  - Hardware requirements ..... 2
  - Software requirements ..... 2
  - Connecting to Information Exchange ..... 3
    - Starting an Information Exchange session ..... 3
    - Ending an Information Exchange session ..... 3
    - Checking the status of the previous session ..... 3
  - Giving commands to Expedite Base/400 ..... 4
    - Understanding command syntax ..... 4
  - Getting responses from Expedite Base/400 ..... 4
    - Identifying Expedite Base/400 error messages ..... 5
    - Identifying Information Exchange error messages ..... 5
    - Receiving acknowledgments ..... 5
  - Sending and receiving data ..... 5
    - Free-format messages ..... 6
    - Limits on the number of files sent and received ..... 6
  - Restart and recovery considerations ..... 8
  - Providing security ..... 8
  - Selecting the Extended Security Option (ESO) ..... 8
  - Working with libraries ..... 9
  
- Chapter 2. Setting up files ..... 11
  - How Expedite Base/400 uses its primary files ..... 11
  - Description of Expedite Base/400 files ..... 12

Required files	12
Profile command file (INPRO)	12
Profile response file (OUTPRO)	14
Message command file (INMSG)	14
Message response file (OUTMSG)	15
Received files	16
Data recovery files	16
Optional files	17
Error message file (ERRORMSG)	17
Extended error text file (ERRORTXT)	17
Status display file (MESSAGE)	17
Base trace file (BASETRC)	17
Link trace file (LINKTRC)	18
Trace message text (TRACEMSG)	18
Certificate validation file (CERTFIL)	19
Electronic data interchange (EDI) control files	19
Calling IEBASE parameters	20
Displaying responses on the terminal	21
Chapter 3. Communicating with other operating systems	23
How Expedite Base/400 uses the CDH	23
Communicating with ASCII operating systems	24
Communicating with host operating systems	25
Communicating with older Information Exchange interfaces	27
Chapter 4. Sending and receiving EDI data	29
Overview of the SENDEDI command	29
Using EDI envelopes	30
Data between EDI segments	31
Sending EDI data	31
Using EDI qualifier tables	33
Using EDI destination tables	35
Using Information Exchange alias tables	36
Bypassing tables	37
Intersystem addressing for EDIFACT and UN/TDI	38
Using EDI distribution lists	39
Specifying Information Exchange control fields	40
Message name (MSGNAME) assignment	40
Message sequence number (MSGSEQNO) assignment	40
Message class (CLASS) assignment	41
SENDEDI response records	41
Receiving EDI data	42
EDI qualifier table entry format	43
EDI destination table entry format	44
Chapter 5. Using session-level recovery	47
Recovery considerations	47
Processing the message response file	48
Examining response records	48
Checking return codes	49
Checking the SENT and RECEIVED records	50

---

OUTMSG examples .....	51
<b>Chapter 6. Using checkpoint-level, file-level, and user-initiated recovery</b> .....	<b>55</b>
Selecting the appropriate recovery level .....	56
Recovery considerations .....	56
Setting up work files .....	57
Creating the session file (SESSION) .....	57
Creating the receive work file (RCVWORK) .....	57
Creating the EDI work file (EDIWORK) .....	57
Creating the output work file (OUTWORK) .....	58
Processing the message response file .....	58
Examining response records .....	58
Checking return codes .....	59
Recovery response file examples .....	60
Response file with a SESSIONEND (not 00000 or 28010) .....	60
Restarting after an error .....	62
Changing files on restart .....	63
Resetting the Expedite Base/400 session .....	64
<b>Chapter 7. Additional features of Expedite Base/400</b> .....	<b>65</b>
Using audit trails .....	65
Retrieving audit trails .....	66
Creating distribution lists .....	66
Free-format messages .....	67
Validating addresses, payment levels, and authorizations .....	67
Using acknowledgments .....	68
Working with libraries .....	70
Using data compression .....	70
<b>Chapter 8. Expedite Base/400 commands</b> .....	<b>71</b>
Command syntax .....	71
Command syntax examples .....	72
Profile commands .....	73
IDENTIFY command .....	74
SNACOMM command .....	77
SSL command .....	78
TCPCOMM command .....	79
TRACE command .....	80
TRANSMIT command .....	82
Message commands .....	84
ARCHIVEMOVE command .....	86
AUDIT command .....	87
CANCEL command .....	90
COMMIT command .....	93
DEFINEALIAS command .....	94
END command .....	97
GETMEMBER command .....	98
LIST command .....	101
LISTLIBRARIES command .....	104
LISTMEMBERS command .....	105
LISTVERIFY command .....	106

MSGINFO command	109
PURGE command	110
PUTMEMBER command	111
QUERY command	115
RECEIVE command	116
RECEIVEEDI command	122
RECEIVESTREAM command	126
SEND command	129
SENDEDI command	135
SENDSTREAM command	140
SESSIONINFO command	144
START command	145
TESTMSG command	147
Chapter 9. Expedite Base/400 response records	149
OUTPRO response records	149
PROFILERC record	150
RETURN record	151
WARNING record	152
OUTMSG response records	153
AUTOEND record	155
AUTOSTART record	156
AVAILABLE record	157
LIBRARYLIST record	162
MEMBERLIST record	164
MEMBERPUT record	166
MOVED record	167
MSGRESP record	168
NOTSENT record	170
RECEIVED record	172
RETURN record	178
SENT record	179
SESSIONEND record	181
SESSIONRESP record	182
STARTED record	185
WARNING record	186
Chapter 10. Expedite Base/400 installation overview	187
Downloading the Expedite Base/400 program	187
Installing the Expedite Base/400 program	188
Setting up SNA LU 6.2 communications	189
Ordering an LU name	190
Setting up an NCP leased connection	190
Setting up an NCP dial connection	196
Setting up TCP/IP communications	202
Setting up a TCP/IP dial connection	202
Obtaining and managing certificates for use with SSL	206
Managing Certificates in DCM	206
Expedite Base/400 objects	210

Appendix A. Product examples	213
SEND and RECEIVE command example	213
SENDEDI and RECEIVEEDI command example	214
SENDSTREAM and RECEIVESTREAM command example	215
INPRO example using SSL	215
START and SEND commands using SSL example	216
Appendix B. Common data header (CDH)	217
Construction of the CDH	217
User data	218
Libraries	218
Entry fields	218
Appendix C. Sample audit report	219
Using the AUDIT command	219
Sample audit report layout	221
Chapter 11. Expedite Base/400 messages and codes	223
Expedite Base/400 completion codes	224
Expedite Base/400 reason codes	225
Expedite Base/400 return codes	226
Successful completion code	226
Message command file syntax errors	227
Profile command file syntax errors	259
Parser errors	265
Destination verification errors	267
EDI errors	269
File validation errors	278
General environment errors	279
Session start and end errors	284
Session errors	287
APPC communications support errors	290
GSKIT errors	292
Unexpected errors	296
Appendix D. Using data compression	297
Understanding the TDAccess files used with Expedite Base/400	297
Preparing work files for data compression	298
Compressing files with COMPRESS(Y)	299
Compressing files with COMPRESS(T)	299
Decompressing received files	300
Decompressing RECEIVESTREAM data	301
Expedite Base/400 considerations when using COMPRESS(Y) or COMPRESS(T)	301
Restart and recovery considerations with TDAccess	302
Error messages and return codes for data compression	303
Glossary	309
Index	315







# To the reader

---

This book gives you the information necessary to program and use Expedite Base/400 for your company's applications. Expedite Base/400 is an Information Exchange base communication program that enables you to transmit data files and messages to and from Information Exchange. Users of Expedite Base/400 can use the features of Information Exchange through a worldwide network.

## Who should read this book

This book is written for experienced AS/400 programmers familiar with Information Exchange who want to write application programs for Expedite Base/400.

## How to use this book

Some type conventions are used in this book. Understanding what they mean can help you learn the material covered.

All application program interface (API) commands, parameters, and record names are displayed in the text in small, uppercase letters.

SESSIONEND

Default values in the text of this book appear in boldface type.

The default value is **est**.

There are many syntax examples in the book. These examples appear in a different typeface from the rest of the text. Required parameters are shown in boldface type in the syntax examples. Parameter variables are printed in italics, and default values are underlined.

**fileid**(*fileid*) format(n|y)

Because you do not have to worry about case when typing API commands, all examples are illustrated for you in lowercase letters. You can use uppercase or lowercase letters as you wish when typing API commands.

All terms included in the glossary are shown in italics at their first occurrence in the text of this book.

To locate information in this book, use the table of contents and the index.

## Related books

The following books contain information related to the topics covered in this book.

- *iSeries Access for Windows™ - Setup V5R2*, SC41-5507 (from the iSeries Information Center)
- *AS/400 Client Access for Windows 95/NT Version 4 Release 4*, SC41-3512
- *AS/400 Communications: Advanced Peer-to-Peer Networking User's Guide*, SC21-8188
- *AS/400 Communications: Advanced Program-to-Program Communications Programmer's Guide*, SC21-8189
- *AS/400 Communications: Intrasystem Communications Programmer's Guide*, SC21-9864
- *Information Exchange Interface Programming Guide*, GC34-2222 \*
- *Information Exchange Charges Reference*, GX66-0653 \*
- *Information Exchange Administration Services User's Guide*, GC34-2221 \*
- *Information Exchange Administration Services Supplement*, GN10-4073 \*
- *Information Exchange Administration Services Messages and Formats*, GC34-2324 \*
- *OS/400 TCP/IP Configuration and Reference Version 4 Release 3*, SC41-5420

I

- \* For your convenience, these documents can be viewed on the GXS Documentation Web site at: [//www.gxsolc.com/edi\\_bes.html](http://www.gxsolc.com/edi_bes.html).

## Introducing Expedite Base/400

---

*Information Exchange* is an electronic *mailbox* system that you can use to send information to and receive information from trading partners. Expedite Base/400 provides the interface that makes it possible to use Information Exchange from an AS/400 environment.

Expedite Base/400 uses Information Exchange to deliver and receive data, such as *electronic data interchange (EDI)* data. To communicate with Expedite Base/400 and transfer data files in and out of Information Exchange, you use *application program interface (API)* commands. You place these commands in control files.

Expedite Base/400 supports several Information Exchange data recovery methods. You can direct Expedite Base/400 to recover data at the level of a session, a checkpoint, or a file, or you can manage the data recovery yourself.

### Information Exchange

Information Exchange provides a means of sending, storing, and retrieving information electronically and makes it possible for users on dissimilar computer systems to communicate with each other. By establishing a computer-to-computer communication network between different locations, Information Exchange can both speed and simplify the delivery of *messages*, *EDI envelopes*, and other data.

Information Exchange is an electronic mailbox system residing on a worldwide network. With Information Exchange as an alternative to terminal-to-computer communication, you send messages to the mailbox and retrieve the messages waiting for you in the mailbox.

Through the worldwide network, Information Exchange can link the geographically scattered locations of a single company or of different companies. For example, a manufacturing company can use Information Exchange to communicate with its suppliers or distributors.

A program running on your computer can communicate with Information Exchange on behalf of one or more Information Exchange *addresses*. This means that your computer can communicate with any Information Exchange address, through Information Exchange, during one physical connection between systems.

## Understanding the Expedite Base/400 operating environment

Expedite Base/400 runs on an AS/400 computer, which uses *extended binary-coded decimal interchange code (EBCDIC)*. Expedite Base/400 can communicate with some *American National Standard Code for Information Interchange (ASCII)* systems, such as a personal computer (PC), as well as with other EBCDIC computers.

## Hardware requirements

The minimum hardware configuration for Expedite Base/400 Version 4 Release 6 is:

- An AS/400 or IBM eServer iSeries system configured to run OS/400 Version 5 Release 1 or Version 5 Release 2.
- 14 Mb of memory for objects
- Sufficient additional memory for personal command, response, control, and data files. The amount of space required depends on your application.

Additionally, the following is required:

- For SNA LU 6.2 communications, an NCP leased or dial LU 6.2 connection.
- For TCP/IP communications, a leased or dial TCP/IP connection to the worldwide network.

## Software requirements

The minimum software requirement is:

- OS/400 Version 5 Release 1 or Release 2.
- For TCP/IP dial access, AS/400 Client Access with Operation Navigator.
- If you will be using SSL communications, Digital Certificate Manager (DCM) is required. It is a free iSeries feature that allows you to centrally manage digital certificates for your applications. To use DCM successfully, ensure that you do the following:
  - Install the cryptographic access provider licensed program (5722-AC3). This product determines the maximum key length that is permitted for cryptographic algorithms based on export and import regulations. You must install this product before you can create certificates.
  - Install the browser-based DCM feature of OS/400 (Option 34).
  - Install the IBM HTTP Server for iSeries (5722-DG1) and start the \*ADMIN server instance.
  - Make sure that you have the IBM Developer Kit for Java™ (5722-JV1) installed.
  - Make sure that TCP is configured for your system so that you can use a web browser and the HTTP Server \*ADMIN instance to access the DCM feature.

## Connecting to Information Exchange

To connect to Information Exchange, you must have an Information Exchange account, user ID, and password. These items must be defined in your Expedite Base/400 profile. Use the IDENTIFY command to provide Expedite Base/400 with these values. For more information on the IDENTIFY command, see “IDENTIFY command” on page 74.

### Starting an Information Exchange session

Before you can transfer data, you must start an Information Exchange session. Usually, Expedite Base/400 starts this session automatically with information provided in the IDENTIFY command. However, you can start a session manually by using the START command.

Before using the START command, specify AUTOSTART(n) in the TRANSMIT command to disable the automatic session start function. When the session starts, the connection to Information Exchange is complete. At this point, Expedite Base/400 performs data transfer requests.



**NOTE:** If you start an Information Exchange session while another Information Exchange session with the same account and user ID is running, the results are unpredictable. You should not use the same Information Exchange account and user ID for more than one Expedite interface.

For more information on the START and TRANSMIT commands, see Chapter 8, “Expedite Base/400 commands.”

### Ending an Information Exchange session

When all file transfer requests are complete, Expedite Base/400 usually ends the session automatically. However, if you specified AUTOEND(n) in the TRANSMIT command to disable the automatic session end function before starting the session, you must end the session manually with the END command.

For more information on the END command, see “END command” on page 97.

### Checking the status of the previous session

You can use Expedite to determine if the previous session was successful or not. After a session fails with a response code of 29999, you can obtain information about the completion status of the previous session by specifying AUTOSTART(n), AUTOEND(n), AND RECOVERY(s) on the TRANSMIT command in your Expedite profile. Next, create an output file containing only START and END records as follows:

```
START CHECK(y);  
END;
```

Run Expedite. Examine the output file to check the LASTSESS parameter value on the STARTED record. The status will be 0 if the last session was successful. The status will be 1 if the last session was not successful. No data is transferred in the above example, and you are not charged for this inquiry.

## Giving commands to Expedite Base/400

You use command files to provide Expedite Base/400 with input. The API commands you enter in the *profile command* and *message command* files give Expedite Base/400 the information it needs to operate. The profile command file that tells Expedite Base/400 how to establish a communication environment is named INPRO. The message command file that Expedite Base/400 uses to send and receive files and messages, including EDI data, is named INMSG.

Expedite Base/400 places response records in the profile response and message response files (OUTPRO and OUTMSG) to tell you whether your API commands have completed.

For detailed information on commands, see Chapter 8, “Expedite Base/400 commands.”

### Understanding command syntax

The *syntax*, or structure, of the commands is shown below:

```
#Comment or description
command parameter(value) parameter(value) ... parameter(value);
```

You can use uppercase or lowercase letters for API commands. The syntax elements are described below:

Syntax element	Description
#	Defines or delimits a comment line. You can type any information you like after a #, and Expedite Base/400 ignores the characters that follow it on the same line. If a # is included in a parameter value, Expedite Base/400 knows from the #'s context that it is part of a command and does not ignore the parameter value or the remaining characters on the line.
command	Identifies an Expedite Base/400 command.
parameter	Identifies a parameter for the associated command.
value	Defines a value associated with the parameter.
...	In the example above, the ellipsis (...) indicates that you can specify as many parameters as necessary. (It is not part of the syntax.)
;	Ends the command.

The Expedite Base/400 response records use a similar syntax.

## Getting responses from Expedite Base/400

Expedite Base/400 places response records in two *response files*. It generates the *profile response file* (OUTPRO) when it processes an INPRO file and generates the *message response file* (OUTMSG) when it processes an INMSG file.

OUTPRO repeats INPRO commands and associates the appropriate return codes to each command. You can examine the RETURN record in OUTPRO to see if the profile commands ran correctly.

OUTMSG works similarly to OUTPRO. Expedite Base/400 places response records in OUTMSG when it processes the INMSG file. OUTMSG contains the completed commands and their associated return codes. Check the RETURN record in OUTMSG to be sure that your files and messages have been sent correctly.

For detailed information on response records, see Chapter 9, “Expedite Base/400 response records.”

## Identifying Expedite Base/400 error messages

Error messages generated by Expedite Base/400 are written to OUTPRO and OUTMSG. They reflect errors resulting from faulty commands or system communication problems. Each error message is identified by a code.

For more information on error messages, see Chapter 11, “Expedite Base/400 messages and codes.”

## Identifying Information Exchange error messages

Information Exchange may generate error messages. The most common reason for an error message is that Information Exchange was unable to deliver a file. Information Exchange puts the error messages into your mailbox with a sender's account name of \*SYSTEM\* and user ID of \*ERRMSG\*. You can identify these error messages by their codes. You must receive these messages with a RECEIVE or RECEIVESTREAM command. For more information on Information Exchange error messages, see *Information Exchange Administration Services Messages and Formats*.

## Receiving acknowledgments

Although they are not error messages, Information Exchange also identifies Information Exchange *acknowledgments* with a sender's account name of \*SYSTEM\* and a user ID of \*ERRMSG\*.

The three types of acknowledgments you can request on any of the Expedite Base/400 SEND commands are described below:

This acknowledgment type:	Generated by Information Exchange when:
receipt	A message reaches the receiver's mailbox after a successful Expedite Base/400 session.
delivery	A destination user receives a message from the Information Exchange mailbox.
purge	A message is purged from the receiver's mailbox.

## Sending and receiving data

You can use three sets of commands to send and receive messages, files, and EDI data using Expedite Base/400:

- SEND - RECEIVE
- SENDEDI - RECEIVEEDI
- SENDSTREAM - RECEIVESTREAM

With the SEND command, you can send each message or file individually. You can send and receive *binary files*, which contain machine instructions such as executable computer programs for a computer to read or text files for people to read.

The RECEIVE command retrieves either all files or specific files, including free-format messages, from the Information Exchange mailbox.

The SENDEDI command transmits multiple EDI envelopes with different addresses from a single file. Information in this file can consist of any combination of *X12*, *Uniform Communication Standard (UCS)*, *Electronic Data Interchange for Administration, Commerce, and Transport (EDIFACT)*, or *United Nations/Trade Data Interchange (UN/TDI)* data. SENDEDI resolves the *destinations* of the different pieces of information from the EDI data, so you do not have to retype existing destination information.

The RECEIVEEDI command can receive multiple EDI envelopes containing different types of data.

The SENDSTREAM command sends data from the message command file (INMSG) to Information Exchange. The RECEIVESTREAM command receives data directly into the message response file (OUTMSG). SENDSTREAM and RECEIVESTREAM are valid only when you use session-level recovery.

For more information on sending and receiving data, see Chapter 3, “Communicating with other operating systems,” and Chapter 4, “Sending and receiving EDI data.”

## Free-format messages

You can create *free-format messages* using the editor on your AS/400 system. Use the SEND and RECEIVE commands to transfer these messages. For more information, see “Free-format messages” on page 67.

## Limits on the number of files sent and received

Information Exchange limits the number of files sent and received between commits. The size of the files does not matter. The current limit of 1,000 is an Information Exchange value that can be set differently in different Information Exchange installations. Contact your marketing representative to determine the maximum for Information Exchange installations outside the U.S.

If you are using session-level recovery, you cannot send or receive more than 1,000 files in a session. If you are sending EDI data, each EDI envelope counts as a file, even if you have multiple EDI envelopes in a single file.

If you use user-initiated recovery, you must not specify more than 1,000 SEND, SENDEDI, or PUTMEMBER commands without specifying a COMMIT command. Do not send more than 1,000 EDI envelopes from a single file, because each envelope counts as a file.

If you use checkpoint-level recovery, you should not have any problems, unless you send more than 1,000 small files where the total number of characters sent is less than your COMMITDATA value on the TRANSMIT command in your profile. The default value is 141,000. If this is a problem, lower the COMMITDATA value. You should not have a problem with the receive file limit with checkpoint-level recovery.

If you try to send more files than the limit, you will get an error from Information Exchange, which Expedite reports to you as a 31360 return code on the SESSIONEND record. In this case, you can break your input file into multiple input files and run Expedite repeatedly until the files are sent. For session-level recovery, see Chapter 5, “Using session-level recovery.” For all other recovery levels, see Chapter 6, “Using checkpoint-level, file-level, and user-initiated recovery.”



If you have more than 1,000 files in your mailbox that match your receive requests and you are using session-level recovery, Information Exchange stops sending files when the 1,000 limit is reached. If you have more files in your mailbox, Expedite returns a 28171 value in the RETURN parameter after the last file is received. The SESSIONEND value is 28010, indicating the session completed successfully but not all commands were processed. The data already received is no longer in your mailbox. You do not need to recover a session that ends with 28171 and 28010.

If your session ends with 28010 and the last RECEIVE command ended with RETURN(28171), you should process the data received or make sure it is not overwritten if you run Expedite again. For more information on session-level recovery, see Chapter 5, “Using session-level recovery.”

If you switch to checkpoint-level, user-initiated, or file-level recovery, you can receive all the files in your mailbox in a single session without getting the 28171 return code.

## Restart and recovery considerations

Expedite Base/400 supports four types of data recovery: session-level, checkpoint-level, file-level, and user-initiated.

When you use *session-level recovery* and an error occurs, you must retransmit all data. Although retransmitting all data is time-consuming, it is the easiest way to recover from errors. For this reason, session-level recovery is the default *recovery level* of Expedite Base/400. For more information on session-level recovery, see Chapter 5, “Using session-level recovery.”

*Checkpoint-level recovery*, *file-level recovery*, and *user-initiated recovery* require more effort on your part (for example, setting up special work files) but can result in more efficient data recovery. For more information on these recovery levels, see Chapter 6, “Using checkpoint-level, file-level, and user-initiated recovery.”

## Providing security

The network provides security at the network access level, the application selection level, and the data access level. You should understand that network security features operate within a widely-used data processing environment. Information Exchange can protect users only if they observe security controls. Passwords must be changed at intervals to ensure mailbox security. Authorization levels enable users to perform certain functions and restrict those users from performing other functions.

Secure socket layer (SSL) access is available for customers that want to use certificates to identify authorized users. To enable SSL, you must use the SSL command and change parameters on the IDENTIFY and START commands.

## Selecting the Extended Security Option (ESO)

ESO provides additional password and Information Exchange mailbox security. The ESO USER flag may be turned on by the *Information Exchange Service Administrator*, using Information Exchange Administration Services. Users with ESO can send files and messages to other users with ESO and users without ESO.

For information on using ESO, see *Using Information Exchange Administration Services*.

ESO contains the following security features:

- Your Information Exchange Service Administrator must change your ESO password if it is the same as your user ID. If you do not provide a new password in the START command, your Information Exchange session will not be successful.
- To conform with the ESO password rules, your new password:
  - Cannot be the same as your Information Exchange user ID
  - Must be at least 6 characters in length
  - Must contain at least 3 different characters
  - Must begin and end with a nonnumeric character
  - Must contain at least 1 nonalphanumeric character
  - Must contain at least 1 alphabetic character
  - Can only be the valid characters A-Z, 0-9, and special characters @, #, and \$
  - Cannot be the same as the current or five previous passwords
  - Cannot contain more than 2 identical, consecutive characters

- Cannot contain more than 3 identical, consecutive characters from the previous password

If your new password does not conform to these rules, it is considered invalid, and your Information Exchange session will not be successful.

- Information Exchange revokes your ESO user ID if you make three consecutive attempts to start an Information Exchange session with an invalid password. All further attempts to start a session will be unsuccessful until your Information Exchange Service Administrator resets your password.
- The Information Exchange Service Administrator can use the Information Exchange Administration Services password reset function to reinstate an ESO user ID.



**NOTE:** Passwords are reset immediately after resetting the ESO option to Y. When the administrator sets the ESO option to Y, all affected users must change their password at their next logon. This is true even if the administrator subsequently resets the ESO option back to N.

## Working with libraries

A *library* is a place to store information for an extended period of time. Unlike messages in a user's mailbox, information in a library is not deleted automatically after a certain amount of time. Libraries are made up of library *members*, which contain the information you want to store. To use libraries in Expedite Base/400:

1. Create the library using Information Exchange Administration Services.
2. Use the PUTMEMBER command to put members into the library and the GETMEMBER command to retrieve members from the library and place them into your mailbox.
3. Use the LISTLIBRARIES and LISTMEMBERS commands to identify libraries and members to which you have access.

For more information on libraries, see “Working with libraries” on page 70 and “Creating and managing libraries” in *Using Information Exchange Administration Services*.



## Setting up files

---

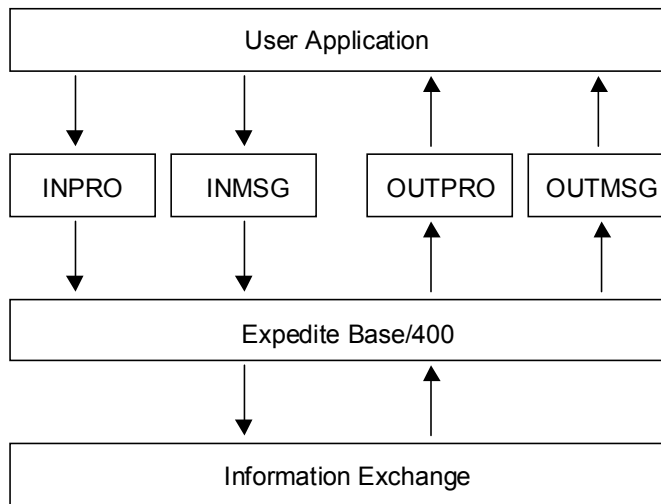
This chapter gives you the information you need to set up Expedite Base/400 before you begin to use it. It provides a general explanation of how Expedite Base/400 works, and describes the files you need to create and their *attributes*, or properties.

### How Expedite Base/400 uses its primary files

Expedite Base/400 is a file-oriented program. To use Expedite Base/400, you place requests in a command file, call the IEBASE program to process the file, and examine the appropriate response file to find completed requests. The primary command and response files used by Expedite Base/400 are listed here:

- Profile command file (INPRO)
- Profile response file (OUTPRO)
- Message command file (INMSG)
- Message response file (OUTMSG)

The following figure illustrates how these files make it possible for you to communicate with Information Exchange.



Before you can use Expedite Base/400, you must create these primary files and any other work files and optional files needed for your applications.

## Description of Expedite Base/400 files

You must create all files used by Expedite Base/400 before you can use the program. Expedite Base/400 does not create any files for you. The four primary command and response files, INPRO, OUTPRO, INMSG, and OUTMSG, are required. Depending on whether you decide to use checkpoint- or session-level recovery, you may need to create other files that can trace requests and will determine the amount of detail provided in error descriptions.

## Required files

The files described here must be created before using Expedite Base/400. You may need to create other files as well, depending on your applications.

### Profile command file (INPRO)

The name of the profile command file is INPRO. You must create an INPRO file before you can use Expedite Base/400. Enter setup information in INPRO using profile commands. Expedite Base/400 uses the information contained in INPRO to connect to Information Exchange and set up the communication environment. You can find more details on profile commands in “Profile commands” on page 73.

Expedite Base/400 uses default values for many of the profile command parameters; however, it does not use default values for the required parameters. If you plan to use AUTOSTART(y), the default option of the TRANSMIT command, Expedite Base/400 requires several parameters from the IDENTIFY command. If you plan to use COMMTYPE(s), the default option of the TRANSMIT command, the SNACOMM profile command is also required. Expedite Base/400 uses default values for many of the profile command parameters; however, it does not use default values for the required parameters.

### Required IDENTIFY parameters

If Expedite Base/400 starts the Information Exchange session automatically, which is the default, you must use the IDENTIFY command to provide the following parameters so Expedite Base/400 can identify you as a user. If you use AUTOSTART(n) in the TRANSMIT command, these parameters are optional.

IEACCOUNT	Information Exchange account
IEUSERID	Information Exchange user ID
IEPASSWORD	Information Exchange password

For more information, see “IDENTIFY command” on page 74.

### Required SNACOMM parameter

If you use SNA communications, which is the default on the TRANSMIT command, you must use the USERLUNAME parameter of the SNACOMM command to indicate the names of the Expedite Base/400 logical units (LUs) you are using. This value is known as the local location name on your APPC device description. Your system programmer can provide you with this LU name.

For more information, see “SNACOMM command” on page 77.

### Required TCPCOMM parameters

If you use TCP/IP communications by specifying COMMTYPE(t) on the TRANSMIT command, you must use the following parameters to indicate the address of the Information Exchange system you will be using. The sample file XTCPIPADD lists the valid values by geographic location.

IETCPHOSTn	Information Exchange IP address
IETCPPORTn	Information Exchange port

For more information, see “SSL command” on page 78.

### INPRO attributes

Create INPRO with the following attributes:

This attribute:	Has these characteristics:
Space	INPRO has no minimum space requirement. It must be large enough to hold all of your profile commands.
Record length	INPRO's record length must be at least 72 bytes long, but not more than 255 bytes.

To create the file, you can issue the following sample AS/400 command:

```
crtpf inpro rcdlen(255)
```

### Sample INPRO

A sample INPRO file is shown here. This file contains the minimum IDENTIFY and SNACOMM command requirements when you use AUTOSTART(y).

```
IDENTIFY IEACCOUNT(ACCT) IEUSERID(USER01) IEPASSWORD(XXXXXX);
SNACOMM USERLUNAME(LUNAME);
```

## Profile response file (OUTPRO)

The name of the profile response file is OUTPRO. You must create an OUTPRO file before you can use Expedite Base/400. When Expedite Base/400 reads the profile command file, it echoes the profile commands to OUTPRO, along with their associated return codes. The return record in OUTPRO contains the return codes for each command. The PROFILEREC record contains the return code for the processing of the entire INPRO file.

For more information on response records, see Chapter 9, “Expedite Base/400 response records.”

### OUTPRO attributes

Create OUTPRO with the following attributes:

This attribute:	Has these characteristics:
Space	OUTPRO must be large enough to hold the echoed INPRO commands and their associated responses.
Record length	OUTPRO's record length must be at least as long as INPRO's, at least 72 bytes long, but not more than 255 bytes.

To create the file, you can issue the following sample AS/400 command:

```
crtpf outpro rcdlen(255)
```

### Sample OUTPRO

The OUTPRO file produced by the command file shown in “Sample INPRO” on page 13 follows:

```
IDENTIFY IEACCOUNT(ACCT) IEUSERID(USER01) IEPASSWORD(XXXXXX);
RETURN(00000);
SNACOMM USERLUNAME(LUNAME);
RETURN(00000);
PROFILERC(00000);
```

## Message command file (INMSG)

The name of the message command file is INMSG. You must create an INMSG file before you can use Expedite Base/400. You can send and receive files by entering file transfer requests with message commands into INMSG.

For more information on message commands, see “Message commands” on page 84.

### INMSG attributes

Create INMSG with the following attributes:

This attribute:	Has these characteristics:
Space	INMSG has no minimum space requirement. It must be large enough to hold all of your message commands.
Record length	INMSG's record length must not exceed 255 bytes.



To create the file, you can issue the following sample AS/400 command:

```
crtpf inmsg rcdlen(255)
```

### Sample INMSG

A sample INMSG file used to send and receive electronic data interchange (EDI) files is shown here. This example uses an Information Exchange *user class* of EDITEST. A user class identifies documents to your *trading partners*, the business associates with whom you exchange electronic information. After sending an EDI file called EDIDATA, you will receive all the files with a user class of EDITEST from the Information Exchange mailbox.

```
sendedi fileid(library/edidata) class(editest);
receiveedi fileid(library/ediout(rcv)) class(editest);
```

## Message response file (OUTMSG)

The name of the message response file is OUTMSG. You must create an OUTMSG file before you can use Expedite Base/400. When Expedite Base/400 processes the message command file, it echoes the completed commands to OUTMSG along with their associated return codes. The RETURN record in OUTMSG contains the return codes for each completed command.

If a RETURN record with a zero return code follows a command, the command completed successfully. If a RETURN record with a nonzero return code follows a command, the command did not complete successfully. The SESSIONEND record contains the final return code for INMSG and INPRO files that have been processed.

For more information on processing OUTMSG, see Chapter 5, “Using session-level recovery,” and Chapter 6, “Using checkpoint-level, file-level, and user-initiated recovery.”

### OUTMSG attributes

Create OUTMSG with the following attributes:

This attribute:	Has these characteristics:
Space	OUTMSG must be large enough to hold the echoed INMSG commands and their associated responses.
Record length	OUTMSG's record length must be at least as long as INMSG's, at least 72 bytes long, and less than or equal to 255 bytes.

To create the file, you can issue the following sample AS/400 command:

```
crtpf outmsg rcdlen(255)
```

### Sample OUTMSG

The OUTMSG file produced after the successful completion of the INMSG file shown in “Sample INMSG” is:

```
AUTOSTART SESSIONKEY(3JKKDIR8);
STARTED LASTSESS(0) RESPCODE(00000) SESSIONKEY (3JKKDIR8)
IEVERSION(04) IERELEASE(07);
RETURN(00000);
sendedi fileid(library1/edidata) class(editest);
```

```

SENT UNIQUEID(57586838) LENGTH(2260) ACCOUNT(ACT1) USERID(DEPT01)
EDITYPE(X12)
DESTINATION(DUNS01) QUALIFIER(01) CONTROLNUM(000022223)
CLASS(EDITEST)
MSGNAME(00022223) MSGSEQNO(00001);
RETURN(00000);
receiveedi fileid(library/edidata(rcv)) class(editest);
RECEIVED ACCOUNT(ACT1) USERID(DEPT01) CLASS(EDITEST) CHARGE(5)
RECEIVER(ACT1 DEPT01) RECVQUAL(01)
SENDER(ACT2 DEPT02) SENDQUAL(01) CONTROLNUM(000022228)
FILEID(EDIOUT) MSGDATE(040325) MSGDATELONG(20040325) MSGTIME(192507)
MSGSEQO(000009) MSGNAME(00022223) MSGSEQNO(00001
SESSIONKEY(3JKKDIR8) DELIMITED(E) TIMEZONE(L)
SYSNAME(EXA4000T) SYSLEVEL(460) DATATYPE(E) EDITYPE(X12)
SENDERFILE(EDIDATA) SENDERLOC(3380K /TSO203)
FILEDATE(040325) FILEDATELONG(20040325)
RECFM(F) RECLLEN(80) RECDLM(E) UNIQUEID(57586838)
SYSTYPE(71) SYSVER(4);
RETURN(00000);
AUTOEND;
RETURN(00000);
SESSIONEND(00000);

```

## Received files

You must create files to hold the data that Expedite Base/400 receives from Information Exchange. Be sure to make these files large enough to hold all of the data that they must receive. Also, make sure that the record length of the file is appropriate for the received data.

## Data recovery files

You must create the following files if you plan to use checkpoint-level, file-level, or user-initiated recovery:

- Output work file (OUTWORK)
- Receive work file (RCVWORK)
- Session file (SESSION)
- EDI work file (EDIWORK). This file is only needed when using the SENDEDI command with the VERIFY parameter equal to *c* or *g*.

See Chapter 6, “Using checkpoint-level, file-level, and user-initiated recovery,” for more information on data recovery files.

## Optional files

You might want to use some of the optional Expedite Base/400 files which are described here.

### Error message file (ERRORMSG)

Expedite Base/400 can provide error descriptions in the RETURN, WARNING, SESSIONEND, and PROFILERC response records. These descriptions are contained in the ERRORMSG file.

When errors occur and the ERRORMSG file is included in the library list, Expedite Base/400 writes error descriptions to response records. If you do not include this file, error message codes display without error message text. We suggest you make use of the ERRORMSG file.

**Attention:** Do not modify this file. Modifications can cause you to receive incorrect error messages.

### Extended error text file (ERRORTXT)

Expedite Base/400 can provide extended error descriptions in the RETURN, WARNING, and SESSIONEND response records. These extended error descriptions are contained in the ERRORTXT file.

When errors occur and the ERRORTXT file is included in the library list, Expedite Base/400 includes error messages in the response records. If you do not include this file, error messages display without extended error text. We suggest you make use of the ERRORTXT file.

**Attention:** Do not modify this file. Modifications can cause you to receive incorrect error messages.

### Status display file (MESSAGE)

Expedite Base/400 can provide status messages. The status messages Expedite Base/400 uses are contained in the status display file (MESSAGE).

**Attention:** Do not modify this file. Modifications can cause you to receive incorrect error messages.

### Base trace file (BASETRC)

You can request several types of traces, separately or in combination. You need to request a trace only when you are working with the Help Desk on a problem. All traces, except LINK traces, go to the base trace file, BASETRC. When you request a trace using the TRACE profile command, include BASETRC in the library list when you run Expedite Base/400.

For more information on how to select traces, see “TRACE command” on page 80.

### BASETRC attributes

Create BASETRC with the following attributes:

This attribute:	Has these characteristics:
Space	The space BASETRC requires varies, depending on the trace options you request and the amount of data you transmit. If you request a PROTOCOL trace, you must make BASETRC large enough to hold all of the data you plan to send or receive during an Expedite Base/400 run, in addition to other trace information that Expedite Base/400 generates.
Record length	BASETRC record length must be at least 72 bytes. A record length of 255 bytes is usually effective.

To create the file, you can issue the following sample AS/400 command:

```
crtpf basetrc rcdlen(255)
```

### Link trace file (LINKTRC)

This file contains information produced by the link trace. You need to request a trace only when you are working with the Help Desk on a problem. When you request a link trace using the TRACE profile command, include LINKTRC in the library list when you run Expedite Base/400.

### LINKTRC attributes

Create LINKTRC with the following attributes:

This attribute:	Has these characteristics:
Space	The space required by LINKTRC varies, depending on the amount of data you transmit. LINKTRC must be large enough to hold all the data you plan to send or receive during an Expedite Base/400 run in addition to other trace information that Expedite Base/400 generates.
Record length	LINKTRC record length must have the same record length as BASETRC (between 121 and 255 bytes). If the record length is too short, Expedite Base/400 does not write a link trace. A record length of 255 bytes is usually effective.

To create the file, you can issue the following sample AS/400 command:

```
crtpf linktrc rcdlen(255)
```

### Trace message text (TRACEMSG)

Expedite Base/400 can provide trace message text in languages other than English. Expedite Base/400 uses this file to change trace file text for national language support.

If you include this file (TRACEMSG) when you run Expedite Base/400, it takes trace messages from this file. If you receive a version of Expedite Base/400 customized for a language other than English, your system programmer can provide you with this file name.



**NOTE:** This file is only included in versions of Expedite Base/400 that are not in the English language.

### Certificate validation file (CERTFIL)

This is a new optional file where customers can store untrusted server certificate serial numbers. This file is only used if you enable SSL by setting the enablessl parameter to Y on the SSL command.

After the negotiation phase, Expedite will compare the serial number from the Information Exchange server certificate with the entries in this file. If a match is found, the session ends, an error is produced, and an error message is written to OUTMSG. GSKIT returns a serial number field of 8 hex characters (11 characters including the separators ':').

#### CERTFIL attributes

If you find that you need to have a CERTFIL, you can either download it from the EDI Services Web site or create one by issuing the following sample AS/400 command:

```
crtpf certfil rcdlen(80)
```

### Electronic data interchange (EDI) control files

The SENDEDI command uses the following files when sending EDI data:

- EDI qualifier table (QUALTBL)
- EDI destination tables (TTABLExx)

See Chapter 4, “Sending and receiving EDI data,” for more information about these files.

To create the files, you can issue the following sample AS/400 commands:

```
crtpf qualtbl rcdlen(255)  
crtpf ttable01 rcdlen(255)
```

## Calling IEBASE parameters

The CALL statement instructs OS/400 to start the Expedite Base/400 program, which is named IEBASE. Your CALL statement may use the PARM parameter.

You can code the following keyword parameters:

```
reset inpro(inpro) outpro(outpro) inmsg(inmsg)
outmsg(outmsg) status check(1|2) version
```

The parameters you can enter are given in the following list:

Parameter:	<b>Description:</b>
RESET	Resets a checkpoint-level recovery session if you have an unrecoverable error and need to reset your session.
INPRO	Changes the default name for the profile command file, which normally has a name of INPRO.
OUTPRO	Changes the default name for the profile response file, which normally has a name of OUTPRO.
INMSG	Changes the default name for the message command file, which normally has a name of INMSG.
OUTMSG	Changes the default name for the message response file, which normally has a name of OUTMSG.
STATUS	Tells you whether Expedite Base/400 is in a restart situation. No data is sent or received when you issue this parameter. You receive a completion code of 1 if Expedite Base/400 would try to restart the previous session. You receive a completion code of 0 if Expedite Base/400 would not try to restart the previous session.
CHECK	Allows you to verify that INPRO and INMSG are valid without running a session with Information Exchange, where: <ol style="list-style-type: none"> <li>1 Indicates that the syntax of the commands in INPRO and INMSG is to be checked.</li> <li>2 Indicates that the syntax of the commands in INPRO and INMSG is to be checked. In addition, the existence of any files specified in the SEND, SENDEDI, and PUTMEMBER commands is to be checked. This is the default.</li> </ol>
VERSION	Results in Expedite Base/400 responding with the current version of the Expedite software you are using. If you specify other command line parameters with the VERSION parameter, the other parameters will be ignored. The format of the string that is returned is as follows: <p style="margin-left: 4em;">Version x.x, mm/dd/yy</p> Where <i>x</i> is a digit from 0-9, <i>mm</i> is the two-digit month, <i>dd</i> is the two-digit day, and <i>yy</i> is the two-digit year.

The following example shows a CALL statement used to change the names of INPRO, OUTPRO, INMSG, and OUTMSG.

```
call pgm(iebase) parm('inpro(inpro1)' 'outpro(outpro1)' 'inmsg(inmsg1)'  
                     'outmsg(outmsg1)')
```

## Displaying responses on the terminal

Expedite Base/400 displays status messages to the terminal during execution. When complete, the system prompts the user to “Press Enter to end terminal session.”

If your use of Expedite Base/400 requires non-interactive displays, then you can override this output to go to a database file or a printer file using a command such as:

```
ovrprtf file(stdout) tofile(qgpl/qprint)
```





## Communicating with other operating systems

---

When transferring files between systems, you must consider:

- The type of system that will receive the files, for example, AS/400, OS/2, or DOS
- The record format of the file being sent
- The type of data being sent (binary or text)

This chapter addresses each of the above issues and the actions you should take in each instance. It also discusses a valuable aid to file transfer, the *common data header (CDH)*.



**NOTE:** In this chapter, “system” refers to a particular type of operating system, rather than to a particular Information Exchange system.

### How Expedite Base/400 uses the CDH

Most Information Exchange interfaces can send and receive additional information describing the contents of a file. The CDH contains this information. Typically, it includes the items listed here:

- The type of file (text or binary)
- The record delimiters used in the file, for example, carriage-return/line-feed (CRLF) characters or record lengths
- Whether the data is EDI-formatted
- The original name of the file on the sending system
- A free-format text description of the file
- Other information describing the original file

When Expedite Base/400 receives a file, it can use the information in the CDH to format the file. Expedite Base/400 places the CDH information for received data in the output file on the RECEIVED record or the AVAILABLE record. (See page 172 for a description of the RECEIVED

record and page 157 for a description of the AVAILABLE record.) You do not have to do anything to send or receive the CDH because Expedite Base/400 does this automatically. However, you need to understand the CDH to fully understand the file transfer capabilities of Expedite Base/400. For more information on the CDH, see Appendix B, “Common data header (CDH).”

## Communicating with ASCII operating systems

Expedite Base/400 runs on AS/400, an EBCDIC system. Expedite Base for Windows and Expedite Base/AIX products run on ASCII systems. This section tells you how to communicate with these ASCII systems.

### Sending text files to an ASCII operating system

Text files usually contain data that you can read, as opposed to executable programs or computerized drawings that are meant for a computer to read. When you send text data to an ASCII system, you must tell Expedite Base/400 to put carriage-return and line-feed characters (X'0D0A') at the end of each record. Carriage-return and line-feed characters enable the file to maintain its original record format and be readable when it gets to the ASCII system. To do this, use the DELIMIT(C) option of the SEND command. For example,

```
send fileid(library/mytext) account(act1) userid(user04) class(pctext) delimit(c);
```

The Expedite product that receives the data translates that data from EBCDIC to ASCII when it receives the file. You do not have to do anything to request this translation.

### Receiving text files from an ASCII operating system

When Expedite Base/400 receives text files, it looks at the CDH to determine whether the data has carriage-return and line-feed characters at the end of each record. If the CDH indicates that these characters are present, Expedite Base/400 removes them and ends the records where the characters were found.

Because all interfaces do not send a CDH, you must tell Expedite Base/400 what to do with records received if there is no CDH. To do this, use the DELIMITED(C) parameter of the RECEIVE command. The DELIMITED(C) parameter tells Expedite Base/400 to assume that the data has carriage-return and line-feed characters when there is no CDH. This permits you to receive text data from older Expedite interfaces that do not send a CDH.

The following is an example of how to use the DELIMITED(C) parameter of the RECEIVE command.

```
receive fileid(library/textout(rcv)) account(act1) userid(user05)  
class(pctext) delimited(c);
```

The sending Expedite program on the ASCII system translates the data from ASCII to EBCDIC when it sends the file. You do not have to do anything to request this translation.

### Sending binary files to an ASCII operating system

Binary files usually contain data that you cannot read, such as executable programs and computerized drawings. When you send binary files to an ASCII system, you must tell the Expedite product receiving the data not to translate the data from EBCDIC to ASCII. To do this, use the DATATYPE(B) option of the SEND command. When you send binary data, do not use the DELIMIT parameter of the SEND command to tell Expedite Base/400 to use delimiters to mark records. These characters may be confused with the actual data in the binary file. The following is an example of how to use the DATATYPE(B) option of the SEND command.

```
send fileid(library/binary) account(act1) userid(user04) class(pcbin)
datatype(b);
```

#### Receiving binary files from an ASCII operating system

You do not need to use any special RECEIVE parameters to receive binary files. Do not specify DELIMITED(C) on the RECEIVE command when receiving binary data. Specifying DELIMITED(C) tells Expedite Base/400 to look for carriage-return and line-feed characters at the end of the record. These characters may be confused with the actual data in the binary file. The following is an example of how to use the RECEIVE command when receiving binary data.

```
receive fileid(library/binary(rcv)) account(act1) userid(user05) class(pcbin)
delimited(c);
```

The sender on an ASCII system must tell the Expedite product that the data is binary. The Expedite Base product then sends the data without translating it from ASCII to EBCDIC.

## Communicating with host operating systems

This section tells you how to communicate with other host systems using interfaces such as Expedite Base/MVS and Expedite/CICS.

#### Sending text files to a host operating system

Text files usually contain data that you can read. When you send text data to another host system, make certain the file will be readable on the receiving system. There are two ways to do this by using the DELIMIT option of the SEND command:

This option	Will have this effect:
DELIMIT(L)	Expedite Base/400 places a 2-byte record length at the beginning of each record. The receiving host system then removes the lengths and separates the records as the lengths indicated.
DELIMIT(C)	Expedite Base/400 inserts carriage-return and line-feed characters (X'0D0A') at the end of each record. The receiving host system then removes these characters and ends the records where indicated by the delimiters before their removal.  <b>NOTE:</b> Text files sent with the DELIMIT(C) option are also readable on ASCII systems.

The following is an example of how to use the DELIMIT(C) option of the SEND command.

```
send fileid(library/mytext) account(act1) userid(user03) class(text) delimit(c);
```

#### Receiving text files from a host operating system

When Expedite Base/400 receives files, it looks at the CDH to determine whether or not the data contains record delimiters, such as carriage-return and line-feed characters. If the CDH indicates that these delimiters are present, Expedite Base/400 removes them, but ends the records where indicated by the delimiters before their removal.

Because all interfaces do not send a CDH, you must tell Expedite Base/400 what to do with records received if there is no CDH. To do this, use the `DELIMITED` parameter of the `RECEIVE` command. The value that you specify on the `DELIMITED` parameter must correspond to the way the sending host system delimited the records. For example:

```
receive fileid(library/textout(rcv)) account(act1) userid(user05)class(text)
delimited(c);
```

#### [Sending binary files to a host operating system](#)

Binary files usually contain data that you cannot read, such as executable programs and computerized drawings. When you send binary data to another host, use the `DATATYPE(B)` option of the `SEND` command. If you need to delimit records when you send the data, use the `DELIMIT(L)` option of the `SEND` command. Do not use the `DELIMIT(C)` option of the `SEND` command when sending binary data. The `DELIMIT(C)` option tells Expedite Base/400 to put carriage-return and line-feed characters at the end of every record. These characters may be confused with the actual data in the binary file. The following is an example of how to use the `DELIMIT(L)` option of the `SEND` command:

```
send fileid(library/binary) account(act1) userid(user04) class(hostbin)
datatype(b) delimit(l);
```

#### [Receiving binary files from a host operating system](#)

When receiving files, Expedite Base/400 looks at the CDH to determine whether or not the data is delimited. If the data is delimited, Expedite Base/400 uses the delimit type specified in the CDH to format the records when it receives the data.

Because all interfaces do not send a CDH, you must tell Expedite Base/400 what to do with records received if there is no CDH. To do this, use the `DELIMITED` parameter of the `RECEIVE` command. The value you specify on the `DELIMITED` parameter must correspond with the way the sending host system delimited the records. Do not specify `DELIMITED(C)` in the `RECEIVE` command when receiving binary data. Specifying `DELIMITED(C)` tells Expedite Base/400 to look for carriage-return and line-feed characters at the end of the record. These characters may be confused with the actual data in the binary file. The following is an example of how to use the `DELIMITED` option of the `RECEIVE` command.

```
receive fileid(library/binary(rcv)) account(act1) userid(user05) class(hostbin)
delimited(l);
```

#### [Sending AS/400 programs](#)

Expedite Base/400 cannot send AS/400 programs directly. You must convert the AS/400 program to a format supported by Expedite Base/400 before sending it. A simple way to do this is by using the `SAVOBJ` command to place the program into a save file (`SAVF`). Then you can use Expedite Base/400 to send the file. See the command samples below.

CRTSAVF command

```
crtsavef file(savelib/savefile)
```

SAVOBJ command

```
savobj obj(pgmname) lib(pgmllib) dev(*savef) objtype(*pgm) savef(savelib/savefile)
```

Expedite Base/400 SEND command

```
send fileid(savlib/savefile) account(act1) userid(user04) class(savefile);
```

#### Receiving AS/400 programs

Expedite Base/400 cannot receive AS/400 programs directly. The AS/400 program must be sent in a format supported by Expedite Base/400 so you can use Expedite Base/400 to receive the file. After receiving the file, you can use the RSTOBJ command to restore the file to program object format. See the command samples below.

CRTSAVF command

```
crtsave file(rcvsavlib/rcvsavf)
```

Expedite Base/400 RECEIVE command

```
receive fileid(rcvsavlib/rcvsavf(mbr)) account(act1) userid(user01) class(savefile);
```

RSTOBJ command

```
rstobj obj(pgmname) savlib(pgmllib) dev(*savf) objtype(*pgm)  
savf(rcvsavlib/rcvsavf) rstlib(*savlib)
```

## Communicating with older Information Exchange interfaces

The CDH is an important aid to file transfer. Some interfaces do not support the CDH, but Expedite Base/400 can still communicate with these interfaces. However, there are restrictions on some features that depend on the CDH. These restrictions include the following:

- Expedite Base/400 ignores the AUTOEDI parameter of the RECEIVE command in files received from older interfaces.
- Expedite Base/400 cannot format data received from older interfaces based on CDH information, because no CDH is received from older interfaces. It formats data as indicated in the DELIMITED parameter of the RECEIVE command. Expedite Base/400 ignores the DLMOV-ERRIDE parameter of the RECEIVE command.
- Some of the information displayed in the RECEIVED record (for example, the DESCRIPTION parameter) is not available for files received from older interfaces.
- When you use a RECEIVE command with NONEDIONLY(Y), or a RECEIVEEDI command with EDIONLY(Y), Expedite Base/400 does not receive files that are lacking a CDH, or that have a CDH indicating the file is not EDI data.

Any Information Exchange interface before Release 3.0 does not support the CDH.



## Sending and receiving EDI data

---

One of the primary uses of Expedite Base/400 is sending and receiving data formatted for electronic data interchange (EDI). Expedite Base/400 provides a single set of commands for all EDI data transmission: SENDEDI and RECEIVEEDI. The following sections explain how these commands work.

For information on sending and receiving compressed EDI data, see Appendix D, “Using data compression.”

### Overview of the SENDEDI command

The SENDEDI command is used to send data that is formatted in EDIFACT, UCS, UN/TDI, and X12 formats to Information Exchange.

A group of EDI transactions with a single destination address is an EDI envelope. Expedite Base/400 uses several different EDI envelope definitions. The definition that is used depends on the type of EDI data contained in the envelope. See “Using EDI envelopes” on page 30 for details on how SENDEDI defines the envelopes.

Expedite Base/400 can transmit multiple EDI envelopes with different addresses from a single file with a single SENDEDI command. It can also transmit multiple types of EDI data from a single file. You can combine EDIFACT, UCS, UN/TDI, and X12 data in one file and transmit it to multiple Information Exchange destinations with one SENDEDI command.

Using the SENDEDI command, you can send data to Information Exchange without specifying an Information Exchange destination. The SENDEDI command matches the EDI destinations contained in the EDI data to Information Exchange destinations.

SENDEDI determines where to send an EDI envelope by examining the contents of that envelope. The location of the destination within an EDI envelope is determined by the envelope definition (the type of EDI data being transmitted). See “Using EDI envelopes” on page 30 for information on where in the envelope SENDEDI finds the envelope definition and the EDI destination.

To determine an EDI envelope's destination, the SENDEDI command searches the *EDI qualifier table* for the name of the appropriate *EDI destination table* and *centralized alias table*. After checking the EDI qualifier table, the SENDEDI command looks for the envelope's destination in

the EDI destination table. If Expedite Base/400 finds the destination, it sends the data to the Information Exchange destination corresponding to the EDI destination. If Expedite Base/400 does not find the destination in the EDI destination table, it uses the centralized alias table to resolve the EDI destination. See “Sending EDI data” on page 31 for more information.

You can still send EDIFACT or X12 data to an Information Exchange destination when you know that Expedite Base/400 cannot resolve the destination from information in the tables. Place ZZ in the qualifier ID of the EDI destination in the file you are sending. Then you can use the Information Exchange account ID and user ID to specify the EDI destination. See “Bypassing tables” on page 37 for more information.

Expedite Base/400 also enables you to send the same EDI envelope to more than one person at a time. To do this, create a *distribution list* and send the envelope to the list. For more information on distribution lists, see “Using EDI distribution lists” on page 39.

## Using EDI envelopes

When you send EDI transactions, you can group the EDI transactions for a single destination within a single envelope. The EDI envelope definitions for each EDI data type are described here.

This EDI data type:	Defines the EDI envelope as the data between and including the:
EDIFACT	UNA (or UNB) and UNZ segments.
UCS	BG and EG segments.
UN/TDI	SCH (or STX) and END segments.
X12	ISA and IEA segments.

The location of the destination within an EDI envelope is determined by the type of EDI data being translated. Expedite Base/400 determines the EDI destination as described here.

This EDI data type:	Uses this segment for the EDI destination:
EDIFACT	UNB. SENDEDI takes the destination from data element 0010 in composite data element S003 (interchange recipient). It takes the EDI qualifier from the data element 0007 in composite data element S003 (interchange recipient).
UCS	BG. SENDEDI takes the EDI destination from the application receiver's code (BG04) element.
UN/TDI	STX. SENDEDI takes the destination from the first subelement of the UNTO element (the recipient code address). If it does not find the recipient code, it uses the second subelement of the UNTO element (the recipient clear address) as the actual Information Exchange account and user ID.
X12	ISA. SENDEDI takes the actual EDI destination from the interchange receiver ID element (ISA08). It takes the EDI qualifier from the interchange ID qualifier element (ISA07).



## Data between EDI segments

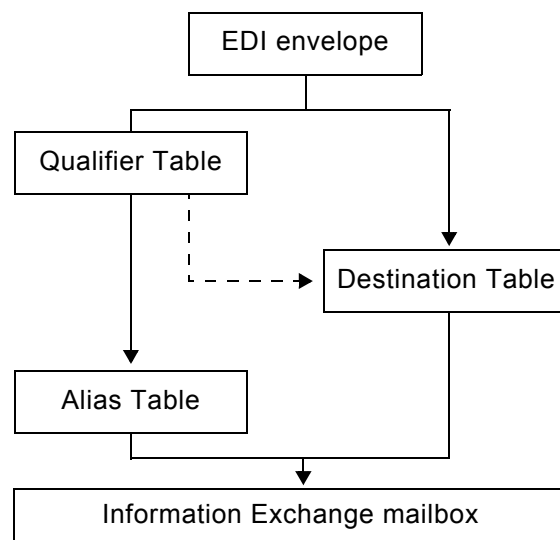
If you insert certain characters between segments when preparing EDI information, Expedite Base/400 does not consider these characters as part of the EDI data. The characters that Expedite Base/400 ignores are new line ('15'x), blank ('40'x), carriage return ('0D'x), EOF ('1A'x), and tab ('05'x).

The SENDEDI command accepts EDI data that has these characters between segments, but it removes the characters before transmitting the data to Information Exchange. The commands which receive EDI data (RECEIVEEDI and RECEIVE with certain options) will accept EDI data that has these characters but will remove the characters before writing the file.

## Sending EDI data

The following sections give you information on the different ways you can send EDI data using the SENDEDI command. For detailed information on the format of this command, see "SENDEDI command" on page 135.

The following flowchart illustrates how the SENDEDI command locates EDI destinations in most cases.



**NOTE:** SENDEDI can also locate destinations using actual Information Exchange destinations within the EDI data and distribution lists. If you plan to use Information Exchange destinations or distribution lists, see "Bypassing tables" on page 37 and "Using EDI distribution lists" on page 39 for more information.

The SENDEDI command can use three tables to determine the Information Exchange destination from the receiver ID specified in the EDI data: qualifier, destination, and alias.

If a qualifier table (QUALTBL) is provided, this is the first table SENDEDI uses. Each entry in this table indicates an EDI destination table, a centralized alias table for a given EDI qualifier and EDI data type, or both. A qualifier table is not required for SENDEDI to function.

The next table SENDEDI uses is the EDI destination table. Each entry in this table indicates an account and user ID combination, an alias and alias name combination, or a distribution list that corresponds to the receiver ID. You can create as many EDI destination tables as you like; each table can have as many entries as you need. No particular filename is required for destination tables. For information on the default table names used by the different EDI data formats, see the table in step 2 below.

If you specified a centralized alias table in QUALTBL, and you either did not specify an EDI destination table or SENDEDI found no match in the EDI destination table, SENDEDI sends the EDI data using the centralized alias table as the ALIAS parameter and the receiver ID from the data as the ALIASNAME.

The SENDEDI command resolves the destination using the following procedure:

1. SENDEDI finds the receiver ID and the qualifier in the EDI data being sent. Because the UCS and UN/TDI standards do not have a qualifier, the qualifier for these two data types defaults to blanks.
2. If a qualifier table (QUALTBL) exists, SENDEDI searches the qualifier table for the first entry that matches both the data type and the qualifier for the EDI envelope being processed.

If SENDEDI finds a matching entry in the qualifier table, and both the EDI destination and the centralized alias table are found on that entry, SENDEDI uses the EDI destination table first to resolve the address. If no match is found, SENDEDI uses the centralized alias table. If only one of these tables is provided, SENDEDI uses that table.

When SENDEDI does not find a match, it leaves the centralized alias blank, and the EDI destination table name defaults as follows:

This EDI data type:	Defaults to this EDI destination table name:
EDIFACT	TTABLE $xx$ , where $xx$ is the first 2 characters of the qualifier. If the qualifier is blank, or is not present, the name TTABLE__ (TTABLE plus 2 underscores) is used.
UCS	TTABLE01.
UN/TDI	IEUNTDI.
X12	TTABLE $xx$ , where $xx$ is the 2-character qualifier. For X12, a blank qualifier always defaults to account and userid. SENDEDI never treats TTABLE $xx$ as a match if $xx$ is blank.

3. If you specify the EDI destination table, or if SENDEDI sets it by default, SENDEDI searches that table for a matching receiver ID (EDIDEST). If SENDEDI finds a matching receiver ID, it sends the EDI data to the Information Exchange destination specified in that entry. If no match is found, or if a centralized alias table was coded in the qualifier table, SENDEDI sends the EDI data using the centralized alias table as the ALIAS parameter and the receiver ID from the data as the ALIASNAME.

When SENDEDI cannot find a destination in the qualifier table or the destination table, SENDEDI determines the Information Exchange destination for each EDI data type as follows:

If the EDI data is:	SENDEDI determines the Information Exchange destination as follows:
EDIFACT, and the qualifier, subelement 0007 in composite data element S003, is ZZ	SENDEDI splits the receiver code, which is subelement 0010 in composite data element S003 (interchange recipient), into the account ID and user ID. The account ID and user ID are separated by one or more blanks, a period, or a slash.
EDIFACT, and the qualifier, subelement 0007 in composite data element S003, is not ZZ	This is an error, and Expedite Base/400 does not send the data.
UCS	You cannot bypass tables when using this format. The EDI destination must be entered in the BG segment in data element B04. This is an error, and Expedite Base/400 does not send the data.
UN/TDI, and the recipient code (UNTO:1) is not specified	SENDEDI splits the recipient clear code (UNTO:2) into an account ID and user ID. The account ID and user ID are separated by one or more blanks, a period, or a slash.
UN/TDI, and the recipient code (UNTO:1) is specified	This is an error, and Expedite Base/400 does not send the data.
X12, and the qualifier is ZZ or blank	The SENDEDI command splits the receiver ID into an account ID and a user ID. It uses the first 7 characters as the account ID and the last 8 characters as the user ID.
X12, and the qualifier is not ZZ or blank	This is an error, and Expedite Base/400 does not send the data.

### Using EDI qualifier tables

Expedite Base/400 converts EDI destinations to Information Exchange destinations differently for different types of EDI data. The EDI qualifier table tells the SENDEDI command how to resolve a destination based on two pieces of information:

- The type of EDI data: X12, UCS, EDIFACT, or UN/TDI.
- The ID qualifier for a specific type of data; for example, 01 for an X12 DUNS number.

Using the above data, the EDI qualifier table provides Expedite Base/400 with one or both of the following:

- The file name of an EDI destination table that Expedite Base/400 can use to convert the actual EDI destination to an Information Exchange destination.

This table is used first if both this table and the centralized alias table are provided.

- The centralized alias table name to use if Expedite Base/400 cannot find the EDI destination in the EDI destination table.

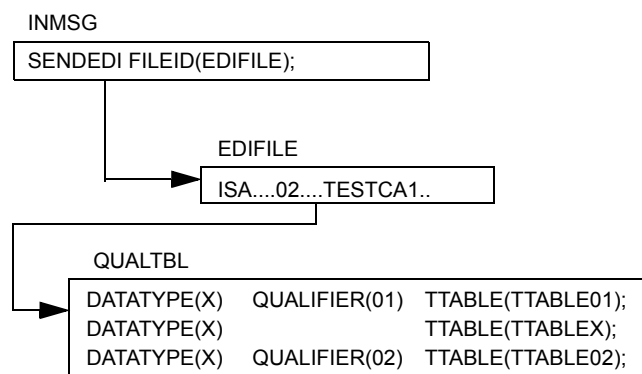
This table is used second if both an EDI destination table and a centralized alias table are provided. This table is used first if it is the only table provided.

Give the name QUALTBL to the file in which you intend to create the EDI qualifier table. For a description of the EDI qualifier table format, see “EDI qualifier table entry format” on page 43.

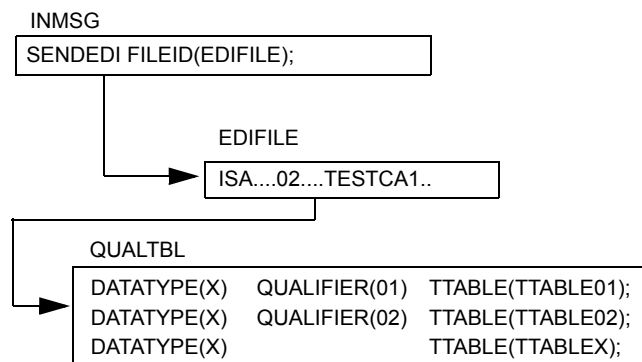


**NOTE:** The SENDEDI command does not require the EDI qualifier table to function. If the table is not present, Expedite Base/400 uses defaults for the EDI destination table name, as described earlier in this section.

The following figure shows the importance of order among qualifier table entries for the same data type. This example does not show the EDI destination tables:



Expedite Base/400 uses TTABLEX to resolve the destination because its qualifier table entry is read before the more explicit qualifier table entry specifying TTABLE02. If TTABLE02 should have been used before TTABLEX, reorder the entries in the qualifier table as shown:



## Using EDI destination tables

Think of an EDI destination table as a list of EDI destinations paired with Information Exchange destinations. The `SENDEDI` command resolves destinations by searching for an EDI destination and then using the corresponding Information Exchange destination as the actual address for an envelope. See “EDI destination table entry format” on page 44 for a description of the EDI destination table format.

To send EDI data to a destination defined in an EDI destination table, follow these steps:

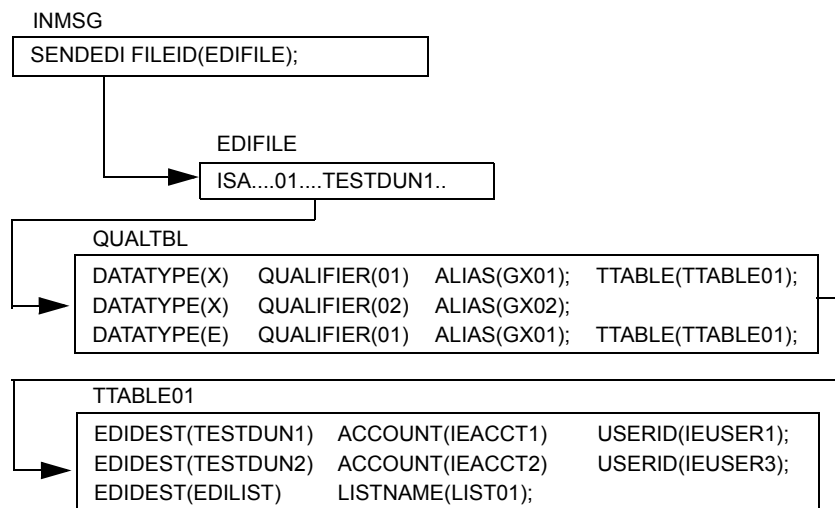
1. Build an EDI destination table that contains your EDI destination and the corresponding Information Exchange destination.
2. Build an EDI qualifier table that contains the name of your EDI destination table and the type of EDI data you want to send.



**NOTE:** If you use the default name for your EDI destination table, you do not have to build an EDI qualifier table.

3. Use the SENDEDI command to send the EDI envelope.

The following figure shows the tables used to send an X12 file to a trading partner with a qualifier of 01 and an EDI destination of TESTDUN1.



Expedite Base/400 sends the data in the example above to a destination with an Information Exchange account ID of IEACCT1 and a user ID of IEUSER1.

## Using Information Exchange alias tables

You might find it time-consuming to maintain EDI destination tables in multiple locations. To avoid this problem, the SENDEDI command allows you to use centralized Information Exchange alias tables. These permanent tables reside within Information Exchange and can convert EDI destinations into Information Exchange destinations. You can make them available to all Information Exchange users (*global alias tables*), members of a particular account (*organization alias tables*), or a single user (*private alias tables*). The EDI qualifier table and the EDI destination table determine which, if any, of these alias tables should be used.



**NOTE:** There are two ways for you to create and maintain alias tables:

1. Using Information Exchange Administration Services (see *Using Information Exchange Administration Services*)
2. Using the DEFINEALIAS command (see “DEFINEALIAS command” on page 94)

To send EDI data to a destination defined in an Information Exchange centralized alias table, follow these steps:

1. Add the target EDI destination to a centralized alias table using Information Exchange Administration Services.
2. Build an EDI qualifier table that contains the name of the Information Exchange centralized alias table and the type of EDI data you want to send.

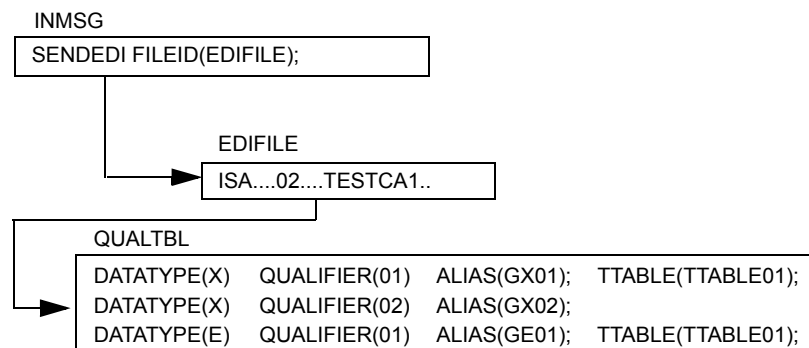


**NOTE:** Expedite Base/400 contains a sample EDI qualifier table which defines standard alias tables for various EDI data formats. In some Information Exchange installations, such as the U.S., these standard alias tables are available through your Information Exchange connection. You can add your EDI destinations to these tables as needed.

If you use an EDI destination table, be sure that your EDI destination table does not contain the name of the target destination.

3. Use the SENDEDI command to send the file containing your EDI data.

The following figure shows the tables used to send an X12 file to a trading partner with a qualifier of 02 and an EDI destination of TESTSCA1. This particular example does not include an EDI destination table.



The data in the example above is sent to an Information Exchange destination with the Information Exchange alias table GX02 and an alias name of TESTSCA1.

## Bypassing tables

If you need to send only EDIFACT, X12, or UN/TDI data to an Information Exchange destination, you can bypass the tables and send the EDI information directly.

To send EDIFACT or X12 data to an Information Exchange destination contained within the EDI data, follow these steps:

1. Place **ZZ** in the qualifier.
2. Use the Information Exchange account ID and user ID as the actual EDI destination.
  - When sending X12 data, separate the account ID and user ID with at least one blank.
  - When sending EDIFACT data, separate the account ID and user ID with a period, a slash, or a blank.

- Use the SENDEDI command to send the file that contains your EDI data.

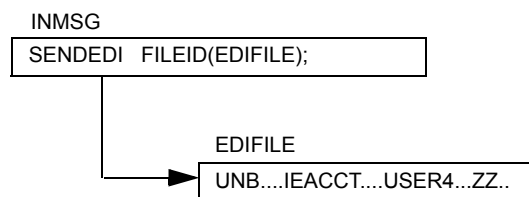


**NOTE:** When you use a ZZ qualifier, Expedite Base/400 tries to resolve the destination by searching the tables. When it does not locate the destination in the tables, it sends the data to the Information Exchange account and user ID specified. For X12 data only, if you do not want Expedite Base/400 to refer to the tables first, use a blank qualifier instead of a ZZ qualifier.

To send UN/TDI data to an Information Exchange destination contained within the EDI data, follow these steps:

- Do not specify the recipient code (UNTO:1).
- Place the Information Exchange account and user ID in the recipient clear subelement (UNTO:2).
- Use the SENDEDI command to send the file that contains your EDI data.

The following figure shows the route that Expedite Base/400 uses to send an EDIFACT file to a trading partner with an Information Exchange destination of IEACCT2 IEUSER4.



Expedite Base/400 sends the data in the example above to an Information Exchange destination with an Information Exchange account ID of IEACCT2 and a user ID of IEUSER4.

If you are sending UCS data, you cannot bypass the use of tables. You must have one of the following files to send UCS data:

- A TTABLEXX file specifying an Information Exchange account and user ID for the UCS destination address.
- A QUALTBL file identifying an Information Exchange alias table. The alias table must have an entry with the UCS destination address and the associated Information Exchange account and user ID.
- A QUALTBL file identifying the destination table to be used for translating the UCS destination address to an Information Exchange account and user ID.

### Intersystem addressing for EDIFACT and UN/TDI

Intersystem addressing is available with EDIFACT and UN/TDI data when using the SENDEDI command.

Instead of using an alias table to address EDI files, you can place an Information Exchange address in the EDIFACT or UN/TDI header specifying the appropriate IDs in the following order:

- System ID (optional if the sender and receiver are using the same system)



2. Account ID
3. User ID

All of the above IDs must be separated by one of the following:

- Period (.)
- Slash (/)
- One or more blank spaces

For EDIFACT tables, SENDEDI splits the receiver code (subelement 0010 in composite data element S003, Interchange Recipient) into the system, account, and user IDs.

For UN/EDT data, SENDEDI splits the recipient clear code (UNTO:2) into the system, account, and user IDs.

## Using EDI distribution lists

To send EDI data to an Information Exchange list, follow these steps:

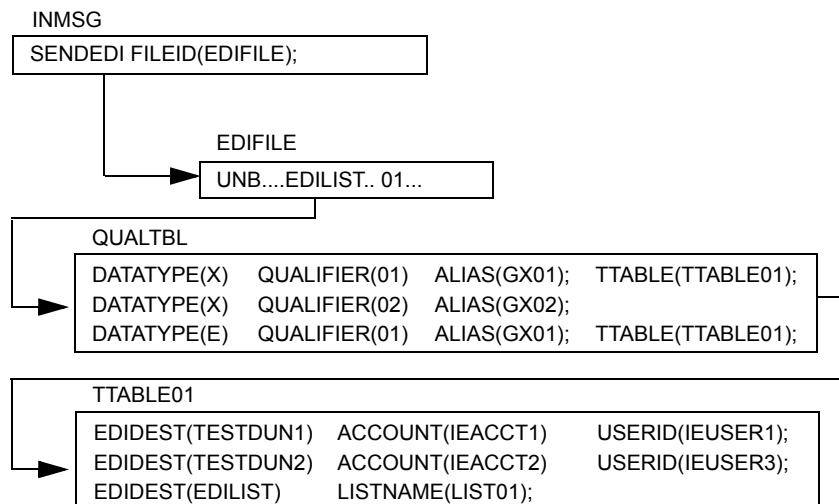
1. Define the Information Exchange list using the LIST command or Information Exchange Administration Services.
2. Build an EDI destination table that contains your EDI destination and the corresponding Information Exchange list name.
3. Build an EDI qualifier table that contains the name of your EDI destination table and the type of EDI data it references.



**NOTE:** If you use a default name for your EDI destination table, you do not have to build an EDI qualifier table.

4. Use the SENDEDI command to send the file that contains your EDI data.

The following figure shows the tables used to send an EDIFACT file to a list of trading partners with an EDI destination of EDILIST and a qualifier of 01.



Expedite Base/400 sends the data in the example above to an Information Exchange destination with a list name of LIST01.

To create a permanent list and store it in Information Exchange, see the “LIST command” on page 101.

You can also use Information Exchange Administration Services to create a permanent list. For more information on using Information Exchange Administration Services to create and update this type of list, see *Information Exchange Administration Services User's Guide*.

## Specifying Information Exchange control fields

You can specify the following Information Exchange control fields by using the parameters within the SENDEDI command.

### Message name (MSGNAME) assignment

If you provide a MSGNAME parameter when using the SENDEDI command, SENDEDI uses this value for the Information Exchange message name. If you do not provide the MSGNAME parameter, the SENDEDI command generates the Information Exchange message name based on the type of EDI data transmitted. The following table describes how the SENDEDI command generates the message name:

This EDI data type:	Generates this MSGNAME:
EDIFACT data	Expedite Base/400 takes the MSGNAME from data element 0020 (Interchange Control Reference) of the EDI data. If the element exceeds 8 bytes in length, Expedite Base/400 uses the first 8 bytes. If the element is fewer than 8 bytes in length, Expedite Base/400 places it in the MSGNAME parameter value, left-justified and padded with blanks.
UCS data	Expedite Base/400 takes the MSGNAME from the interchange control number. Because the UCS interchange control number has a maximum length of 5 bytes, Expedite Base/400 places the interchange control number in the MSGNAME parameter value, left-justified and padded with blanks.
UN/TDI data	Expedite Base/400 takes the MSGNAME from the sender's reference field (SNRF) of the EDI data. If the SNRF exceeds 8 bytes in length, Expedite Base/400 uses the first 8 bytes. If the SNRF is fewer than 8 bytes in length, Expedite Base/400 places it in the MSGNAME parameter value, left-justified and padded with blanks.
X12 data	Expedite Base/400 takes the MSGNAME from the last 8 bytes of the interchange control number of the X12 data.

### Message sequence number (MSGSEQNO) assignment

If you provide a MSGSEQNO parameter in the SENDEDI command, the SENDEDI command uses this value for the Information Exchange message sequence number. If you do not provide the MSGSEQNO parameter, the SENDEDI command generates the Information Exchange message sequence number in the following way for all EDI data types.

The SENDEDI command counts each EDI envelope transmitted from a single data file. It places the count in the MSGSEQNO of the Information Exchange messages corresponding to the EDI envelopes. SENDEDI formats MSGSEQNO as a series of numeric characters ranging from 00001 to 99999. For example, three EDI envelopes sent from a single file would have the following MSGSEQNO values:

- 00001 for the first EDI envelope in the file
- 00002 for the second EDI envelope in the file
- 00003 for the last EDI envelope in the file

Each time you use the SENDEDI command for a new file that contains EDI envelopes, or when the MSGSEQNO reaches 99999, the MSGSEQNO counter resets to 00001.

### Message class (CLASS) assignment

If the SENDEDI command CLASS parameter is not blank, the SENDEDI command uses this value. If you do not provide the CLASS parameter, the SENDEDI command generates the parameter based on the type of EDI data transmitted.

#### EDIFACT and UN/TDI data

For EDIFACT and UN/TDI data, Expedite Base/400 takes the class from the Application Reference field (APRF) of the EDI data. If the APRF exceeds 8 bytes in length, Expedite Base/400 uses the first 8 bytes. If the APRF is fewer than 8 bytes in length, Expedite Base/400 places the APRF in the Class field, left-justified and padded with blanks. If the APRF is not present, Expedite Base/400 sets the CLASS parameter as follows:

This EDI data type:	Defaults to this class:
EDIFACT	#EE
UN/TDI	#EU

#### UCS and X12 data

For UCS and X12 data, Expedite Base/400 sets the CLASS parameter as follows:

This EDI data type:	Defaults to this class:
UCS	#EC
X12	#E2

### SENDEDI response records

The SENT record provides a record of the EDI envelopes sent by the SENDEDI command. For a description of the format of this record, see “SENT record” on page 179.

The NOTSENT record provides a record of the EDI envelopes not sent by the SENDEDI command because of a destination verification failure. NOTSENT records are provided only when the VERIFY parameter with a value of *c* or *g* is specified on the SENDEDI command. For a description of the format of this record, see “NOTSENT record” on page 170.



**NOTE:** When the SENDEDI command does not complete, you can use the SENT records in the message response file (OUTMSG) to determine which EDI envelopes were sent and you can use the NOTSENT records to determine which envelopes were not sent.

## Receiving EDI data

Use the RECEIVEEDI command to receive EDI data from Information Exchange. You can receive multiple EDI envelopes that contain different types of data with a single RECEIVEEDI command.

The RECEIVEEDI command is similar to the RECEIVE command, but differs in the ability to reformat received data based on EDI segment delimiters. The EDIOPT parameter of the RECEIVEEDI command controls this function. For more information on the format of this command, see “RECEIVEEDI command” on page 122.

If your trading partner sends you EDI data that includes a CDH, the RECEIVE command recognizes the EDI format of the data and reformats it accordingly. The AUTOEDI parameter of the RECEIVE command controls this function. Expedite Base/400 always prepares a CDH when sending data. Information Exchange interfaces prior to Release 3.0 do not support the CDH.

Using the RECEIVEEDI command does not guarantee that you receive only EDI data. Use the CLASS parameter to select only the class of messages you want to receive.

If the person sending you data uses the default user classes of the SENDEDI command, you can use the Information Exchange wildcard receive feature to simplify receipt. For example, by specifying #E? as the user class in the RECEIVEEDI command, you ask Information Exchange to return only files that have a user class beginning with #E. This includes all files sent with the default EDI user classes.



**NOTE:** The SENDEDI command places a single EDI envelope in an Information Exchange message group. The RECEIVEEDI command expects each EDI envelope to be contained in a separate Information Exchange message group. If you put multiple EDI envelopes in a single file and send them using the SEND (not the SENDEDI) command, the results are unreliable.

If the CDH indicates the data is not EDI, the RECEIVEEDI command does not perform any special EDI processing.

If there is no CDH, RECEIVEEDI attempts to process the files as EDI data. If this is not possible, Expedite Base/400 writes the data without reformatting.

## EDI qualifier table entry format

The EDI qualifier table resides in the file referenced by the name QUALTBL. Each entry in this table indicates an EDI destination table, a centralized alias table for a given EDI qualifier or EDI data type, or both.

### Format

`datatype(data type) qualifier(qual) ttable(ttable) alias(alias);`

### Parameters

#### **datatype**

Indicates the EDI data type for this entry.

blank	For all EDI data types. This entry matches any of the supported EDI data types. This is the default.
x	For X12 data
c	For UCS data
e	For EDIFACT data
u	For UN/TDI data

#### **qualifier**

Indicates the EDI qualifier for this entry. If this parameter is blank, the entry matches any EDI qualifier. Use 1 to 4 alphanumeric characters. The default is blank.

#### **ttable**

Indicates the name of the EDI destination table. If you specify a table in the TTABLE parameter, SENDEDI uses that table to try to resolve the Information Exchange destination. If you do not specify a destination table, SENDEDI does not use a table for EDI data matching this entry. Use 1 to 33 alphanumeric characters.

#### **alias**

Indicates the centralized alias table for this entry. If you specify this parameter and SENDEDI does not find the destination in the EDI destination table, SENDEDI uses this alias table with the EDI receiver ID as the alias name.

blank	No centralized alias table is used. This is the default.
gxxx	Global alias table, where <i>xxx</i> identifies a 1- to 3-character table name.
oxxx	Organizational alias table, where <i>xxx</i> identifies a 1- to 3-character table name.
pxxx	Private alias table, where <i>xxx</i> identifies a 1- to 3-character table name.

You can use a “#” (number-sign) symbol to define comments. Type any information after #. The program ignores the characters after # until it encounters a new line. You do not have to end comments with #. Expedite Base/400 considers the end of the line to be the end of that comment. If you want the comment to continue, begin a new line with #.

## EDI destination table entry format

The EDI destination table resides in a file specified by the `TTABLE` parameter in the EDI qualifier table. Each entry in this table indicates the Information Exchange destination associated with a given EDI receiver ID.

### Format

**edidest**(*edidest*)

alias(*alias*) aliasname(*aliasname*);

or

sysid(*sysid*) account(*account*) userid(*userid*);

or

account(*account*) userid(*userid*);

or

listname(*listname*);

### Parameters

#### **edidest**

Indicates the EDI receiver ID for this entry. If this parameter matches the receiver ID from the EDI data, Expedite Base/400 sends the message to the Information Exchange destination specified by the parameters you select.

Use 1 to 35 alphanumeric characters.

#### **alias**

Indicates the alias table type and table name.

**blank** No centralized alias table is used. This is the default.

**gxxx** Global alias table, where *xxx* identifies a 1- to 3-character table name.

**oxxx** Organizational alias table, where *xxx* identifies a 1- to 3-character table name.

**pxxx** Private alias table, where *xxx* identifies a 1- to 3-character table name.

#### **aliasname**

Indicates an alias name defined in the alias table. Use 1 to 16 alphanumeric characters.

#### **sysid**

Indicates the system ID of the destination user ID. You need this only if the specified account ID and user ID reside on another Information Exchange system. You can use it only with the `ACCOUNT` and `USERID` parameters. Use 1 to 3 alphanumeric characters.

#### **account**

Indicates the account name of the destination user ID. Use 1 to 8 alphanumeric characters.

#### **userid**

Indicates the destination user ID. Use 1 to 8 alphanumeric characters.

**listname**

Indicates the name of a previously defined list of account IDs and user IDs. Use 1 to 8 alphanumeric characters.

You can use a “#” (number-sign) symbol to define comments. Type any information after #. The program ignores the characters after # until it encounters a new line. You do not have to end comments with #. Expedite Base/400 considers the end of the line to be the end of that comment. If you want the comment to continue, begin a new line with #.





## Using session-level recovery

---

Session-level recovery is the default recovery level for Expedite Base/400. Session-level recovery does not require additional work files to run, as do the other methods of data recovery. When you use session-level recovery to transmit data and an error occurs, Expedite Base/400 stops the transmission and produces a SESSIONEND record with a return code. You can check this return code to determine the cause of the error and correct the problem.

Although retransmission of large amounts of data can take several hours, there are advantages to using session-level recovery. When you use session-level recovery, you do not need to be concerned with determining which files were successfully sent and which need to be retransmitted. With session-level recovery, if transmission stops because of an error, you send or receive all of the data again.

### Recovery considerations

**Attention:** Do not use the same account and user ID on multiple systems. If you start an Information Exchange session using session-level recovery while another Information Exchange session with the same account and user ID is running, Information Exchange ends the first session and proceeds with the second session. Information Exchange does not deliver data sent in the first session, and does not delete received data from the mailbox. This means that data received in the first session may be received again in error. To prevent this, use an account name and user ID on only one system at a time.

If you use multiple START and END commands in INMSG, you create an environment similar to that of checkpoint-level recovery. Each END command stops a session. Requests in each session complete even if a subsequent session ends in error. Only sessions that end in error require you to send or receive data again.

There is a limit to the amount of data you can send during a single Information Exchange session. If you plan to send very large amounts of data, check with your Global Services representative to find out what the limit is at your Information Exchange installation.

Customers using a leased line may select session-level recovery, because it is very unlikely the line will go down during the data transfer. Recovery is easier if you use session-level recovery, as it is an all-or-nothing data transfer. However, it is recommended that you use checkpoint-level

recovery when sending a large number of files because of the lengthy processing time required. Customers have experienced time-out problems when sending or receiving more than 700 files (the size of the files does not matter). Use checkpoint-, file-, or user-level recovery instead.

When using session-level recovery, Information Exchange commits the data sent or received only at the end of the session. When Expedite Base/400 requests an end to the Information Exchange session, Information Exchange does not respond until the commit process is completed. If a large number of files are transferred during the session, the processing takes longer, especially during prime business hours. It is possible for Expedite Base/400 to time out while waiting for Information Exchange to complete the process of committing the data for the entire session. In this case, Expedite ends the session with a 29999 return code, "Session end response failure." It may not be clear, then, whether the session ended successfully. One way to determine if a session was successful is to use the CHECK parameter in the START command. See "START command" on page 145 for more information.

When Expedite Base reports the 29999 SESSIONEND return code for a session-level recovery session, you should switch to checkpoint-level, file-level, or user-level recovery for future sessions that contain a similar number of commands.

You can also use other methods to determine if the previous session was or was not successful.

If you were only receiving files from your Information Exchange mailbox, make sure all data was received by verifying that it is no longer in your mailbox. You can do this by viewing your mailbox with Information Exchange Administration Services or by running a QUERY command to get a list of AVAILABLE response records for each file in your mailbox. If the data is still in your mailbox, switch from session-level recovery to checkpoint-level recovery and run the session again to receive the data.

If you were sending files, you must check the audit trail to see if the files were sent. You can do this by using Information Exchange Administration Services, or by using Expedite to request an audit be sent to your mailbox. If the files were not sent, switch to checkpoint-level recovery and run the session again.

## Processing the message response file

When you transmit data, Expedite Base/400 processes your message command file and creates a message response file. The following sections give you the information you need to process the message response file when you use session-level recovery.

### Examining response records

Response records are free-format records with the same syntax as commands. See "Understanding command syntax" on page 4 for a description of the command syntax used with Expedite Base/400. Response records always start at the beginning of a line. However, parameters in response records can occur in any position in a record and in any order. In addition, all parameters in a response record might not appear.

We suggest that you be prepared for the possible addition of new parameters to existing response records and of entirely new response records in the future.

When examining response records, consider the following:

- Assume a default value if you do not get the response record parameter that you are expecting. This is not an error.

- If a response record parameter has a longer length than you expect, truncate the parameter.
- Be prepared to handle parameters that are split across records. Splitting can occur if the parameter is longer than the record length of your response file.

## Checking return codes

To be sure that Expedite Base/400 finished processing the message command file, check the return code in the SESSIONEND record.

The following table lists the return codes you can receive, explains them, and tells you what to do when you receive them.

If the SESSIONEND return code is:	Then:	Do the following:
00000	Expedite Base/400 finished processing the command file. There were no processing problems, and all of the command RETURN records also contain 0 return codes. All of your requests completed normally.	Nothing. Informational only.
28010	The Information Exchange session completed normally, but Expedite Base/400 could not complete all the commands in your message command file.	Check the command RETURN records to see which commands did not complete. Commands that completed normally have 0 RETURN records. Commands that did not complete normally have nonzero RETURN records, and, if your system is set up to do so, can display error descriptions.
Anything other than 00000 or 28010	Expedite Base/400 did not finish processing the command file. Your Information Exchange session failed, and none of the file transfer requests in your message command file were completed. No mail has been placed in your trading partner's mailbox or removed from your mailbox.	Correct the problem and restart Expedite Base/400. The SESSIONEND record might include an error description to help you find the problem, if your system is set up to display error descriptions. A command RETURN record might contain the same code and description. If OUTMSG contains no return code, check the OUTPRO file for the error.

If the error is due to a problem with a command, such as a syntax error, the command in error is followed by a RETURN record with the same return code as the SESSIONEND record. If you do not find the RETURN record in OUTMSG, the error might be in INPRO, and you would find the RETURN record in OUTPRO.

The SESSIONEND and RETURN records often include an error description. Detailed error descriptions are included in Chapter 11, “Expedite Base/400 messages and codes.”

If the return code is not 00000 or 28010, some data might have been written to your files before the Information Exchange session failed. Check OUTMSG. If the RECEIVE commands you entered have echoed to OUTMSG, Expedite Base/400 may have written new data to the files referenced in the RECEIVE commands.

Nonfile transfer requests can complete even if the Expedite Base/400 SESSIONEND return code is other than 28010 or 00000. These requests include:

- ARCHIVEMOVE
- AUDIT
- CANCEL
- DEFINEALIAS
- GETMEMBER
- LIST
- LISTLIBRARIES
- LISTMEMBERS
- LISTVERIFY
- MESSAGEINFO
- PURGE
- SESSIONINFO
- TESTMSG

If these requests are followed by a RETURN(00000) record in the OUTMSG file, they completed normally, and you do not need to issue them again. If you issue the AUDIT, TESTMSG, or LISTVERIFY commands again, you will receive two identical responses in your mailbox, instead of receiving only one.

## Checking the SENT and RECEIVED records

Expedite Base/400 produces SENT records for every file sent to Information Exchange and produces RECEIVED records for every file received from Information Exchange. These records provide you with a history of all Expedite Base/400 file transfers that took place during the last session.

A single RECEIVE command often produces multiple RECEIVED records, indicating that Expedite Base/400 has received multiple files as a result of a single command. Also, a single SENDEDI command can produce several SENT records, indicating that several EDI envelopes were sent with a single command.

If there is no SENT record, no file was sent. If there is no RECEIVED record, no file was received.

## OUTMSG examples

The following sections contain examples to help you understand OUTMSG processing. The INMSG file used as a basis for the OUTMSG examples follows.



**NOTE:** In the 'SESSIONEND(28010)' and 'SESSIONEND (not 00000 or 28010)' examples, INMSG has been altered to include a syntax error that provides you with several different OUTMSG files. Each OUTMSG has a different SESSIONEND record.

### INMSG example

```
#Send a file named test to act1 user01 in user class tdata
send fileid(library1/test) account(act1) userid(user01) class(tdata);

#Send an EDI file called edidata in user class editest
sendedi fileid(edidata) class(editest);

#Receive an EDI file with user class #e2 into ediout
receiveedi fileid(ediout) class(#e2);

#Receive a file called test with user class tdata
receive fileid(library/test(sample1)) class(tdata);
```

### Normal completion example - SESSIONEND(00000)

In the following OUTMSG example, all commands from the INMSG shown above complete successfully:

```
AUTOSTART SESSIONKEY(HSXVNW1M);
STARTED LASTSESS(0) RESPCODE(00000) SESSIONKEY (HSXVNW1M)
IEVERSION(04) IERELEASE(07);
RETURN(00000);

#send a file named test to act1 user01 in user class tdata#
SEND FILEID(LIBRARY1/TEST) ACCOUNT(ACT1) USERID(USER01) CLASS(TDATA);
SENT UNIQUEID(57586838) LENGTH(80);
RETURN(00000);

#send an edi file called edidata in user class editest#
sendedi fileid(library1/edidata) class(editest);
SENT UNIQUEID(75150113) LENGTH(2260) ACCOUNT(ACT1) USERID(USER01)
EDITYPE(X12)
DESTINATION(DUNS01) QUALIFIER(01) CONTROLNUM(000022223)
CLASS(EDITEST)
MSGNAME(00022223) MSGSEQNO(00001);
RETURN(00000);

#Receive an EDI File with user class #e2 into ediout
receiveedi fileid(library/ediout(sample1)) class(#e2);
RECEIVED ACCOUNT(ACT1) USERID(USER01) CLASS(#E2) CHARGE(5)
RECEIVER(ACT1 DEPT01) RECVQUAL(01)
SENDER(ACT2 DEPT02) SENDQUAL(01) CONTROLNUM(000022228)
FILEID(LIBRARY1/EDITEST(RCV1)) MSGDATE(040325) MSGDATELONG(20040325)
```

```
MSGTIME(182209) MSGSEQO(000021) MSGNAME(00022223) MSGSEQNO(00001)
SESSIONKEY(HXSVNW1M) DELIMITED(E) TIMEZONE(L)
SYSNAME(EXPMV00T) SYSLEVEL(460) DATATYPE(E) EDITYPE(X12)
SENDERFILE(USER01.EDIDATA.FILE) SENDERLOC(3380K /TSO203)
FILEDATE(040326) FILEDATELONG(20040326) RECFM(F) RECLLEN(80) RECDLM(E)
UNIQUEID(75150113)
SYSTYPE(21) SYSVER(1);
RETURN(00000);
```

```
#Receive a file called test with user class tdata#
```

```
receive fileid(library/test(sample1)) class(tdata);
RECEIVED ACCOUNT(ACT1) USERID(USER01) CLASS(TDATA) CHARGE(5)
FILEID(LIBRARY1/TEST(RCV1)) MSGDATE(040325) MSGDATELONG(20040325)
MSGTIME(182208) MSGSEQO(000023)
SESSIONKEY(HXSVNW1M) DELIMITED(N) SYSNAME(EXPMV00T) SYSLEVEL(460)
TIMEZONE(L)
DATATYPE(E) EDITYPE(UNFORMATTED) SENDERFILE(USER01.TEST.DATA)
SENDERLOC(3380K /TSO203) FILEDATE(040326) FILEDATELONG(20040326)
RECFM(FB) RECLLEN(80)
RECDLM(N) UNIQUEID(57586838) SYSTYPE(21) SYSVER(1);
RETURN(00000);

AUTOEND;
RETURN(00000);
SESSIONEND(00000);
```

#### Minor error example - SESSIONEND(28010)

In the following OUTMSG example, you see the result of a typographical error. The programmer intends to enter **edidata** as part of the SENDEDI command, but types **eiddata** instead. The result is a 23410 error after the SENDEDI command and a 28010 error in the SESSIONEND record. Note that the other requests in INMSG process successfully.

```
AUTOSTART SESSIONKEY(HXTB008G);
STARTED LASTSESS(0) RESPCODE(00000) SESSIONKEY (HXTB008G)
IEVERSION(04) IERELEASE(07);
RETURN(00000);
```

```
#Send a file named test to act1 user01 in user class tdata#
```

```
send fileid(test) account(act1) userid(user01) class(tdata);
SENT UNIQUEID(57586838) LENGTH(80);
RETURN(00000);
```

```
#Send an EDI file called edidata in user class editest#
```

```
sendedi fileid(eiddata) class(editest);

RETURN(23410) ERRDESC(Unable to open EDI send file.)
REASON(1004 - File EIDDATA not found.)
```

```
ERRTEXT(EXPLANATION: Expedite could not open the file indicated in the)
ERRTEXT(SENDEDI command. There may be a REASON parameter that gives more)
ERRTEXT(information about why the file could not be opened.)
ERRTEXT(USER RESPONSE: If there is a REASON parameter, check it for more)
ERRTEXT(information about the error. Also, make sure the file)
```

```

ERRTEXT(specified is not empty. Check the FILEID parameter for)
ERRTEXT(errors and try the command again.);

#Receive an EDI file with user class #e2 into ediout#

receiveedi fileid(library/ediout(sample1)) class(#e2);
RECEIVED ACCOUNT(ACT1) USERID(USER01) CLASS(#E2) CHARGE(5)
RECEIVER(ACT1 DEPT01) RECVQUAL(01)
SENDER(ACT2 DEPT02) SENDQUAL(01) CONTROLNUM(000022228)
FILEID(LIBRARY1/EDITEST(RCV1)) MSGDATE(040325) MSGDATELONG(20040325)
MSGTIME(182446) MSGSEQO(000025)
MSGNAME(00022223) MSGSEQNO(00001) SESSIONKEY(HXTB008G) DELIMITED(E)
TIMEZONE(L)
SYSNAME(EXPMV00T) SYSLEVEL(460) DATATYPE(E) EDITYPE(X12)
SENDERFILE(USER01.EDIDATA.FILE) SENDERLOC(3380K /TSO203)
FILEDATE(040326) FILEDATELONG(20040326) RECFM(F) RECLLEN(80) RECDLM(E)
UNIQUEID(75150113)
SYSTYPE(21)
SYSVER(1);
RETURN(00000);
#Receive a file called test with user class tdata#

receive fileid(library/test(sample1)) class(tdata);
RECEIVED ACCOUNT(ACT1) USERID(USER01) CLASS(TDATA) CHARGE(5)
FILEID(LIBRARY1/TEST OUT(RCVEDI1)) MSGDATE(040325)
MSGDATELONG(20040325) MSGTIME(182445) MSGSEQO(000027)
SESSIONKEY(HXTB008G) DELIMITED(N) SYSNAME(EXPMV00T) SYSLEVEL(460)
TIMEZONE(L)
DATATYPE(E) EDITYPE(UNFORMATTED) SENDERFILE(USER01.TEST.DATA)
SENDERLOC(3380K /TSO203) FILEDATE(040326) FILEDATELONG(20040326)
RECFM(FB) RECLLEN(80)
RECDLM(N) UNIQUEID(57586838) SYSTYPE(21) SYSVER(1);
RETURN(00000);

AUTOEND;
RETURN(00000);
SESSIONEND(28010)
ERRDESC(The IE session completed normally but not all requests were processed.)
ERRTEXT(EXPLANATION: The Information Exchange session completed normally, but)
ERRTEXT(not all of the requests in the command file, INMSG, were processed.)
ERRTEXT(or some requests generated warnings.)
ERRTEXT(USER RESPONSE: check the message response file, OUTMSG, to see)
ERRTEXT(which requests did not process normally. Correct)
ERRTEXT(those requests in a new INMSG retry the program.);

```

#### Severe error example - SESSIONEND (not 00000 or 28010)

In the following OUTMSG example, you see the result of a typographical error. The programmer intends to enter the SENDEDI command, but types **sendedk** instead. The result is a 15040 error after the SENDEDK command and a 15040 error in the SESSIONEND record. Note that although the SEND command preceding **sendedk** completed, no data is placed in the recipient's Information Exchange mailbox, because the session does not complete.

```

AUTOSTART SESSIONKEY(ISS5HK8S);
STARTED LASTSESS(0) RESPCODE(00000) SESSIONKEY (ISS5HK8S)

```

```
IEVERSION(04) IERELEASE(07);
RETURN(00000);

#Send a file named test to act1 user01 in user class tdata

send fileid(library1/test) account(act1) userid(user01) class(tdata);
SENT UNIQUEID(57586838) LENGTH(80);
RETURN(00000);

#Send an EDI file called edidata in user class editest

sendedk
RETURN(15040) ERRDESC(Command not recognized.)
ERRTEXT(EXPLANATION: You specified an unrecognized command in the command file.)
ERRTEXT(USER RESPONSE: Check the message response file, OUTMSG, )
ERRTEXT(profile response file, OUTPRO, or response work file, OUTWORK,
ERRTEXT(to determine which command produced the error.)
ERRTEXT(Correct the command file and retry the program.);
SESSIONEND(15040) ERRDESC(Command not recognized.)
ERRTEXT(EXPLANATION: You specified an unrecognized command in the command file.)
ERRTEXT(USER RESPONSE: Check the message response file, OUTMSG, )
ERRTEXT(profile response file, OUTPRO, or response work file, OUTWORK,
ERRTEXT(to determine which command produced the error.)
ERRTEXT(Correct the command file and retry the program.);
```



## Using checkpoint-level, file-level, and user-initiated recovery

This chapter explains in detail how to use the checkpoint-level, file-level, and user-initiated data recovery features of Information Exchange. It provides information on resuming or resetting an existing Information Exchange session.

When you use session-level recovery (described in the previous chapter), Expedite Base/400 must retransmit all data after an error. If you are sending a large amount of data, retransmission can take several hours.

When you use checkpoint-level, file-level, or user-initiated recovery, Expedite Base/400 can recover data more efficiently. The following table shows when Expedite Base/400 takes checkpoints for each of these recovery methods.

Checkpoint-level recovery	File-level recovery	User-initiated recovery
<ul style="list-style-type: none"> <li>While receiving data for a RECEIVE or RECEIVEEDI command, if the file was sent with checkpoints</li> </ul>	<ul style="list-style-type: none"> <li>While receiving data for a RECEIVE or RECEIVEEDI command, if the file was sent with checkpoints</li> </ul>	<ul style="list-style-type: none"> <li>While receiving data for a RECEIVE or RECEIVEEDI command, if the file was sent with checkpoints</li> </ul>
<ul style="list-style-type: none"> <li>At the end of each RECEIVE or RECEIVEEDI command</li> </ul>	<ul style="list-style-type: none"> <li>At the end of each RECEIVE or RECEIVEEDI command</li> </ul>	<ul style="list-style-type: none"> <li>At the end of each RECEIVE or RECEIVEEDI command</li> </ul>
<ul style="list-style-type: none"> <li>At the end of each SEND, SENDEDI, or PUTMEMBER command, if the next command is not a SEND, SENDEDI, or PUTMEMBER command</li> </ul>	<ul style="list-style-type: none"> <li>After each file sent as a result of a SEND, SENDEDI, or PUTMEMBER command</li> </ul>	<ul style="list-style-type: none"> <li>After each COMMIT command, unless there is nothing to commit</li> </ul>

Checkpoint-level recovery	File-level recovery	User-initiated recovery
<ul style="list-style-type: none"> <li>After sending the number of bytes you specify in the COMMITDATA parameter of the TRANSMIT command</li> </ul>	<ul style="list-style-type: none"> <li>After each file is received</li> </ul>	<ul style="list-style-type: none"> <li>At the end of each session, even if you have not specified a COMMIT command</li> </ul>

The processes for using checkpoint-level, file-level, and user-initiated recovery are very similar. Expedite Base/400 uses the same work files for these recovery methods. Considerations for restarting after an error and resetting the Expedite Base/400 session, described later in this chapter, also apply to all three recovery methods.



**NOTE:** SENDSTREAM and RECEIVESTREAM are only valid for session-level recovery.

## Selecting the appropriate recovery level

Session-level recovery is the default in Expedite Base/400. Request one of the other recovery methods by specifying one of the values below on the RECOVERY parameter of the TRANSMIT command:

- c      Checkpoint-level recovery
- f      File-level recovery
- u      User-initiated recovery

## Recovery considerations

Checkpoint-level, file-level, and user-initiated recovery are easy to use if you allow Expedite Base/400 to complete the current session successfully before processing the OUTMSG file or making any new requests.

**Attention:** Do not use the same account and user ID on multiple systems. If you start an Information Exchange session using checkpoint-level, file-level, or user-initiated recovery while another Information Exchange session with the same account and user ID is running, Information Exchange ends the first session and proceeds with the second session. The results in the first session depend on whether a checkpoint ended successfully:

- If a checkpoint did end successfully, Information Exchange delivers any data sent prior to the checkpoint and deletes any data from the mailbox that was received prior to the checkpoint.
- If a checkpoint did not end successfully, Information Exchange does not deliver any data and does not delete any received data from the mailbox. This means that data received in the first session may be received again in error.

In either case, you may get an error when you restart the first session. You should not run multiple sessions for the same account and user ID from different machines.

## Setting up work files

To take checkpoints, Expedite Base/400 must store the status of your Information Exchange session in a safe place. It must also close and reopen your message response file (OUTMSG) each time a checkpoint is completed. For these reasons, Expedite Base/400 requires that you create extra work files, SESSION, RCVWORK, and OUTWORK, when using one of these recovery methods.

When you use the SENDEDI command with VERIFY(C) or VERIFY(G) specified, you must also create an EDIWORK file. The following sections provide more details on how to create each work file.

### Creating the session file (SESSION)

Expedite Base/400 uses the session file to record the status of your Information Exchange session when it takes a checkpoint.

Expedite Base/400 does not create the session file. You must create it as a permanent file and refer to it with the name SESSION when you run Expedite Base/400. The session file must be at least 35000 bytes long.

You should not erase or modify the session file after an unsuccessful Expedite Base/400 session. If you erase or modify the session file, Expedite Base/400 reprocesses all of the requests in your INMSG file. If you have already received files, Expedite Base/400 overwrites them. If you have already sent files, Expedite Base/400 resends them.

To create the file, you can issue the following sample AS/400 command:

```
crtpf session rcdlen(1)
```

### Creating the receive work file (RCVWORK)

When you use checkpoint-level, file-level, or user-initiated recovery, Expedite Base/400 receives data into a work file. After it receives all the data, Expedite Base/400 moves it from the work file into the file specified with the FILEID parameter of the RECEIVE or RECEIVEEDI command. At this point, RCVWORK is ready to receive the next transmission.

Expedite Base/400 does not create the receive work file. You must create it as a permanent file and refer to it with the name RCVWORK when you run Expedite Base/400. Make the receive work file large enough to hold all of the data received by any single RECEIVE or RECEIVEEDI command in your message command file. For example, if you specify one RECEIVE command for which you expect to receive 5000 bytes of data and another RECEIVE command for which you expect to receive 5000000 bytes of data, you must make your receive work file large enough to hold 5000000 bytes of data. If you specify one RECEIVE command for which you expect to receive records of 255 bytes, and another for which you expect to receive records of 4,000 bytes, you must make your receive work file large enough to hold record lengths of 4,000 bytes.

To create the file, you can issue the following sample AS/400 command:

```
crtpf rcvwork rcdlen(4000)
```

### Creating the EDI work file (EDIWORK)

When you use the SENDEDI command with the VERIFY parameter equal to either *c* or *g*, Expedite Base/400 uses the EDI work file to track which EDI envelopes are sent and which are not.

Expedite Base/400 does not create the EDI work file. You must create it as a physical file and refer to it with the name `EDIWORK` when you run Expedite Base/400. The EDI work file must be able to hold at least as many characters as the maximum number of envelopes in the EDI file being sent.

You should not erase or modify the EDI work file after an unsuccessful Expedite Base/400 session. If you erase or modify the EDI work file, Expedite Base/400 might send duplicate data.

To create the file, you can issue the following sample AS/400 command:

```
crtpf ediwork rcdlen(255)
```

## Creating the output work file (OUTWORK)

When you use one of these recovery methods, Expedite Base/400 does not put echoed commands and response records directly into the message response file. First, it places echoed commands and response records into an output work file, then it copies them to the message response file when taking a checkpoint.

Expedite Base/400 does not create the output work file. You must create it as a permanent file and refer to it with the name `OUTWORK` when you run Expedite Base/400. Make this file the same size as your message response file (`OUTMSG`).

To create the file, you can issue the following sample AS/400 command:

```
crtpf outwork rcdlen(255)
```

## Processing the message response file

Processing the message response file (`OUTMSG`) for these recovery methods is very much like processing for session-level recovery; however, there are differences in the way you handle errors.



**NOTE:** When using checkpoint-, file-, or user-initiated recovery, Expedite Base/400 must open and close the message response file (`OUTMSG`) each time a checkpoint is completed.

## Examining response records

Response records are free-format records with the same syntax as commands. See “Understanding command syntax” on page 4 for a description of command syntax. Response records always start at the beginning of a line. However, parameters in response records can occur in any position in a record and in any order. In addition, all parameters in a response record might not appear. We suggest that you be prepared for the possible addition of new parameters to existing response records and of entirely new response records in the future.

When examining response records, consider the following:

- Assume a default value if you do not get the response record parameter you are expecting. This is not an error.
- If a response record parameter has a longer length than you expect, truncate the parameter.
- Be prepared to handle parameters that are split across records. Splitting can occur if the parameter is longer than the record length of your response file.

## Checking return codes

To ensure that Expedite Base/400 finished processing the message command file, check the return code in the SESSIONEND record.

The following table lists the return codes you can receive, explains them, and tells you what to do when you receive them:

If the SESSIONEND return code is:	Then:	You must do the following:
00000	Expedite Base/400 finished processing the command file. There were no processing problems, and all command RETURN records contain 0 return codes. All of your requests completed normally.	Nothing. Informational only.
28010	The Information Exchange session completed normally, but Expedite Base/400 could not complete all the commands in your message command file.	Check the command RETURN records to see which commands did not complete. Commands that completed normally have 0 RETURN records. Commands that did not complete normally have nonzero RETURN records, and if your system is set up to do so, can display error descriptions.
Any code other than 00000 or 28010	Expedite Base/400 did not finish processing the command file. Your Information Exchange session failed. However, some of the commands in your command file may have completed. Expedite Base/400 might have placed mail in your trading partner's mailbox or removed mail from your mailbox.	Correct the problem and restart Expedite Base/400. The SESSIONEND record might include an error description to help you find the problem, if your system is set up to display error descriptions. A command RETURN record might contain the same code and description. If there are no return records in OUTMSG, check OUTPRO and OUTWORK for the error.

If the error was because of a problem with a command, you can determine which command is wrong by looking in one of three places:

- OUTPRO - If the error was in a profile command, the command in error appears in OUTPRO. In this case, OUTWORK and OUTMSG do not contain any echoed commands.
- OUTWORK - If no checkpoint was taken while the command was processing, the command appears in OUTWORK. The command in error is followed by a RETURN record with the same return code as the SESSIONEND record. All syntax errors appear in OUTWORK.
- OUTMSG - If a checkpoint was taken while the command was processing, Expedite Base/400 displays the command in error in OUTMSG. The command is not followed by a RETURN record. However, Expedite Base/400 echoes the command in error as the last command in OUTMSG before SESSIONEND.

## Recovery response file examples

The SESSIONEND(00000) and SESSIONEND(28010) examples in “OUTMSG examples” on page 51 are valid for all data recovery methods. However, the following examples pertain only to check-point-level, file-level, and user-initiated recovery.

### Response file with a SESSIONEND (not 00000 or 28010)

#### Session-level recovery example

In the following example, you see the result of a typographical error. The programmer intends to enter **sendedi**, but types **sendedk** by mistake. This results in a 15040 error in the SESSIONEND record and another 15040 error in the RETURN record following “sendedk”. Because this is a syntax error, the error message is found in OUTWORK. Note that the SEND command preceding “sendedk” was completed. The presence of a SENT record in OUTMSG indicates that data was placed in the trading partner's Information Exchange mailbox, even though the Information Exchange session did not complete.

#### INMSG:

```
#Send a file named test to act1 user01 in user class tdata#
send fileid(library1/test(send1)) account(act1) userid(user01) class(tdata);

#Send an EDI file called edidata in user class editest#
SENDEDK FILEID(library1/edidata);
```

#### OUTMSG:

```
AUTOSTART SESSIONKEY(YWOHBNBV);
STARTED LASTSESS(0) RESPCODE(00000) SESSIONKEY (YWOHBNBV)
IEVERSION(04) IERELEASE(07);
RETURN(00000);

#Send a file named test to act1 user01 in user class tdata#

send fileid(test) account(act1) userid(user01) class(tdata);
SENT UNIQUEID(57586838) LENGTH(80);
RETURN(00000);
SESSIONEND(15040) ERRDESC(Command not recognized.)
ERRTEXT(EXPLANATION: You specified an unrecognized command in the command file.)
ERRTEXT(USER RESPONSE: Check the message response file, OUTMSG, profile)
ERRTEXT(response file, OUTPRO, or response work file, OUTWORK, )
ERRTEXT(to determine which command produced the error.)
ERRTEXT(Correct the appropriate command file and retry the program.);
```

#### OUTWORK :

```
#Send an EDI file called edidata in user class editest#

sendedk
RETURN(15040) ERRDESC(Command not recognized.)
ERRTEXT(EXPLANATION: You specified an unrecognized command in the command file.)
ERRTEXT(USER RESPONSE: Check the message response file, OUTMSG, profile)
ERRTEXT(response file, OUTPRO, or response work file, OUTWORK, )
```

ERRTEXT(to determine which command produced the error.)  
 ERRTEXT(Correct the appropriate command file and retry the program.);

#### User-initiated recovery example

This example illustrates a session using user-initiated recovery and the COMMIT command. The session does not end successfully because of a syntax error on the RECEIVE command. The RECEIVE command was entered as “RECVE.” The following example shows the command file, INMSG:

```
send fileid(library1/edidata) account(act1) userid(user01);
receive fileid(testdata);
sendedi fileid(edidata);
commit;
send fileid(test) account(act1) userid(user02);
recve fileid(testdata);
commit;
send fileid(test) account(act1) userid(user03);
```

The session ends with a 15040 error on the “RECVE” command and a 15040 error on the SESSIONEND record. Because this is a syntax error, the command in error is found in the OUTWORK file. Postprocessing of the response file shows which commands completed successfully. The following examples show the OUTMSG message response file and the OUTWORK response work file:

#### OUTMSG:

```
AUTOSTART SESSIONKEY(STUPU7DO);
STARTED LASTSESS(0) RESPCODE(00000) SESSIONKEY (STUPU7DO)
IEVERSION(04) IERELEASE(07);
RETURN(00000);

send fileid(library1/test(send1)) account(act1) userid(user01);
SENT UNIQUEID(STUPWAJN) LENGTH(2400);
RETURN(00000);

receive fileid(library1/test(send1));
RECEIVED ACCOUNT(*SYSTEM*) USERID(*ERRMSG*) CHARGE(6) LENGTH(257)
FILEID(test data) MSGDATE(040422) MSGDATELONG(20040422)
MSGTIME(192954) MSGSEQO(001687)
MSGNAME(05003) SESSIONKEY(STUPU7DO) DELIMITED(N) SYSNAME(EXPMV00T)
SYSLEVEL(460) TIMEZONE(L) DATATYPE(E) RECFM(???) RECLLEN(0)
RECDLM(N) UNIQUEID(00000000) SYSTYPE(01) SYSVER(0);
RETURN(00000);

sendedi fileid(edidata);
SENT UNIQUEID(STUPWCLX) LENGTH(764) ACCOUNT(act1) USERID(user01)
EDITYPE(X12) DESTINATION(act1 user02) QUALIFIER(ZZ)
CONTROLNUM(005500210) CLASS(#E2) MSGNAME(05500210)
MSGSEQNO(00001);
SENT UNIQUEID(STUPWCME) LENGTH(764) ACCOUNT(act1) USERID(user01)
EDITYPE(X12) DESTINATION(act1 user02) QUALIFIER(ZZ)
CONTROLNUM(005500210) CLASS(#E2)
MSGNAME(05500210) MSGSEQNO(00002);
SENT UNIQUEID(STUPWCNK) LENGTH(764) ACCOUNT(act1) USERID(user01)
```

```

EDITYPE(X12) DESTINATION(act1 user02) QUALIFIER(ZZ)
CONTROLNUM(005500210) CLASS(#E2) MSGNAME(05500210)
MSGSEQNO(00003);
RETURN(00000);

commit;
RETURN(00000);

SESSIONEND(15040) ERRDESC(Command not recognized.)
ERRTEXT(EXPLANATION: You specified an unrecognized command in the)
ERRTEXT(command file.)
ERRTEXT(USER RESPONSE: Check the message response file, OUTMSG,)
ERRTEXT(profile response file, OUTPRO, or response work file,)
ERRTEXT(OUTWORK, to determine which command produced the error.)
ERRTEXT(Correct the appropriate command file and retry the program.);

```

**OUTWORK:**

```

send fileid(test) account(act1) userid(user01);
SENT UNIQUEID(STUPWEX6) LENGTH(13746);
RETURN(00000);

recv
RETURN(15040) ERRDESC(Command not recognized.)
ERRTEXT(EXPLANATION: You specified an unrecognized command in the)
ERRTEXT(command file.)
ERRTEXT(USER RESPONSE: Check the message response file, OUTMSG,)
ERRTEXT(profile response file, OUTPRO, or response work file,)
ERRTEXT(OUTWORK, to determine which command produced the error.)
ERRTEXT(Correct the appropriate command file and retry the program.);

```

The RETURN(00000) records indicate that the first three commands before the COMMIT command completed successfully. The RETURN(00000) and SENT records for the second SEND command indicate that this command completed successfully. However, because the second COMMIT command was not processed because of the syntax error, Information Exchange will not deliver the file for the second SEND command.

The mistyped command (RECVE) should be corrected in the INMSG file and the session restarted. Expedite Base/400 will continue processing after the last successful COMMIT command, which in this example is after the second SEND command.

## Restarting after an error

Certain conditions can cause Expedite Base/400 to stop running without completing the Information Exchange session. Examples of these conditions include communication failures and syntax errors in your message command file. When one of these conditions occurs, correct the problem that caused the condition, then restart Expedite Base/400 by resubmitting your job. Expedite Base/400 then uses the information contained in its session file (SESSION) to resume the previous session at the last checkpoint. If the session file is not present, Expedite Base/400 cannot resume the session. Instead, it reprocesses all the requests in your message command file (INMSG).



**Attention:** Do not erase or alter the session file at any time. Damaging the session file during a restart situation can result in duplicated or lost data. Also, do not alter or erase the message response file during a restart situation. You can, however, alter the message command file to the extent necessary to correct any syntax or command error that caused the processing to stop. Never add or remove commands from the message command file between restarts.

## Changing files on restart

When you restart Expedite Base/400 from the last checkpoint, Expedite Base/400 uses the command and response files from the session that ended in an error. In addition, Expedite Base/400 uses several control files to keep track of its status.

**Attention:** Avoid changing any of these files when possible. Changes are sometimes required (for example, in the case of a syntax error), but be aware that improper changes can prevent Expedite Base/400 from restarting correctly and can cause the resulting data to be unreliable.

The following list shows which files you can change before restarting and which you cannot change. Feel free to change any file that is not on this list.

- **Session file (SESSION).** Never change this file before restarting. This is a control file used by Expedite Base/400 to restart the session.
- **Files being sent or received.** Do not modify files while sending or receiving them. Changes make the data that is sent or received unreliable.
- **Message command file (INMSG).** Do not change any commands or parameters in INMSG if they have already echoed to the message response file.

Changing INMSG commands or parameters that have already been processed and written to the message response file results in an error when you restart the program. You can change commands and parameters that have been written only to the output work file (OUTWORK) or that are not contained in either in the output work file (OUTWORK) or the message response file (OUTMSG).

- **Message response file (OUTMSG).** Do not change the message response file. Because it is a record of events that have already occurred, there is no reason to change it, and doing so can corrupt data.
- **Profile command file (INPRO).** Modify the profile command file if you want to change a profile value. However, do not change the RECOVERY, MSGSIZE, or COMMITDATA parameters in the TRANSMIT command. Expedite Base/400 processes the profile commands each time you restart.
- **Profile response file (OUTPRO).** Do not change the profile response file. Because it is a record of events that have already occurred, there is no reason to change it, and doing so can corrupt data. Expedite Base/400 creates a new profile response file when you restart.
- **EDI qualifier and destination tables.** Change only syntax errors in these files. If you change the destination used for a SENDEDI command while the command processes, the resulting information is unreliable.
- **EDI work file (EDIWORK).** Never change this file before restarting. This is a control file used by Expedite Base/400 to determine which EDI envelopes were sent and which were not.

- **Receive work file (RCVWORK).** Never change this file before restarting. This is a control file used by Expedite Base/400 to write data received during the processing of a RECEIVE or RECEIVEEDI command.

## Resetting the Expedite Base/400 session

Resetting your Expedite Base/400 session after an error prevents the program from restarting at the last checkpoint. Under normal conditions, you should never reset your session. If Expedite Base/400 returns an error other than 28010, resolve the error and restart the program to complete the session.

In the event of an emergency (for example, if the SESSION file becomes damaged, or if you decide you do not want Expedite Base/400 to complete the requests currently in the message command file), you must reset the session. Follow the steps below:

1. Determine what data has been transmitted.

Before you can reset your session, you must determine what data Expedite Base/400 has already sent or received. The best way to do this is to examine your Information Exchange *audit trail*. An audit trail provides a way of tracking and verifying basic information about the status of messages. You can examine your audit trail using Information Exchange Administration Services, the online product used by the Information Exchange Service Administrator to perform administrative tasks.

If you do not want to use the audit trail, you can use the SENT, NOTSENT, and RECEIVED records in the message response file to help determine which data has been sent and received. Normally, there is a SENT record for each file or EDI envelope sent and a RECEIVED record for each file or EDI envelope received. Expedite Base/400 produces a NOTSENT record for every EDI envelope that could not be sent because of a destination verification failure. NOTSENT records are produced only when you use the SENDEDI command with a value of c or g specified on the VERIFY parameter.

2. Remove the completed requests.

After you have determined what data has been sent and received and what commands have completed, remove the completed requests from your message command file.

This can be difficult when you send EDI data, because a single SENDEDI command can send many EDI envelopes from a single file. You have to remove the envelopes that have already been sent from your EDI data file.

If you do not remove completed SEND requests from your message command file, or if you do not remove the processed envelopes from your EDI file, Expedite Base/400 sends that data again when you restart it with the RESET command parameter. This causes your trading partner to receive two copies of the data. If you do not remove completed receive requests from your message command file, Expedite Base/400 destroys the data you have already received.

3. Reset the session.

After you have removed the completed send and receive requests from your message command file, restart Expedite Base/400 with the RESET parameter on the IEBASE call statement.

## Additional features of Expedite Base/400

---

You can use Expedite Base/400 to retrieve audit data, create distribution lists, transfer free-format messages, validate addresses, payment levels, and authorizations, or work with libraries. This chapter describes the use of these features.

### Using audit trails

Information Exchange provides an audit trail that you can retrieve using Expedite Base/400 or view using Information Exchange Administration Services (see *Using Information Exchange Administration Services*). Audit trails are files that contain information about files you send and receive. Audit trails can include the following information:

- Account and user ID of the person who received the file
- Account and user ID of the person who sent the file
- Date and time the file was sent or received
- User class of the file
- Current status of the file (for example, delivery date and time)

You can use the information in audit trails to see the status or final disposition of a file. For example, if you send an order electronically to a manufacturer on July 14 and do not receive the items as expected, you can request an audit trail of the files you sent to the manufacturer on July 14. The audit trail may show that the manufacturer never received the order and it is still in their mailbox. If the audit trail shows that they received the order, you may want to check with the manufacturer to find out if the order was ever filled.

You can also use audit trails to see how many files you sent or received over a specific time period. For example, you can request an audit trail that shows all of the files you sent over the last three weeks.

## Retrieving audit trails

Use the `AUDIT` command to retrieve an audit trail from Information Exchange and place the information in your mailbox. When audit data is available, use the `RECEIVE` command to retrieve it. The source account for the file is `*SYSTEM*`, the user ID is `*AUDITS*`, and the user class is `#SAUDIT`. Information Exchange does not prepare a common data header (CDH) to accompany retrieved audit records.

Audits are not available immediately; you cannot successfully issue the `RECEIVE` command after the `AUDIT` command to receive the audit data. Audits are normally available during the next session. For more information on the `AUDIT` and `RECEIVE` commands, see “`AUDIT` command” on page 87, and “`RECEIVE` command” on page 116.

The following steps show how to request and receive audit trails and what an audit trail looks like.

1. The following example shows the command file, `INMSG`, you would use to create an audit file with information about all files received between July 10 and July 20 with a user class of `ORDERS`.

```
audit startdate(040525) enddate(040526) class(orders);
```

The response file, `OUTMSG`, indicates a `RETURN(00000)` if the audit command was processed properly. The result would be a file placed in your mailbox with account `*SYSTEM*` and user ID `*AUDITS*`, with user class `#SAUDIT`.

2. The next step is to receive the audit file from your mailbox. The following example shows two commands, either of which could be used in the command file, `INMSG`, to receive the audit files.

```
receive fileid(audits/file(member)) class(#saudit);
receive fileid(audits/file(member)) account(*system*)userid(*audits*);
```

If there is an audit file in your Information Exchange mailbox, the response file, `OUTMSG`, will have a `RECEIVED` record indicating the file was received.

3. The following example shows the audit trail as it would appear in the file called `audits/file(member)`.



**NOTE:** When the audit trail is received from your mailbox, the data will be divided into records based on the record length of the file to which it is written. For this example, the `AUDIT` record has been formatted for display purposes.

```
ACCT SENDER 1 ACCT USERA A69438870554CAE3F51RECEIVED
SORDERS EXP/400 410 0000000200053473 040525123630 0000000
0405251437010405261237030405262123704
```

See the *Information Exchange Interface Programming Guide* for the format of an audit trail record.

## Creating distribution lists

You can send the same file to more than one person at a time by making a list of users and sending the file to that list. There are two types of distribution lists: permanent and temporary.

*Permanent distribution lists* are permanently stored in Information Exchange. The advantage of this type of list is that many people using different types of computers can use it. To create a permanent list, use the LIST command or Information Exchange Administration Services.

*Temporary distribution lists* last only for the duration of your Information Exchange session. When your Information Exchange session ends, Expedite Base/400 deletes your temporary list. To create a temporary list, use the LIST command. For a detailed description of the LIST command, see “LIST command” on page 101.

**Attention:** If you use the LIST command with session-level recovery, do not use the same distribution list name more than once in the same session. If you do, Information Exchange replaces the existing list definitions with the new list definitions. Therefore, any file you send uses the most recent list that you define.

## Free-format messages

Some Information Exchange interfaces support a free-format message facility. With this facility, you can type short messages on panels and send them to other users. These messages are made up of 79-byte fixed-length records and are sent in message class FFMSG001.

Expedite Base/400 does not provide free-format message panels. However, you can use an editor to create free-format messages and then send them using the send commands.

You can also receive free-format messages with the receive commands by using the FORMAT parameter.

When you use the FORMAT(y) parameter with a send command, Expedite Base/400 attempts to send the file as a free-format message. It pads records that are shorter than 79 bytes with blanks and splits records that are longer than 79 bytes into 79-byte records. It also sends the file in class FFMSG001 if you do not specify a class. When you use the FORMAT(y) parameter with the receive commands, Expedite Base/400 formats the file as a free-format message.

## Validating addresses, payment levels, and authorizations

To reduce transmission costs and ensure that your message can be delivered when communicating with unfamiliar trading partners, you may want to verify that:

- The address to which you want to send the message is a valid address.
- You can use a particular payment level when sending the message to the intended destinations.
- The intended destinations have appropriate authorization levels to receive messages from your Information Exchange address.

Expedite Base/400 can check for these items before sending a file, provided you are on the same system as the intended destinations. You can ask Expedite Base/400 to do this by using the VERIFY parameter of the SEND, SENDEDI, and SENDSTREAM commands. You are normally not charged to use the VERIFY parameter. However, in some circumstances you may incur a charge:

If the destination system is:	And the destination:	Then a charge is:
On the same system	Can be verified	Not incurred
On the same system	Is invalid	Incurred

If the destination system is:	And the destination:	Then a charge is:
Another Information Exchange system	Cannot be verified	Not incurred

You can use the VERIFY parameter for both distribution lists and individual destinations. However, when used to verify a distribution list, this parameter only verifies that the distribution list exists, not that the individual entries in the list are valid.

If a verification fails, Expedite Base/400 does not send the data. You can use the VERIFY parameter to specify whether or not Expedite Base/400 should send the data even if Information Exchange cannot verify the destination of a trading partner on another system.

To indicate how you, as the sender, want to pay the file charges, use the CHARGE parameter of the SEND, SENDEDI, and SENDSTREAM commands. You can specify one of six payment options. For more information, see *Information Exchange Charges Reference*.

See “SEND command” on page 129, “SENDEDI command” on page 135, and “SENDSTREAM command” on page 140 for more information on the VERIFY and CHARGE parameters.

## Using acknowledgments

You can request three types of acknowledgments by using the ACK parameter on any of the Expedite Base/400 send commands:

- receipt Information Exchange generates a receipt acknowledgment when a message reaches the receiver’s mailbox after a successful Expedite Base/400 session.
- delivery Information Exchange generates a delivery acknowledgment when a destination user receives a message from the Information Exchange mailbox.
- purge Information Exchange generates a purge acknowledgment when a message is purged from the receiver’s mailbox.

The following table shows the format of an acknowledgment:

Field #	Col	Field size	Field name	Description
1	1	5	MSGNUMBR	Message number
2	6	2	MSGSEVCD	Severity code
3	8	3	CMDLNTH	Length of command responsible
4	11	5	MSGDESLN	Length of message text
5	16	x	MSGRESPN	Command responsible
6	16+x	y	MSGDESCR	Message text

The x represents the length specified by the CMDLNTH field, and the y represents the length specified by the MSGDESLN field.

MSG NUMBR

The message number for acknowledgment type:

01000	Delivery acknowledgment
01001	Receipt acknowledgment
01002	Purge acknowledgment

#### MSGSEVCD

The severity code for acknowledgments, which is always 00.

#### CMDLNTH

The length of the MSGRESPN (command responsible) field.

#### MSGDESLEN

The length of the MSGDESCR (Information Exchange message text) field.

#### MSGRESPN

The command responsible for the generation of the message, which for all acknowledgments is the send message command. Expedite Base/400 generates the message in the Information Exchange format for the SEND, SENDSTREAM, SENDEDI, or PUTMEMBER command.

#### MSGDESCR

The message text generated by Information Exchange. For each type of acknowledgment, the text generated is:

receipt	<p><b>n</b> GROUP ACCEPTED FOR <b>y</b> <b>z</b>.</p> <p><b>n</b> is the UNIQUEID assigned by Expedite Base/400 on the SEND, SENDSTREAM, SENDEDI, or PUTMEMBER command. You can use this value to associate the acknowledgment with a file you sent.</p> <p><b>y</b> is replaced with the word LIST or USER, depending on whether the destination was a named distribution list or a single Information Exchange account and user ID.</p> <p><b>z</b> is the account and user ID, list name, or alias and alias name to which the file was delivered.</p>
delivery	<p><b>n</b> GROUP DELIVERED TO USER <b>z</b>.</p> <p><b>n</b> is the UNIQUEID assigned by Expedite Base/400 on the SEND, SENDSTREAM, SENDEDI, or PUTMEMBER command. You can use this value to associate the acknowledgment with a file you sent.</p> <p><b>z</b> is the account and user ID or the alias and alias name to which the file was delivered.</p>
purge	<p><b>n</b> GROUP PURGED FOR THE QUEUE OF <b>y</b>. Reason = <b>z</b>.</p> <p><b>n</b> is the UNIQUEID assigned by Expedite Base/400 on the SEND, SENDSTREAM, SENDEDI, or PUTMEMBER command. You can use this value to associate the acknowledgment with a file you sent.</p> <p><b>y</b> is the account and user ID or the alias and alias name of the user from whose mailbox the file was purged.</p> <p><b>z</b> is a simple explanation of why the message was purged.</p>

## Working with libraries

Expedite Base/400 provides four commands that allow you to work with Information Exchange libraries.

The GETMEMBER command allows you to copy a library member from an existing Information Exchange library to an Information Exchange mailbox. The PUTMEMBER command allows you to add a library member to an existing Information Exchange library. There is a charge for using the GETMEMBER and PUTMEMBER commands.

Expedite Base/400 also provides two commands, LISTLIBRARIES and LISTMEMBERS, which allow you to search for libraries and members to which you have access.

To use libraries in Expedite Base/400:

1. Create the library using Information Exchange Administration Services. If you do not have access to Information Exchange Administration Services, contact your marketing representative or service administrator.
2. Use the PUTMEMBER and GETMEMBER commands to put members into the library, retrieve them from the library, and place them into your mailbox.

For more information on these commands, see “GETMEMBER command” on page 98 and “PUTMEMBER command” on page 111.

3. Use the LISTLIBRARIES and LISTMEMBERS commands to identify libraries and members to which you have access.

For more information on these commands, see “LISTLIBRARIES command” on page 104 and “LISTMEMBERS command” on page 105.

## Using data compression

Compression reduces the size of the files that are transmitted through Information Exchange. Significant savings in network charges and transmission time are possible when using data compression. If you have installed the appropriate compression software, Expedite Base/400 will compress and decompress the data you send and receive.

To use data compression, you must specify a valid COMPRESS parameter on the SEND, SENDEDI, or SENDSTREAM commands. When you use data compression, some Expedite Base/400 parameters are impacted, and session restarts may have to be handled differently. For more information, see Appendix D, “Using data compression.”



**NOTE:** When using the COMPRESS parameter, both the sender and receiver of compressed data must have the licensed TDAcess product in order to compress and decompress the data.



## Expedite Base/400 commands

---

To use Expedite Base/400, you need to understand the basic command syntax and how to use the profile and message file commands. The following sections describe the API command syntax and provide you with details on the use of each command.

### Command syntax

An example of Expedite Base/400 command syntax is:

```
#Comment or description
command parameter(value) parameter(value) ...parameter(value);
```

The following are descriptions of the syntax elements:

Syntax Element	Description
#	Defines or delimits a comment line. You can type any information you like after a #, and Expedite Base/400 ignores the characters that follow it on the same line. If you include a # in a parameter value, Expedite Base/400 knows from the value's context that the # is part of a command and does not ignore the parameter value or the characters that follow it.
command	Identifies the Expedite Base/400 command.
parameter	Identifies a parameter of the associated command.
value	Defines the value associated with the parameter.
... or ::	Indicates that you can specify as many parameters as necessary. (It is not part of the syntax.)
;	Ends the command.

You can type Expedite Base/400 commands and parameters in uppercase or lowercase letters. The commands and parameters can span several lines in a command file. However, the following limitations apply:

- You must type the entire command name, for example, IDENTIFY, on a single input line.
- You must type the entire parameter name, for example, IEACCOUNT, on a single input line.
- A left parenthesis must immediately follow each parameter. Do not use spaces between parameter names and values. For example, type **inaccount(acct)** rather than **inaccount (acct)**.
- End each command with a semicolon (;).

## Command syntax examples

Each section describing a command in this chapter contains an example of the syntax for that command. The examples are formatted as follows:

- Parameter values are shown in *italics*.
- Required parameter values are shown in **boldface** type.
- Default parameter values are underlined.
- Some parameter values include *blank* as an option.

The word *blank* in a syntax example or description indicates a space containing no characters. It does not mean that you should enter the word *blank*.

- In some examples, you have a choice between parameters or groups of parameters. In cases like these, your choices are separated from each other by the word *or*.

Choose one line of parameters from any group of lines separated in this way. If there are required parameters in more than one of the choices, you still need to choose only one line but you must include the required parameters from that line.

## Profile commands

The profile commands give you a simple method for creating and maintaining your profile. You place these commands in the INPRO command file. Expedite Base/400 places return codes reflecting the completion of these commands in the OUTPRO response file.

The profile commands are listed here with the page numbers on which their detailed descriptions appear:

- IDENTIFY, page 74  
Gives you a way to set up the fields in the profile related to your account ID, user ID, and password information. With this command, you can also specify the time zone.
- SNACOMM, page 77  
For SNA communications, specifies the logical unit (LU) names for your own system and for the Information Exchange system to which you connect.
- SSL, PAGE 78  
For SSL communications, specifies that SSL communications be enabled.
- TCPCOMM, PAGE 79  
For TCP/IP communications, specifies the host and port for the Information Exchange system to which you connect.
- TRACE, page 80  
Specifies the information Expedite Base/400 should record in the trace file during the session. Possible traces include BASE, IOFILE, LINK, or PROTOCOL.
- TRANSMIT, page 82  
Specifies transmission control parameters.

## IDENTIFY command

The IDENTIFY command sets up the Information Exchange account, user ID, and password information, as well as the time zone. This command is required when you specify AUTOSTART(Y) in the TRANSMIT command. For more information, see “Profile command file (INPRO)” on page 12.

Three new parameters are added to the IDENTIFY command: KEYRINGFILE, KEYRINGPASSWORD, and APPLICID. These new parameters are only used if SSL is enabled on the new SSL command, and you specify AUTOSTART(Y). If you use the APPLICID parameter, the KEYRINGFILE and KEYRINGPASSWORD parameters are ignored. Using the KEYRINGFILE parameter allows the user to specify the name (and path) of the KDB file that contains the user’s public and private keys. Using the KEYRINGPASSWORD parameter allows the user to specify the KDB file password. Using the APPLICID allows the user to specify an application ID as an alternate tag into the default \*SYSTEM KDB file.

### Syntax

#### identify

```
ieaccount(IE account) ieuserid(IE user ID)
iepassword(IE password) iepassword(new IE password)
timezone(time zone) codepage(code page)

keyringfile(KDB filename) keyringpassword(key ring password)
OR
applicid(application ID) ;
```

### Example

```
identify ieaccount(account) ieuserid(userid) iepassword(mypswd) timezone(est)
keyringfile(keyring.kdb) keyringpassword(mykeypswd);
```

### Parameters

#### ieaccount

Information Exchange account. Expedite Base/400 uses this field together with the USERID parameter to identify the user. If you specify an ACCOUNT parameter, you must also specify a USERID parameter. Use 1 to 8 alphanumeric characters.

#### ieuserid

Information Exchange user ID. Expedite Base/400 uses this field together with the ACCOUNT parameter to identify the user. If you specify a USERID parameter, you must also specify an ACCOUNT parameter. Use 1 to 8 alphanumeric characters.

#### iepassword

Information Exchange password. Use 1 to 8 alphanumeric characters.

**niepassword**

New Information Exchange password. If you specify this value, the Information Exchange password changes upon completion of the Information Exchange session. If the Information Exchange session ends in an error, the password does not change. Use 1 to 8 alphanumeric characters.



**NOTE:** Remove this parameter and change the IEPASSWORD parameter after you change your password.

**timezone**

Your local time zone. You can enter one of the time zone codes listed here, or you can specify an offset from Greenwich mean time (GMT) by indicating the number of hours and minutes east or west of the Greenwich meridian. The format for specifying the hours and minutes is *ehhmm* or *whhmm*, where:

Type:	To indicate:
e	east
w	west
hh	hours
mm	minutes

For example, to specify eastern daylight time, you can enter either *edt* or *w0400*, where *w* indicates west, and *0400* indicates 4 hours and 0 minutes.

Time zone	GMT offset	Description
AHD	W0900	Alaska-Hawaii daylight
AHS	W1000	Alaska-Hawaii standard
AST	W0400	Atlantic standard time
BST	E0100	British summer time
CDT	W0500	Central daylight time
CST	W0600	Central standard time
EAD	E1000	Eastern Australia daylight time
EDT	W0400	Eastern daylight time
EMT	E0200	Eastern Mediterranean time
EST	W0500	Eastern standard time
GMT	E0000	Greenwich mean time
JST	E0900	Japanese standard time
MDT	W0600	Mountain daylight time
MST	W0700	Mountain standard time
PDT	W0700	Pacific daylight time
PST	W0800	Pacific standard time

Time zone	GMT offset	Description
WED	E0200	Western Europe daylight time
WES	E0100	Western Europe standard time
YDT	W0800	Alaska daylight time
YST	W0900	Alaska standard time

Use 1 to 5 alphanumeric characters. The default is **gmt**.

**codepage**

Code page you want to associate with the data you send. If you specify the code page, Expedite Base/400 includes the value in the CDH when you send data. Recipients can decide whether or not they need to provide special data translation facilities. Use 1 to 3 numeric characters.

**keyringfile**

The name (and path) of the kdb file that contains the certificate. This applies to TCP/IP communication performed with SSL enabled. The keyringpassword is required with this parameter. This parameter is case sensitive. If you use this parameter, do not use the applicid parameter.

**keyringpassword**

The kdb file password. This value is not case sensitive. It can be from 1 to 128 characters in length. This parameter is case sensitive. If you use this parameter, do not use applicid.

**applicid**

The name of the application that you defined in the Digital Certificate Manager (DCM) application database, which is tied to a specific certificate.

The valid values are whatever are valid when you defined the application id in DCM and it can be from 1 to 100 characters in length. If you use this parameter, do not use the keyringfile and keyringpassword parameters.

## SNACOMM command

Expedite Base/400 uses the SNACOMM command to indicate the names for your LUs and the Information Exchange system to which you connect. Your system programmer can provide you with this name. This command is required for using SNA communications. For more information, see “Profile command file (INPRO)” on page 12.

### Syntax

#### **snacomm**

```
ieluname(IE LU name) userluname(user LU name) ielumode(IE LU mode)  
ielunetid(IE LU network ID);
```

### Example

```
snacomm userluname(explunm1);
```

### Parameters

#### **ieluname**

LU name for the Information Exchange system. Use 1 to 8 alphanumeric characters. The default is **ibm0rely**, which is the system’s LU name in the United States.



**NOTE:** If you are in a country other than the United States, contact your GXS Sales representative for more information.

#### **userluname**

LU name for your application. If more than one copy of Expedite Base/400 is executed concurrently, you can repeat this parameter to define a group of LU names. Expedite Base/400 then selects the first unused LU name from the group. You can specify this parameter as many as 99 times. If you specify the SNACOMM command more than once, the USERLUNAMES on the last SNACOMM command override any previous USERLUNAMES specified. This value is known as the local location name on your APPC device description. Use 1 to 8 alphanumeric characters. This parameter is required.

#### **ielumode**

Mode name used to communicate with the Information Exchange system. Use 1 to 8 alphanumeric characters. The default is **LU62**.

#### **ielunetid**

Network ID for the Information Exchange system. Use 1 to 8 alphanumeric characters. The default is **IBMIN**, which is the network ID in the U.S.

## SSL command

Use the SSL command to specify that you want to communicate with Information Exchange using a secure TCP/IP connection. To use this command, you will need to request a connection using TCP/IP, point to the correct TCP/IP address, install a certificate obtained from the IGS Public Key Infrastructure, and set up your AS/400 environment to use SSL.

**NOTE:** For information about setting up your AS/400 environment, see the installation instructions.

### Syntax

```
ssl enablessl(n|y);
```

### Example

```
ssl enablessl(y);
```

### Parameters

#### **enablessl**

Indicates whether you want to enable SSL for this session for Expedite Base/400 to connect to Information Exchange. If you don't select a value, or select **n**, Expedite Base/400 will connect to the Information Exchange system, but SSL will not be enabled.

The valid values are **n** and **y**. The default is **n**.



## TCPCOMM command

Use the TCPCOMM command to specify the TCP/IP host and port to use when connecting to the worldwide network. You can enter up to five sets of host-port combinations. You correlate the members of a set by specifying a number from 1 to 5 at the end of the parameter. This command is required for TCP/IP communications.

The parameter values specified on this command are only used when the connection is established. Once the session has been established, Expedite Base/400 will not try connecting with another host-port combination if the session fails.

### Syntax

#### **tcpcomm**

**ietcphostn**(*IE TCP/IP hostn*) **ietcportn**(*IE TCP/IP port IDn*)

### Example

```
tcpcomm ietcphost1(infoexch.us.ibm.com) ietcport1(3001)
ietcphost2(32.76.15.6) ietcport2(3001);
```

**Results:** Expedite Base/400 will attempt to connect to host infoexch.us.ibm.com port 3001. If communication is unsuccessful, Expedite Base/400 will attempt to connect to IP address 32.76.15.6 port 3001.

### Parameters

#### **ietcphostn**

Indicates a host and domain name or the IP address Expedite Base/400 can try when connecting to Information Exchange. To specify a host and domain name, use up to 64 alphanumeric characters. To specify an IP address, use the format *nnn.nnn.nnn.nnn*, where each *nnn* represents a numeric value between 0 and 255. This is a required parameter.

The valid values for *n* are **1** to **5**.

#### **ietcportn**

Indicates the port ID Expedite Base/400 uses when connecting to Information Exchange. Use up to five numeric digits to specify a number from 1 to 65535. This is a required parameter.

The valid values for *n* are **1** to **5**.

## TRACE command

The TRACE command specifies the type of information recorded in the trace file during a session. Possible traces are PROTOCOL, LINK, IOFILE, and BASE. All traces are used by the Help Desk for problem determination.

If you request PROTOCOL, BASE, or IOFILE traces, you must create a file to hold the trace information and refer to it with the filename BASETRC. LINK trace is only used for TCP/IP connections. If you request a link trace, you must create a file to hold the link trace information and refer to it with the filename LINKTRC.

### Syntax

#### **trace**

```
protocol(n|y) link(n|y) base(n|y) iofile(n|y);
```

### Example

#### **trace**

```
protocol(y);
```

### Parameters

#### **protocol**

Indicates whether the trace file should contain Information Exchange protocol information.

- n** Do not include the Information Exchange protocol information in the trace file. This is the default.
- y** Include the Information Exchange protocol information in the trace file, showing the commands, data, and responses that were transferred to and from Information Exchange.

#### **link**

Indicates whether TCP/IP protocol link information should be traced and the data recorded in a separate file named LINKTRC.

- n** Do not include the link information in the trace file. This is the default.
- y** Include the link information in the trace file.

#### **base**

Indicates whether the trace file should contain the Expedite Base/400 module information.

- n** Do not include the Expedite Base/400 module information in the trace file. This is the default.
- y** Include the Expedite Base/400 module information in the trace file.

**iofile**

Indicates whether the trace file should show the parsing of the INPRO and INMSG files. This trace can be useful for finding the cause of syntax errors. Use this trace if you are having problems creating or modifying the INPRO or INMSG files.

- n Do not include the command parsing information in the trace file. This is the default.
- y Include the command parsing information in the trace file.

## TRANSMIT command

You can use the TRANSMIT command to specify a level of data recovery. You can also use this command to specify whether to start and end an Information Exchange session automatically. For information on session-level recovery, see Chapter 5, “Using session-level recovery.” For more information on alternative data recovery methods, see Chapter 6, “Using checkpoint-level, file-level, and user-initiated recovery.”

### Syntax

#### transmit

```
autostart(y|n) autoend(y|n)
msgsize(message size) commitdata(commit data)
maxmsgs(maximum messages) recovery(s|c|f|u) timeout(timeout)
commtype(s|t);
```

### Example

```
transmit recovery(c);
```

### Parameters

#### autostart

Indicates whether Expedite Base/400 should start an Information Exchange session automatically using the account ID and user ID in the IDENTIFY command. Expedite Base/400 automatically starts a session only when it begins processing INMSG. If you want to start multiple Information Exchange sessions in INMSG, you must specify both AUTOSTART(n) and AUTOEND(n).

- n Expedite Base/400 does not automatically start the Information Exchange session. You must include the START command in the INMSG file.
- y Expedite Base/400 automatically starts the Information Exchange session. Do not include the START command in the INMSG file when you transmit data. This is the default.

#### autoend

Indicates whether Expedite Base/400 sends an Information Exchange session end command when it finishes processing your command file. Expedite Base/400 automatically ends a session only when it finishes processing INMSG. If you want to start multiple Information Exchange sessions in INMSG, you must specify both AUTOSTART(n) and AUTOEND(n).

- n Expedite Base/400 does not automatically end the Information Exchange session. You must include the END command in the INMSG file.
- y Expedite Base/400 automatically ends the Information Exchange session. Do not include the END command in the INMSG file when you transmit data. This is the default.

#### msgsize

Segments the data for sending. Your trading partner can take checkpoints only for the message size you specify with this parameter. If you use a large value, your trading partner cannot take frequent checkpoints. Lower values permit the receiving interface to complete more frequent checkpoints while receiving the data.

Valid values are 1000 through 99999. The default value is **47000** bytes.



**NOTE:** You should use the default value when communicating with a slower receiver, such as a PC. You must use a value that is less than or equal to the value used in the COMMITDATA parameter.

#### **commitdata**

Indicates the maximum amount of data that Expedite Base/400 sends between checkpoints. For maximum efficiency, the value for the COMMITDATA parameter should be an even multiple of the MSGSIZE value. Lower values lessen the amount of data to be retransmitted if there is a communication failure. Higher values provide faster data transmission.

This parameter applies only to checkpoint-level recovery. It is ignored for session-level, file-level, or user-initiated recovery.

Valid values are 1000 through 9999999 and must be at least as large as MSGSIZE. The default value is **141000**.



**NOTE:** You must use a value that is greater than or equal to the value used in the MSGSIZE parameter.

#### **maxmsgs**

Maximum number of message segments Expedite Base/400 can receive between Information Exchange commits. The valid values are 1 to 9999. The larger the number specified, the more data Information Exchange sends to you without committing it. A lower number causes more frequent data commits. The default value is **10**. Unless you experience frequent losses of connection, do not change this parameter.

#### **recovery**

Indicates what type of data recovery will be used.

- s Session-level recovery. This is the default. See Chapter 5, “Using session-level recovery,” for more information.
- c Checkpoint-level recovery. See Chapter 6, “Using checkpoint-level, file-level, and user-initiated recovery,” for more information.
- f File-level recovery. See Chapter 6, “Using checkpoint-level, file-level, and user-initiated recovery,” for more information.
- u User-initiated recovery. See “COMMIT command” on page 93 and Chapter 6, “Using checkpoint-level, file-level, and user-initiated recovery,” for more information.

#### **timeout**

Number of minutes Expedite Base/400 waits for a response from Information Exchange before generating an error. Valid values are 0 to 99. The default value is **2**.

**Note:** If you specify 0, the maximum value is used.

#### **commtype**

Specifies the type of communication to be used.

- s Use SNA LU 6.2 to communicate with Information Exchange. This is the default.
- t Use TCP/IP to communicate with Information Exchange.

## Message commands

Use the message commands to send and receive files, control your Information Exchange mailbox, and define lists of users that you can use when sending and receiving files. Enter message commands to the Expedite Base/400's message command file, INMSG. The message commands are listed here with the page numbers on which their detailed descriptions appear:

- ARCHIVEMOVE, page 86  
Moves messages from the Information Exchange short-term archive to your mailbox. Expedite Base/400 places a MOVED record in the response file as a result of the ARCHIVEMOVE command.
- AUDIT, page 87  
Loads an audit in your mailbox.
- CANCEL, page 90  
Cancels previously sent files if the receiver has not retrieved them from the mailbox.
- COMMIT, page 93  
Sends a commit to Information Exchange and processes the response.
- DEFINEALIAS, page 94  
Defines a new alias, redefines an existing alias, or changes or deletes an existing alias table.
- END, page 97  
Ends an Information Exchange session.
- GETMEMBER, page 98  
Copies a library member from an existing Information Exchange library to an Information Exchange mailbox.
- LIST, page 101  
Sets up a list of account and user IDs that you can use to send and receive files.
- LISTLIBRARIES, page 104  
Returns a list of Information Exchange libraries to which you have access.
- LISTMEMBERS, page 105  
Returns a list of members within an Information Exchange library.
- LISTVERIFY, page 106  
Gives you information about lists.

- **MSGINFO**, page 109  
Gives you information about messages currently available in your mailbox.
- **PURGE**, page 110  
Deletes a specific file from your mailbox.
- **PUTMEMBER**, page 111  
Adds a member to an existing Information Exchange library.
- **QUERY**, page 115  
Gets information about all the messages in your mailbox. You can also use **QUERY** to see the CDH information associated with each message, if desired.
- **RECEIVE**, page 116  
Receives all files or specific files, including free-format messages.
- **RECEIVEEDI**, page 122  
Receives EDI-formatted files. Expedite Base/400 supports X12, UCS, EDIFACT, and UN/TDI standards.
- **RECEIVESTREAM**, page 126  
Receives data directly into the message response file (OUTMSG). When Expedite Base/400 receives data using the **RECEIVESTREAM** command, it places the data immediately after the **RECEIVED** record in the OUTMSG file.
- **SEND**, page 129  
Sends files, including free-format messages that you saved into a file, to a user or a list of users.
- **SENDEDI**, page 135  
Sends EDI-formatted files. Expedite Base/400 supports X12, UCS, EDIFACT, and UN/TDI standards.
- **SENDSTREAM**, page 140  
Sends data from the message command file (INMSG). Place the data immediately after the **SENDSTREAM** command in the command file.
- **SESSIONINFO**, page 144  
Gives you information about your Information Exchange session.
- **START**, page 145  
Starts an Information Exchange session.
- **TESTMSG**, page 147  
Tells Information Exchange to load one or more test messages into the mailbox.

## ARCHIVEMOVE command

Use the ARCHIVEMOVE command to move messages from the Information Exchange short-term archive to your mailbox. Expedite Base/400 places a MOVED record in the response file as a result of the ARCHIVEMOVE command.

### Syntax

**archivemove**

**archiveid**(*archive ID*);

### Example

```
archivemove archiveid(myrefid);
```

### Parameters

#### **archiveid**

Archive reference identifier for the messages you want to move from the archive to your mailbox. This is the value specified in the ARCHIVEID parameter of the RECEIVE, RECEIVEEDI, or RECEIVESTREAM command when the message was received. If you did not specify an ARCHIVEID when the message was received, the archive reference identifier assigned by Information Exchange is placed in the SESSIONKEY parameter of the RECEIVED record. Use 1 to 8 alphanumeric characters. This parameter is required.



## AUDIT command

Information Exchange provides an audit trail that you can retrieve. Use the AUDIT command to tell Information Exchange to load audit information into your mailbox. When an audit becomes available, you must issue a RECEIVE command to retrieve it. The audit information comes from the \*SYSTEM\* account ID and the \*AUDITS\* user ID in message user class #SAUDIT.

For details on audit information, see *Using Information Exchange Administration Services*. A sample audit format program is included on the Expedite Base/400 program tape. A sample audit report, which the audit program generates, is included in Appendix C, “Sample audit report.”



**NOTE:** Information Exchange does not make audits available immediately. Therefore, you cannot successfully issue the RECEIVE command immediately after issuing the AUDIT command. Audits become available during a subsequent session.

For more information, see the *Information Exchange Interface Programming Guide*.

### Syntax

#### audit

account(*account*) userid(*user ID*)

or

sysid(*system ID*) account(*account*) userid(*user ID*)

or

alias(*alias*) aliasname(*alias name*)

altacct(*alternate account*) altuserid(*alternate user ID*) msgtype(*s|r|b*) class(*class*)

startdate(*starting date*) enddate(*ending date*) status(*blank|u|p|d*)

timezone(*l|g*) level(*1|2|3*);

### Example

```
audit account(acct) userid(user01) class(#e2) level(2);
```

**Results:** This command places a level 2 audit report in your mailbox. This audit will include information about all files with a class of #e2 sent to and received from account *acct* and user ID *user01*, or any files that user ID might have purged.

### Parameters

#### account

Account name of an Information Exchange trading partner. Expedite Base/400 uses this field together with the USERID parameter to indicate that you want only audit records for message exchanges with this account and user ID. If you specify an ACCOUNT parameter, you must also specify a USERID parameter. Use 1 to 8 alphanumeric characters.

#### userid

User ID of an Information Exchange trading partner. Expedite Base/400 uses this parameter together with the ACCOUNT parameter to indicate that you want only audit records for message exchanges with this user ID and account. If you specify a USERID parameter, you must also specify an ACCOUNT parameter. Use 1 to 8 alphanumeric characters.

**sysid**

System ID of an Information Exchange trading partner. You need the system ID only if you specify the ACCOUNT and USERID parameters for a trading partner on another Information Exchange system. If you specify a SYSID parameter, you must also specify the ACCOUNT and USERID parameters. Use 1 to 3 alphanumeric characters.

**alias**

Table type and table name of an alias table. Use this parameter with the alias name to indicate that you want only audit records for messages exchanged with a single trading partner.

blank	You did not use an alias name. This is the default.
gxxx	Global alias table, where xxx identifies a 1- to 3-character table name.
oxxx	Organizational alias table, where xxx identifies a 1- to 3-character table name.
pxxx	Private alias table, where xxx identifies a 1- to 3-character table name.



**NOTE:** You create and maintain alias tables by using Information Exchange Administration Services (see *Using Information Exchange Administration Services*), or by using the DEFINEALIAS command (see “DEFINEALIAS command” on page 94).

**aliasname**

Alias name defined in the alias table. Use this parameter to indicate that you want audit records only for messages exchanged with a particular trading partner. Use 1 to 16 alphanumeric characters.

**altacct**

Alternate account for which you want to receive an audit. Use this field to request audit records for an account/user ID other than your own. If left blank, audit records for your own account will be returned. If you specify ALTACCT, you must specify ALTUSERID. You must have authority to view audit records for the alternate account or user ID. Use 1 to 8 characters.



**NOTE:** ALTACCT is only valid for level 3 audit requests.

**altuserid**

Specifies that you want to retrieve audit records for another user or users in your account. To retrieve audit records for another user in your account, put the user's ID here. To retrieve audit records for all user IDs in your account, use ? as the parameter value.

If you want to retrieve audit records for several specific users, you must specify each user ID in a separate AUDIT command. Use 1 to 8 alphanumeric characters.



**NOTE:** Information Exchange puts an error message in your mailbox and does not use this field if you do not have service administrator authorization.

**msgtype**

Type of audit records to be retrieved.

- s Retrieve all audit records for files you sent.
- r Retrieve all audit records for files sent to you.
- b Retrieve all audit records for files you have sent, and those that have been sent to you. This is the default.

**class**

Indicates that you want to retrieve only those audit records for files with the user classification given here. Use 1 to 8 alphanumeric characters.

**startdate**

Starting date of a time range for the messages and files about which you want audit information. The format is *YYMMDD* or *YYYYMMDD*. The default value is determined by Information Exchange, and is currently **000102** (January 2, 1900).

**enddate**

Ending date of a time range for the messages and files about which you want audit information. The format is *YYMMDD* or *YYYYMMDD*. The default value is determined by Information Exchange, and is currently **420916** (September 16, 2042).

**status**

Specifies that you want only audit records for messages with a particular status.

- blank Retrieve audit records for all files. This is the default.
- u Retrieve audit records only for files that have not been delivered.
- p Retrieve audit records for purged files only.
- d Retrieve audit records for delivered files only.

**timezone**

Date and time point of reference used by the *STARTDATE* and *ENDDATE* parameters.

- g Greenwich mean time.
- l Your local time, as specified by the *TIMEZONE* parameter of the *IDENTIFY* command. This is the default.

**level**

Style of an audit report.

- 1 Retrieve basic information. This is the default.
- 2 Retrieve basic information plus enhanced time fields.
- 3 Retrieve level 2 audit information plus the EDI exchange control number, for one or more users in another account.

## CANCEL command

Use the CANCEL command to cancel files that you previously sent by Information Exchange to a single account and user ID, a list of users, or a destination specified by an alias name. You can cancel these files only if the receiver has not retrieved them from Information Exchange. Only the sender of a file can request to cancel the file from the receiver's mailbox.

### Syntax

#### cancel

**alias**(*alias*) **aliasname**(*alias name*)  
 or  
**account**(*account*) **userid**(*user ID*)  
 or  
**listname**(*list name*)  
 priority(**blank**|*p*) msgname(*message name*)  
 msgseqno(*message sequence number*) class(*class*)  
 timezone(**l**|**g**) ack(**blank**|*h*|*t*)  
 startdate(*starting date*) starttime(*starting time*)  
 enddate(*ending date*) endtime(*ending time*);

### Example

```
cancel account(acct) userid(user01) class(special);
```

**Results:** This command removes any files in the mailbox for account *acct* and user ID *user01* that you have sent with a class of *special* and that have not yet been received.

### Parameters

#### alias

Alias table type and table name.

blank	No alias table is being used. This is the default.
gxxx	Global alias table, where xxx identifies a 1- to 3-character table name.
oxxx	Organizational alias table, where xxx identifies a 1- to 3-character table name.
pxxx	Private alias table, where xxx identifies a 1- to 3-character table name.



**NOTE:** You create and maintain alias tables by using Information Exchange Administration Services (see *Using Information Exchange Administration Services*), or by using the DEFINEALIAS command (see “DEFINEALIAS command” on page 94).

#### aliasname

Alias name defined in the alias table. Use 1 to 16 alphanumeric characters.

#### account

Account of a single-destination Information Exchange user with whom you are exchanging data. Expedite Base/400 uses this field together with the USERID parameter to identify the user. If you specify an ACCOUNT parameter, you must also specify a USERID parameter. Use 1 to 8 alphanumeric characters.

**userid**

User ID of an Information Exchange user to whom you are sending data. Expedite Base/400 uses this field together with the ACCOUNT parameter to identify the user. If you specify a USERID parameter, you must also specify an ACCOUNT parameter. Use 1 to 8 alphanumeric characters.

**listname**

Name of a previously defined list of accounts and user IDs. Use 1 to 8 alphanumeric characters.

**priority**

Limits the files canceled to those with the priority specified here.

blank      Normal priority. This is the default.

p            High priority.

**msgname**

Limits the files canceled to those with the message name specified in this parameter. Use 1 to 8 alphanumeric characters.

**msgseqno**

Limits the files canceled to those with the message sequence number specified in this parameter. Use 1 to 5 alphanumeric characters.

**class**

Limits the messages canceled to those with the user classification specified in this parameter. Use 1 to 8 alphanumeric characters.

**timezone**

Date and time reference in the STARTDATE, STARTTIME, ENDDATE, and ENDTIME parameters.

g            Greenwich mean time.

l            Your local time, as specified on the TIMEZONE parameter of the IDENTIFY command. This is the default.

**ack**

Types of acknowledgments you want to receive from Information Exchange regarding the cancellation.

blank      No acknowledgments are created regarding the cancellation of a message group. This is the default.

h            Acknowledgments include only header information.

t            Acknowledgments include both header and text information.

**startdate**

Starting date of a time range for the files sent to Information Exchange that you want to cancel. For a file to qualify for cancellation, the submit date of the beginning of the file must fall within this date range. The format is YYMMDD or YYYYMMDD. The default value is determined by Information Exchange, and is currently **000102** (January 2, 1900).

**starttime**

Starting time of a time range for the files sent to Information Exchange that you want to cancel. For a file to qualify for cancellation, the submit time of the beginning of the file must fall within this time range. Use the *HHMMSS* format for the start time. The default value is determined by Information Exchange, and is currently **000000** (0 hours).

**enddate**

Ending date of a time range for the files sent to Information Exchange that you want to cancel. For a file to qualify for cancellation, the submit date of the beginning of the file must fall within this date range. The format is *YYMMDD* or *YYYYMMDD*. The default value is determined by Information Exchange, and is currently **420916** (September 16, 2042).

**endtime**

Ending time of a time range for files sent to Information Exchange that you want to cancel. For a file to qualify for cancellation, the submit time of the beginning of the file must fall within this time range. Use the *HHMMSS* format for the end time. The default value is determined by Information Exchange, and is currently **235347** (23:53:47).

## COMMIT command

Use the COMMIT command to send a commit to Information Exchange and process the response. In order for Expedite Base/400 to perform the commit, a SEND, SENDEDI, or PUTMEMBER command must be requested between COMMIT commands or between a session start and a COMMIT command.

The message response record associated with the COMMIT command is the RETURN record.

Checkpoints are taken automatically:

- After each COMMIT command, unless there is nothing to commit
- After the end of each session, even if you have not specified a COMMIT command
- While receiving data for a RECEIVE or RECEIVEEDI command, if Information Exchange requests a checkpoint
- At the end of each RECEIVE or RECEIVEEDI command

**The COMMIT command is valid only for user-initiated recovery.** The TRANSMIT profile command determines the type of recovery. To specify user-initiated recovery, set the RECOVERY parameter of TRANSMIT to *u*.

On restart, data transmission resumes after the last successful checkpoint or COMMIT command. If the error occurred while processing the first COMMIT command in the message command file and a checkpoint has not occurred, Information Exchange does not deliver any data and does not delete received data from the mailbox. In this case, Expedite Base/400 retransmits all data upon restart.

The COMMIT command has no parameters.

### Syntax

**commit;**

## DEFINEALIAS command

Use the DEFINEALIAS command to:

- Define a new alias
- Redefine an existing alias
- Change or delete an existing alias table



**NOTE:** Your user ID must be authorized to update the alias table, or you will receive a system error message in your mailbox.

Although you can generally specify parameters in any order, the DEFINEALIAS command entries must include an ALIASTABLE parameter and one of the following groups of parameters:

- ACCOUNT and USERID
- SYSID, ACCOUNT, and USERID
- ALIAS and ALIASNAME

You must pair these entries correctly. You must specify a DEFINENAME parameter with the ACCOUNT and USERID parameters or the SYSID, ACCOUNT, and USERID parameters. Another option is to specify the DEFINENAME parameter with the ALIAS and ALIASNAME parameters.

You may not specify another DEFINENAME parameter unless the definition for the previous DEFINENAME has been specified.

If the value of the FUNCTION parameter is *e*, you do not need to specify a DEFINENAME parameter.



**NOTE:** Do not define an alias more than once in a session, or the first alias will be overwritten.

### Syntax

#### **definealias**

```

aliastable(define alias table) function(a|n|c|d|e)
authority(p|a|g)
definename(define aliasname 1)
account(account 1) userid(user ID 1)
      or
sysid(system ID 1) account(account 1) userid(user ID 1)
      or
alias(alias 1) aliasname(aliasname 1)
      :
      :
definename(define aliasname n)
account(account n) userid(userid n)
      or
sysid(system ID n) account(account n) userid(user ID n)
      or
alias(alias n) aliasname(aliasname n);

```



### Example

The following example shows valid DEFINEALIAS entries:

#### definealias

```

aliasable(GX01) function(A) authority(G)
definename(DUNS01) account(ACT1) userid(USER1)
definename(DUNS02) sysid(EUR) account(ACT2) userid(USER2)
definename(DUNS03) alias(GX02) aliasname(DUNS51);

```

The following example shows invalid DEFINEALIAS entries:

#### definealias

```

aliasable(GX01) function(A) authority(G)
definename(DUNS01) account(ACT1)                               [INVALID entry: no userid]
definename(DUNS02) account(ACT2) userid(USER2)
userid(USER1)                                                  [INVALID entry: 2 userids]
definename(DUNS03) alias(GX02) aliasname(DUN51);

```

### Parameters

#### aliasable

Alias table type and table name.

- `gxxx` Global alias table, where `xxx` identifies a 1- to 3-character table name.
- `oxxx` Organizational alias table, where `xxx` identifies a 1- to 3-character table name.
- `pxxx` Private alias table, where `xxx` identifies a 1- to 3-character table name.

#### function

Type of operation you want Expedite Base/400 to request for the alias table.

- `a` Add the following entries into the alias table. This is the default.
- `c` Change the following entries in the alias table.
- `d` Delete the following entries from the alias table. If Information Exchange deletes all the names from the alias table, you get the same results as with option `e`.
- `e` Erase the entire alias table.
- `n` Create a new alias table using the name specified in the ALIASTABLE parameter.

#### authority

Authorization level of the alias table being referenced. Authority is valid only if FUNCTION is `n`. You cannot change the authority of an existing table.

- `p` The alias table can be updated only by the owner of the table. This is the default.
- `a` The alias table can be updated by any administrator in the account.
- `g` The alias table can be updated by any user.

#### definename

Alias name you wish to define in the table specified by ALIASTABLE. Use 1 to 16 alphanumeric characters.

**sysid**

System ID of an Information Exchange user. You need the system ID only if you specify the ACCOUNT and USERID parameters for a user on another Information Exchange system. You do not need to specify the system ID if you and your trading partner are on the same Information Exchange system. Use 1 to 3 alphanumeric characters.

**account**

Account of an Information Exchange user to whom you are sending data. Expedite Base/400 uses this field together with the USERID parameter to identify the user. If you specify an ACCOUNT parameter, you must also specify a USERID parameter. Use 1 to 8 alphanumeric characters.

**userid**

User ID of an Information Exchange user to whom you are sending data. Expedite Base/400 uses this field together with the ACCOUNT parameter to identify the user. If you specify a USERID parameter, you must also specify an ACCOUNT parameter. Use 1 to 8 alphanumeric characters.

**alias**

Table type and name of an alias that points to an alias name in another table.

blank	An alias name was not used. This is the default.
gxxx	Global alias table, where xxx identifies a 1- to 3-character table name.
oxxx	Organizational alias table, where xxx identifies a 1- to 3-character table name.
pxxx	Private alias table, where xxx identifies a 1- to 3-character table name.



**NOTE:** You create and maintain alias tables by using Information Exchange Administration Services (see *Using Information Exchange Administration Services*), or by using the DEFINEALIAS command (see “DEFINEALIAS command” on page 94).

**aliasname**

Alias name in the alias table specified in the ALIAS parameter. Use 1 to 16 alphanumeric characters. This parameter is required if you specified an alias name in the ALIAS parameter.

## END command

Use the END command to end the Information Exchange session. The END command has no parameters.

### Syntax

**end;**



**NOTE:** Specifying an END command in the command file results in an error if you use **y** as the AUTOEND value in the profile.

## GETMEMBER command

Use the GETMEMBER command to copy a member from an existing Information Exchange library to an Information Exchange mailbox. When the member is available in the mailbox, specify a RECEIVE command as the next command in the input file to receive this member. You should use the RECEIVE or RECEIVESTREAM command as usual to receive the data from the mailbox to your system. If the destination is left blank on the GETMEMBER command, the default will be your own mailbox. The MSGNAME, MSGSEQNO, and CLASS parameters, if not specified, will default to values indicated when the member was stored in the library.

The library member requested may not be immediately available in your Information Exchange mailbox. If you do not receive the member and the GETMEMBER command (a RETURN(00000) code) completed successfully, try to receive the member in a subsequent session.

### Syntax

#### getmember

```
owner(library owner account)
library(library name)
member(member name)
account(account) userid (user ID)
    or
sysid(system ID) account(account) userid(user ID)
    or
alias(alias) aliasname(alias name)
    or
listname(list name)
msgname(message name) msgseqno(message sequence no)
class(class) charge(1|3|5|6)
ack(blank|a|b|c|d|e|f|r) retain(retain time);
```

### Example

```
getmember owner(abcsales) library(sales1) member(1q04) fileid(myfiles/prices);
```

**Results:** The member *prices* is retrieved from *1q04* in *sales1* under the owner *abcsales*. If more than one such file exists in your mailbox, it is appended to *prices*.

### Parameters

#### owner

Owner of the account of the requested library. The default is your own account. Use 1 to 8 alphanumeric characters.

#### library

Library from which the member is to be copied. Use 1 to 8 alphanumeric characters.

#### member

Library member to be copied. Use 1 to 8 alphanumeric characters.

#### sysid

System ID of an Information Exchange system user receiving the member. You need the system ID only if you specify the ACCOUNT and USERID parameters for a user on another Information Exchange system. If you specify a SYSID parameter, you must specify the ACCOUNT and USERID parameters. Use 1 to 3 alphanumeric characters.

**account**

Account name of the Information Exchange user receiving the file. Expedite Base/400 uses this field together with the USERID parameter to identify the user. If you specify an ACCOUNT parameter, you must specify a USERID parameter. Use 1 to 8 alphanumeric characters. The default is your own account.

**userid**

User ID of the Information Exchange user receiving the member. Expedite Base/400 uses this field together with the ACCOUNT parameter to identify the user. If you specify a USERID parameter, you must specify an ACCOUNT parameter. Use 1 to 8 alphanumeric characters. The default is your own user ID.

**alias**

Table type and table name of an alias table. This field is used in conjunction with the ALIASNAME parameter to identify the user receiving the file.

blank	An alias name was not used. This is the default.
gxxx	Global alias table, where xxx identifies a 1- to 3-character table.
oxxx	Organizational alias table, where xxx identifies a 1- to 3-character table name.
pxxx	Private alias table, where xxx identifies a 1- to 3-character table name.



**NOTE:** You create and maintain alias tables by using Information Exchange Administration Services (see *Using Information Exchange Administration Services*), or by using the DEFINEALIAS command (see “DEFINEALIAS command” on page 94).

**aliasname**

Alias name defined in the alias table. This field is used in conjunction with the ALIAS parameter to identify the user receiving the file. Use 1 to 16 alphanumeric characters.

**listname**

Name of a previously defined list of accounts and user IDs. This field is used to identify the list of users receiving the file. Use 1 to 8 alphanumeric characters.

**msgname**

Name you specify for the file. Use 1 to 8 alphanumeric characters.

**msgseqno**

Number assigned by you as a message control number for the data. Use 1 to 5 alphanumeric characters.

**class**

Specific name for the data. A receiver can use this name to receive only files of this class. The default is **blank**.

**charge**

Specifies how the sender wants the file charges paid.

1	The receiver pays all charges.
3	The receiver pays all charges, if agreed to by the receiver. If not, the sender and receiver split the charges, if agreed to by the receiver. Otherwise, the library owner pays all charges. This is the default.

- 5 The library owner pays the receive charge, if agreed to by the library owner. Otherwise, the receiver pays.
- 6 The library owner pays the receive charge.

**ack**

Type of acknowledgments you want to receive from Information Exchange.

- blank No acknowledgments created. This is the default.
- a Purge acknowledgments only.
- b Both receipt and delivery acknowledgments.
- c Both receipt and purge acknowledgments.
- d Delivery acknowledgments only. Acknowledgments are queued to the user ID when Information Exchange delivers the data to a destination user ID and reaches a subsequent recovery point with the same destination user ID.
- e Either purge or delivery acknowledgments.
- f Receipt acknowledgments and either purge or delivery acknowledgments.
- r Receipt acknowledgments only when Expedite Base/400 finishes sending a file.



**NOTE:** Expedite Base/400 sends the receipt acknowledgment to whomever is paying for the GETMEMBER request. This is either the library owner or the individual who requested the member (issued the GETMEMBER command).

**retain**

Number of days Information Exchange keeps the file in the mailbox if no one receives it. Valid values are **blank** and **0** through **180**.

The maximum retention period and the default retention period can be different on different Information Exchange systems. These periods are system-dependent. The default retention period is determined when Information Exchange is installed. For installations in the U.S., the default retention period is 30 days. Contact your marketing representative for more information on these values.

If you specify **0** or **blank**, Information Exchange retains the file for the default retention period. If you specify a retention period that is longer than the maximum retention period for your system, Information Exchange retains the file for the default retention period.

## LIST command

Use the LIST command to create a list of account and user IDs to use when sending and receiving files.

### Syntax

#### list

```
listname(list name) function(a|d|e|n) listtype(p|a|g|t)
account(account 1) userid(userid 1)
:
:
alias(alias 1) aliasname(aliasname 1)
:
:
alias(alias n) aliasname(aliasname n)
:
:
sysid(system ID n) account(account n) userid(user ID n);
```

### Example

```
list listname(mylist) function(n) account(xxx1) userid(user01)
sysid(xxx) account(xxx1) userid(user02)
alias(gtbl) aliasname(alias1) account(xxx2) userid(user02);
```

**Results:** A destination list is generated which contains account *xxx1*, user ID *user01*, and aliasname *alias1* from the alias table *gtbl*.

A LIST command can include as many ALIAS and ALIASNAME entries or ACCOUNT and USERID entries as required to create the list. Although you can generally specify parameters in any order, the LIST command has the following limitations:

- LIST command entries must include either an ACCOUNT and USERID or an ALIAS and ALIASNAME. You must pair these entries correctly. For example, you must specify an ACCOUNT parameter with a USERID parameter.
- If you specify the SYSID parameter, you must specify it either before or between the ACCOUNT and USERID parameters to which it belongs. It cannot follow the ACCOUNT and USERID parameters for that system ID.
- At least one list entry is required for functions *a* (add entries), *d* (delete entries), or *n* (new list). No entries are permitted for function *e* (erase entire list).

**Attention:** If you use the LIST command with session-level recovery, do not use the same distribution list name more than once in the same session. If you do, Information Exchange replaces the existing list definitions with the new list definitions. Therefore, any file you send uses the most recent list that you define.

### Parameters

#### listname

Name of the list to process. This parameter is required when you use the LIST command. Use 1 to 8 alphanumeric characters.

#### function

Type of operation you want Expedite Base/400 to perform on the list.

**a** Add the following entries to the list.

- d Delete the following entries from the list. If Expedite Base/400 deletes all the names from the list, you get the same result as with function *e*.
- e Erase the entire list.
- n Create a new list or replace a prior list with the same name specified in the LISTNAME parameter. This will replace an existing list with the same name. This is the default.

**listtype**

Type of list being referenced.

- p The list is a permanent, private list.
- a The list is a permanent list accessible to all members of your account.
- g The list is a permanent list with account level grouping. This type of list is used to limit communication with other users. Use it with care. For more information on this type of list, see *Information Exchange Administration Services User's Guide*.
- t The list is temporary. It goes away when your session ends. This is the default.

**account**

Account of an Information Exchange user from whom you are receiving data.

Expedite Base/400 uses this field together with the USERID parameter to identify the user. If you specify an ACCOUNT parameter, you must also specify a USERID parameter. Use 1 to 8 alphanumeric characters.

**userid**

User ID of an Information Exchange user to whom you are sending data. Expedite Base/400 uses this field together with the ACCOUNT parameter to identify the user. If you specify a USERID parameter, you must also specify an ACCOUNT parameter. Use 1 to 8 alphanumeric characters.

**alias**

Table type and table name of an alias.

- blank An alias name was not used. This is the default.
- gxxx Global alias table, where xxx identifies a 1- to 3-character table name.
- oxxx Organizational alias table, where xxx identifies a 1- to 3-character table name.
- pxxx Private alias table, where xxx identifies a 1- to 3-character table name.



**NOTE:** You create and maintain alias tables by using Information Exchange Administration Services (see *Information Exchange Administration Services User's Guide*), or by using the DEFINEALIAS command (see “DEFINEALIAS command” on page 94).

**aliasname**

Alias name defined in the alias table. Use 1 to 16 alphanumeric characters.



**sysid**

System ID of an Information Exchange user. You need the system ID only if you specify ACCOUNT and USERID parameters for a user on another Information Exchange system. If you specify a SYSID parameter, you must specify the ACCOUNT and USERID parameters. You do not need to specify the system ID if you and your trading partner are on the same system. Use 1 to 3 alphanumeric characters.

## LISTLIBRARIES command

Use the LISTLIBRARIES command to request a list of Information Exchange libraries. Expedite Base/400 returns a list of libraries to which you have access and which meet the criteria you specified on the command.

The message response records associated with the LISTLIBRARIES command are the LIBRARYLIST and RETURN records.

### Syntax

#### **listlibraries**

authority(w|r) selection(a|c)  
owner(*library owner account*);

### Example

```
listlibraries;
```

**Results:** Expedite Base/400 returns a list of all libraries to which you have access.

### Parameters

#### **authority**

Indicates the user's access authority for the list of libraries requested.

w Write or update access authority. This is the default.

r Read access authority.

Expedite Base/400 ignores this parameter if you use SELECTION(C) on this command.

#### **selection**

Indicates the level of library search.

a A list of libraries with a specific owning account. This is the default.

c A complete list of libraries.

#### **owner**

Indicates the owning account to use when SELECTION(A) is specified. The default is the user's account. Expedite Base/400 ignores this parameter if you use SELECTION(C) on this command.

## LISTMEMBERS command

Use the LISTMEMBERS command to receive a list of members within an Information Exchange library.

The message response records associated with the LISTMEMBERS command are the MEMBERLIST and RETURN records.

### Syntax

#### **listmembers**

*owner*(*library owner account*)

**library**(*library name*);

### Example

```
listmembers library(mylib);
```

**Results:** Expedite Base/400 returns a list of members within the library *mylib* to which you have access.

### Parameters

#### **owner**

Account of the library owner. The default is the user's account.

#### **library**

Library containing the members to be listed. You must have read access to the library. This parameter is required.

## LISTVERIFY command

Use the LISTVERIFY command to obtain information about lists. Information Exchange places the responses to the LISTVERIFY command in your Information Exchange mailbox as messages from the \*SYSTEM\* account and \*LSTRSP\* user ID, with a user class of LISTS.

### Syntax

#### listverify

```
listname(list name) function(l|d|c|a|r|s|b)
charge(1|2|3|4|5|6);
```

### Example

```
listverify listname(mylist) function(b);
```

**Results:** Expedite Base/400 verifies that you can both send and receive messages from all users in *mylist*.

### Parameters

#### listname

Name of a previously defined list that you want verified. Use 1 to 8 alphanumeric characters.



**NOTE:** The LISTNAME parameter is required if you select any function other than *a* and *d*.

#### function

Type of operation you want Information Exchange to perform. This parameter is required.



**NOTE:** The LISTNAME parameter is required if you select options: *l*, *c*, *r*, *s*, or *b*. Information Exchange will verify the user IDs in the list and create an error message for any user IDs that cannot be verified. No messages are created when Information Exchange is able to verify the user ID and complete your request.

Options *a* and *d* cannot be used when you specify the LISTNAME parameter.

- 1 List the users in a private, permanent list. Provide the name of the list in the LISTNAME parameter. The message is built with 20-character destination IDs. The format is alphanumeric, left-justified, and padded on the right with blanks. Each destination ID must occupy 20 positions, as follows:

- **Character 1 (coded value).** This indicates whether the address is a true Information Exchange address or an alias ID entry, as follows:

This value:	Indicates:
g	An alias table reference to a global table
o	An organization (account) alias table
p	A private table
i	An intersystem address

blank A reference to a true Information Exchange address

- **Characters 2 through 4 (alphanumeric).** If character 1 is *g*, *o*, or *p*, this field names the alias table to be used to search for the name given in characters 5 through 20. If character 1 is *i*, this is the system identifier. If character 1 is blank, these characters are not used.

- **Characters 5 through 20 (alphanumeric).** If character 1 is *g*, *o*, or *p*, this field specifies the 16-character alias name. If character 1 is blank or *i*, this field specifies the Information Exchange account (characters 5 through 12) and user ID (characters 13 through 20) of the desired destination.

d List all private, permanent lists belonging to you. Information Exchange builds a message that consists of 8-character list names.

c List the users in an account or group level list. Provide the name of the list in the LISTNAME parameter.

Information Exchange builds a message that consists of 20-character user IDs in the same syntax as that used in the FUNCTION(L) parameter (alias type, table name, or Information Exchange address).

a List all of the permanent lists belonging to your account. Information Exchange builds a message that consists of 8-character list names.

r Verify that you can receive messages from all users in the list provided by the LISTNAME parameter. The user IDs in the list are checked to find out whether or not they:

- Are shown in the Information Exchange directory
- Have been authorized to send messages to your user ID

s Verify that you can send messages to all users in the list provided by the LISTNAME parameter. The user IDs in the list are checked to see whether or not they:

- Are valid user IDs of registered Information Exchange users
- Have been authorized to receive messages from your user ID

b Verify that you can both send messages to and receive messages from all users in the list provided by the LISTNAME parameter.

### charge

Specifies the charge authority if you want charge authority verified. When sending messages, specify the same CHARGE parameter that you plan to use in the SEND, SENDEDI, or SENDSTREAM command. When receiving messages, specify the same CHARGE parameter that your trading partner plans to use in the SEND, SENDEDI, or SENDSTREAM command. If you do not specify this parameter, Expedite Base/400 does not verify charge authority. Use 1 numeric character.

1 The receiver pays all charges.

2 The receiver pays all charges, if agreed to by the receiver. Otherwise, the sender and receiver split the charges.

3 The receiver pays all charges, if agreed to by the receiver. If not, the sender and receiver split the charges, if agreed to by the receiver. Otherwise, the sender pays all charges.

- 4 The sender and receiver split the charges, if agreed to by the receiver. Otherwise, the sender pays all charges.
- 5 The sender and receiver split the charges.
- 6 The sender pays all charges.

## MSGINFO command

Use the MSGINFO command to obtain information about messages that are currently available in your mailbox. Expedite Base/400 places a MSGRESP record in the response file as a result of the MSGINFO command. The MSGRESP record contains summary information about available messages. You can obtain detailed information by using the QUERY command.

The MSGINFO command has no parameters.

### Syntax

**msginfo;**

## PURGE command

Use the PURGE command to delete a specific file from your Information Exchange mailbox. To authorize use of this command, the Service Administrator must use Information Exchange Administration Services to set the “Use message purge command” field to **y** in your Information Exchange profile.

The message response record associated with the PURGE command is the RETURN record.

### Syntax

#### **purge**

**msgkey**(*message key*);

### Example

```
purge msgkey(abc1de2fa34bcdef5abb);
```

**Results:** Expedite Base/400 deletes the file having a message key of *abc1de2fa34bcdef5abb* from your mailbox.

### Parameters

#### **msgkey**

Message identifier of the file to be deleted. This 20-character message key can be obtained from the AVAILABLE record for this file in response to a QUERY command.



## PUTMEMBER command

Use the PUTMEMBER command to add a member to an existing Information Exchange library.



**NOTE:** You must have authority to update the library or you will receive a system error message in your mailbox.

### Syntax

#### **putmember**

```
owner(library owner account)
library(library name)
member(member name)
fileid(file ID)
replace(ny)
format(ny)
class(class)
ack(blank|d)
msgname(message name)
msgseqno(message sequence no)
datatype(e|b) truncate(ny)
delimit(n|c|l) crlfeof(crlfeof)
verify(ny) description(description)
destfile(destination file name) destloc(destination file location);
```

### Example

```
putmember owner(abcsales) library(sales1) member(1q04)
fileid(myfiles/prices(send1));
```

**Results:** The member *prices* is placed in *1q04* in *sales1* under the owner *abcsales*. If more than one such file exists in your mailbox, it is appended to *prices*.

### Parameters

#### **library**

Name of the library to update. Use 1 to 8 alphanumeric characters.

#### **member**

Name of the member as it should appear in the library. Use 1 to 8 alphanumeric characters.

#### **owner**

Library owner's account. The default is your account. Use 1 to 8 alphanumeric characters.

#### **fileid**

File to be sent. Use 1 to 33 alphanumeric characters. Format is:

```
library/filename(member)
```

**replace**

Indicates whether a member with the same name should replace an existing library member.

- y Replace the member with the same name as an existing member in the library.
- n Do not replace the member with the same name as an existing member in the library. This is the default.

**format**

Specifies whether you want the data sent as a file or as a free-format message.

- y Format the data as a free-format message. This implies fixed 79-byte records. Expedite Base/400 pads records with blanks. You cannot specify DATATYPE(B), DELIMIT(B), or DELIMIT(L) with this option.
- n Send the data without free-format message formatting. This is the default.

**class**

Specific name for the data. A receiver can use this name to receive only files of this class. If you specify FORMAT(y), Expedite Base/400 defaults to **ffmsg001**; otherwise, the Expedite Base/400 defaults to **blank**. Use 1 to 8 alphanumeric characters.

**ack**

Type of acknowledgments you want to receive from Information Exchange.

- blank No acknowledgments created. This is the default.
- d Delivery acknowledgments only. Acknowledgments are queued to the user ID when Information Exchange delivers the data to the library and reaches a subsequent recovery point with the same destination user ID.

**msgname**

Name you specify for the file. Use 1 to 8 alphanumeric characters.

**msgseqno**

Number you assign to the data as a message control number. Use 1 to 5 alphanumeric characters.

**datatype**

Indicates whether the data is EBCDIC or binary. ASCII systems translate EBCDIC data to ASCII when they receive it from Information Exchange.

- e EBCDIC data. This is the default.
- b Binary data.

**truncate**

Indicates whether you want the Expedite Base/400 to remove trailing blanks from each record before sending them to Information Exchange.

- y Truncate trailing blanks.
- n Do not truncate trailing blanks. This is the default.

**delimit**

Specifies how to preserve the record structure of the file. Expedite Base/400 inserts the appropriate delimiter type into the data and puts the appropriate information into the CDH for the receiver's use.

- c You want the records delimited by CRLF characters. Expedite Base/400 inserts X'0D0A' at the end of each record to indicate the record structure. Expedite Base/400 ends the file with XX'1A'. You can change the CRLFEOF characters with the CRLFEOF parameter.
- l You want the records delimited with delimiters that are 2 bytes long. Expedite Base/400 inserts a 2-byte delimiter at the beginning of each record to delineate the record structure.
- n You want to send the records with no record delimiters. This is the default.

**crlfeof**

Redefines the carriage-return, line-feed, and end-of-file characters used with the DELIMIT(C) option. You need this parameter when sending to a PC that receives with a translate table other than the standard Information Exchange translate table. Use 6 hexadecimal characters. The default value is **0D0A1A**.

**verify**

Verifies that the library is defined and that you have update access before sending the file. You cannot verify a library if it resides on another system.

- y Verify that the library is defined and that you have update access before sending the file. If you cannot verify the library for either reason, the file is not sent and there is a WARNING record in the output file for this command.
- n Do not verify that the library is defined or that you have access. This is the default.

When you use the VERIFY parameter on the PUTMEMBER command, you are not charged for the verification request.

**description**

Provides a free-format description of the file. Use 1 to 79 alphanumeric characters. The description is only available to receiving interfaces that support the CDH. For more information on the CDH, see Appendix B, "Common data header (CDH)."

**destfile**

File name to be used in the CDH as the original file name. If the receiver is using a workstation-based Expedite program and specifies ORIGFILE(Y) on the RECEIVE or RECEIVEEDI command, then Expedite Base uses this file name to store the data when it is received. If you specify a file name that is not valid on the receiver's system, then Expedite Base uses the file name specified by the receiver in the FILEID parameter on the RECEIVE or RECEIVEEDI command. By default, Expedite Base determines the original file name from the FILEID parameter on the SEND, SENDEDI, or PUTMEMBER command. Use 1 to 54 characters.

**destloc**

File location to be used in the CDH. When the file is received or the receiver's mailbox is queried, Expedite Base shows this value in the RECEIVED or AVAILABLE record in the SENDERLOC parameter. By default, Expedite Base uses the file location on the sender's system. Use 1 to 65 characters.

**Notes:**

- 1) Expedite Base/400 users who want to implement the DESTFILE and DESTLOC feature must be using an Expedite Base Version 4 Release 3 or later product on both the sending and receiving systems.
- 2) Expedite Base/400 does not support the ORIGFILE parameter on the RECEIVE or RECEIVEEDI commands. Users on this system must allocate the receive file before running Expedite, and must specify the file name on the RECEIVE or RECEIVEEDI commands.

## QUERY command

The QUERY command returns a list of all files in your Information Exchange mailbox. Expedite Base/400 places AVAILABLE records in the response file in response to the QUERY command. It writes an AVAILABLE record for each file in your mailbox.

### Syntax

#### **query**

```
cdh(y|n);
```

### Example

```
query;
```

### Parameters

#### **cdh**

Indicates whether you want the CDH information included in the response.

y      Include the CDH information in the response. This is the default.

n      Do not include the CDH information in the response.

## RECEIVE command

The RECEIVE command retrieves files from the Information Exchange mailbox.

### Syntax

#### receive

```
alias(alias) aliasname(alias name)
or
sysid(system ID) account(account) userid(user ID)
or
account(account) userid(user ID)
or
listname(list name)
or
requeued(n|y)
fileid(file ID) format(n|y) class(class)
archiveid(archive ID) autoedi(y|n) ediopt(y|n)
delimited(c|l|n) dlmoverride(y|n) endstr(endstr)
resrecl(s|e) allfiles(y|n) nonedionly(n|y) msgkey(message key)
startdate(starting date) starttime(starting time)
enddate(ending date) endtime(ending time) timezone(l|g);
```



**NOTE:** If you are using supported data compression software and receive compressed data, not all parameters are supported. See Appendix D, “Using data compression,” for more information.

### Example

```
receive fileid(library1/sales(price)) class(prices) startdate(040701) starttime(000000)
enddate(041231) endtime(180000);
```

**Results:** Files in your Information Exchange mailbox with a user class of *prices* are received in *price*. If more than one such file exists in your mailbox, it is appended to *price*. Only files sent between 00:00:00 hours (your local time) on July 1, 2004, and 18:00:00 hours on December 31, 2004, are received from your mailbox.

### Parameters

#### alias

Table type and table name of an alias table.

blank	An alias name was not used. This is the default.
gxxx	Global alias table, where <i>xxx</i> identifies a 1- to 3-character table name.
oxxx	Organizational alias table, where <i>xxx</i> identifies a 1- to 3-character table name.
pxxx	Private alias table, where <i>xxx</i> identifies a 1- to 3-character table name.



**NOTE:** You create and maintain alias tables by using Information Exchange Administration Services (see *Using Information Exchange Administration Services*), or by using the DEFINEALIAS command (see “DEFINEALIAS command” on page 94).

**aliasname**

Alias name defined in the alias table. Use 1 to 16 alphanumeric characters.

**sysid**

System ID of a single-source user ID. You need the system ID only if you specify the ACCOUNT and USERID parameters for a user on another Information Exchange system. If you specify a SYSID parameter, you must specify the ACCOUNT and USERID parameters. Use 1 to 3 alphanumeric characters.

**account**

Account of a single-destination user to whom you are sending data. Expedite Base/400 uses this field together with the USERID parameter to identify the user. If you specify an ACCOUNT parameter, you must also specify a USERID parameter. Use 1 to 8 alphanumeric characters.

**userid**

User ID of an Information Exchange user from whom you are receiving data. Expedite Base/400 uses this field together with the ACCOUNT parameter to identify the user. If you specify a USERID parameter, you must also specify an ACCOUNT parameter. Use 1 to 8 alphanumeric characters.

**listname**

Name of a previously defined list of account IDs and user IDs. Use 1 to 8 alphanumeric characters.

**requeded**

Specifies whether Expedite Base/400 receives only files that have been retrieved from archive.

- n Receive all files in your mailbox. This is the default.
- y Receive files retrieved from archive only. If you specify *y*, you cannot specify an Information Exchange source.

**fileid**

Name of the file where the received data should go. This parameter is required. Use 1 to 33 alphanumeric characters. Format is:

library/filename(member)

**format**

Specifies whether you want to receive the data as a file or as a free-format message. See “Free-format messages” on page 67 for more information.



**NOTE:** Expedite Base/400 ignores this option when processing EDI or record delimiters.

- y Expedite Base/400 formats the data as a free-format message. This implies fixed 79-byte records.
- n Expedite Base/400 does not format the data as a free-format message. This is the default.

**class**

Message classification of the files to receive. You can limit the files received to those with the value specified in this field. Use 1 to 8 alphanumeric characters.

You can use a question mark as a *wildcard character* to substitute for any character or characters. For example, to select all messages with a user classification that begins with AB1, you would type: **AB1?**

To select all those with a user classification that ends with 999, you would type: **?999**

If you specify FORMAT(y), the default value is **ffmsg001**, which is reserved for free-format messages. If you specify FORMAT(n), the default is **blank**, which indicates all classes.

#### archiveid

Archive reference identifier you want assigned to the files delivered in response to RECEIVE command requests. Use 1 to 8 alphanumeric characters.

#### autoedi

Specifies whether EDI processing (as defined in Chapter 4, “Sending and receiving EDI data,”) should be done automatically for files if the CDH indicates that the files are EDI formatted.

- y Automatically perform EDI processing if the CDH indicates that the file is EDI formatted. This is the default.
- n Do not perform EDI processing for any of the files.



**NOTE:** This parameter does not do anything for files sent without a CDH. For more information on the CDH, see Appendix B, “Common data header (CDH).”

#### ediopt

Specifies whether EDI data should be split at the end of EDI segments. This option is ignored unless the CDH indicates that the received message is EDI data and AUTOEDI is set to y.

- y Split records at the end of EDI segments if the CDH indicates that the message is EDI formatted. This is the default.
- n Do not split records at the end of EDI segments.

#### delimited

Determines the record structure of the received data. If you specify y for DLMOVERRIDE, Expedite Base/400 uses the information in the DELIMITED parameter even if the CDH indicates the record is structured differently. If you specify n for DLMOVERRIDE, Expedite Base/400 ignores the DELIMITED parameter if the CDH indicates the record structure of the data.

- c Split records at CRLF characters. If the CDH includes the CRLFEOF parameter, Expedite Base/400 uses the CRLF characters from that parameter. Otherwise, Expedite Base/400 uses X'0D0A' for the CRLF.
- l Split records according to the 2-byte length delimiters at the beginning of each record.
- n Store the data as it is received. The record length depends on the record length of the data set allocated to receive the data. This is the default.



**dlmoverride**

Specifies whether the DELIMITED parameter should override the structure specified by the CDH.

- y Format the data according to the DELIMITED parameter, even if the CDH indicates a record delimiter type.
- n If the CDH indicates a record delimiter type, format the data according to the CDH. This is the default.

**endstr**

Specifies the string used at the end of each message received. This permits you to separate messages within a file.

- s: Begin the delimiter string with *s*: to place the delimiter immediately after the received data, regardless of record boundaries.
- r: . Begin the delimiter string with *r*: to place the delimiter at the beginning of the record, following the received data. This is the default

The *s*: and the *r*: are not counted as part of the delimiter string and are not placed in the received file. If you do not specify this parameter, or if you leave it blank, Expedite Base/400 does not write a delimiter string at the end of each message. Use 1 to 79 alphanumeric characters.

**resrecl**

Indicates how Expedite Base/400 should resolve a disparity between the length of the record built while receiving data and the logical record length of the data set allocated to receive data.



**NOTE:** If the data is processed as EDI data (using the AUTOEDI option), records are always split if they are too long for the data set. If the data does not contain delimiters, record length is determined by the logical record length of the response data set, and this does not apply.

- e If the record received is too long for the data set, terminate the session with an error.  
  
This option is valid only for session-level recovery, because you might find it difficult to correct the error and still restart correctly using checkpoint-level recovery.
- s If the record received is too long for the data set, split the record and write a warning message to the response file. This is the default.

**allfiles**

Specifies whether Expedite Base/400 should receive all files that match the RECEIVE specifications or just the first file in the Information Exchange mailbox that matches the RECEIVE specifications.

- y Receive all files that match the RECEIVE specifications. This is the default.
- n Receive only the first file that matches the RECEIVE specifications.

**nonedionly**

Specifies that only non-EDI data should be received.

- y Receive only those messages from your mailbox that are identified in the CDH as not having one of the EDI formats.
- n Receive all messages from your mailbox that satisfy your RECEIVE request. This is the default.

**msgkey**

Unique message key you can use to receive a specific file from the mailbox. Its value is taken from the AVAILABLE record in response to a QUERY command. Use 20 hexadecimal characters.

**startdate**

Starting date of a time range for the files you want to receive from Information Exchange. For a file to qualify to be received, the date the file was sent to Information Exchange must fall within this date range. The format is *YYMMDD* or *YYYYMMDD*. The default value is determined by Information Exchange, and is currently **000102** (January 2, 1900).

If you limit the range to one day and an error occurs while sending the file, files could be left undetected in the mailbox. For example, assume that an error occurs while sending a file on June 11, 2004, and the file is not actually placed in the mailbox until 11:00 a.m. on June 12.

If you issue a RECEIVE command at 8:00 a.m. on June 12, 2004 with a range of 00:00:00 to 24:00:00 on June 11 specified, the file will not be received because it has not yet arrived in the mailbox. If you then issue another RECEIVE command at 8:00 a.m. on June 13, 2004 with a range of 00:00:00 to 24:00:00 on June 12 specified, the file will still not be received because its starting date of June 11 is outside of the specified range.

To avoid missing any files, you should either move the starting date earlier, or periodically check your mailbox.

**starttime**

Starting time of a time range for the files you want to receive from Information Exchange. For a file to qualify to be received, the time the file was sent to Information Exchange must fall within this time range. The format is *HHMMSS*. The default value is determined by Information Exchange, and is currently **000000** (0 hours).

**enddate**

Ending date of a time range for the files you want to receive from Information Exchange. For a file to qualify to be received, the date the file was sent to Information Exchange must fall within this date range. The format is *YYMMDD* or *YYYYMMDD*. The default value is determined by Information Exchange, and is currently **420916** (September 16, 2042).

**endtime**

Ending time of a time range for files you want to receive from Information Exchange. For a file to qualify to be received, the time the file was sent to Information Exchange must fall within this time range. Use the *HHMMSS* format for the ending time. The default value is determined by Information Exchange, and is currently **235347** (23:53:47).

**timezone**

Time reference in the STARTTIME and ENDTIME parameters.

- l Your local time, as specified by the TIMEZONE parameter of the IDENTIFY command. This is the default.

g      Greenwich mean time (GMT).

## RECEIVEEDI command

Use the RECEIVEEDI command to retrieve EDI-formatted files from the Information Exchange mailbox.

You can use the CLASS parameter to make sure you are receiving EDI data. To do so, you and your trading partner must agree on a name for the EDI data. Your trading partner must specify that name in the user CLASS parameter when sending the data, and you must specify that name in the CLASS parameter on the RECEIVEEDI command. The following example shows a RECEIVEEDI command using the CLASS parameter, which can be used to receive all the files with the class 'editest' from the mailbox:

```
receiveedi fileid(filea) class(editest)
```

You can also use the EDIONLY parameter to receive the data marked in the CDH as EDI data. If your trading partner sent the data with an interface that supports the DFORMAT field in the CDH, then the CDH will be marked properly. When you use a RECEIVEEDI command with EDIONLY(Y) specified, Expedite Base/400 does not receive files that are lacking a CDH.

Information Exchange interfaces before Release 3.0 do not support the CDH.

The following example shows a RECEIVEEDI command using the EDIONLY parameter. You can use this parameter to receive all the mailbox files that are marked in the CDH as EDI data:

```
receiveedi fileid(filea) edionly(y)
```

If you receive a file whose CDH indicates that the file is not EDI data, Expedite Base/400 receives the file into records in the format indicated by the CDH. If a file has no CDH and is not recognizable as EDI data, Expedite Base/400 receives the file without reformatting records. For more information on the CDH, see Appendix B, "Common data header (CDH)."

For more information on receiving EDI data, see "Receiving EDI data" on page 42.

### Syntax

#### receiveedi

```
alias(alias) aliasname(alias name)
    or
sysid(system ID) account(account) userid(user ID)
    or
account(account) userid(user ID)
    or
listname(list name)
    or
requeued(n|y)
fileid(file ID) class(class) archiveid(archive ID)
ediop(y|n|f) allfiles(y|n) edionly(n|y)
msgkey(message key)
startdate(starting date) starttime(starting time)
enddate(ending date) endtime(ending time)
timezone(l|g);
```

### Example

```
receiveedi fileid(orders) class(prices) startdate(040701) starttime(000000) enddate(041231)
endtime(180000);
```

**Results:** Files in your Information Exchange mailbox with a user class of *prices* are received in *orders*. If more than one such file exists in your mailbox, it is appended to *orders*. Only files sent between 00:00:00 hours (your local time) on July 1, 2004, and 18:00:00 hours on December 31, 2004, are received from your mailbox.

## Parameters

### alias

Table type and table name of an alias table.

blank	An alias name was not used. This is the default.
gxxx	Global alias table, where xxx identifies a 1- to 3-character table name.
oxxx	Organizational alias table, where xxx identifies a 1- to 3-character table name.
pxxx	Private alias table, where xxx identifies a 1- to 3-character table name.



**NOTE:** You create and maintain alias tables by using Information Exchange Administration Services (see *Using Information Exchange Administration Services*), or by using the DEFINEALIAS command (see “DEFINEALIAS command” on page 94).

### aliasname

Alias name defined in the alias table. Use 1 to 16 alphanumeric characters.

### sysid

System ID of a single-source user ID. You need the system ID only if you specify the ACCOUNT and USERID parameters for a user on another Information Exchange system. If you specify a SYSID parameter, you must specify the ACCOUNT and USERID parameters. Use 1 to 3 alphanumeric characters.

### account

Account of a single-destination user to whom you are sending data. Expedite Base/400 uses this field together with the USERID parameter to identify the user. If you specify an ACCOUNT parameter, you must also specify a USERID parameter. Use 1 to 8 alphanumeric characters.

### userid

User ID of an Information Exchange user from whom you are receiving data. Expedite Base/400 uses this field together with the ACCOUNT parameter to identify the user. If you specify a USERID parameter, you must also specify an ACCOUNT parameter. Use 1 to 8 alphanumeric characters.

### listname

Name of a previously defined list of accounts and user IDs. Use 1 to 8 alphanumeric characters.

### requeued

Specifies whether Expedite Base/400 only receives files that have been retrieved from archive.

n	Receive any file in your mailbox. This is the default.
y	Receive files retrieved from archive only. If you specify y, you cannot specify an Information Exchange destination.

**fileid**

Name of the file where the received data should go. This parameter is required. Use 1 to 33 alphanumeric characters. Format is:

library/filename(member)

**class**

Specifies the message classification of the file you are receiving. You can limit the files received to those with the value specified in this field. You should always specify a class when receiving EDI files. Use 1 to 8 alphanumeric characters. The default value is **blank**, which indicates all classes.

If the EDI data was sent with a `SENDEDI` command using the default class, it arrives in your Information Exchange mailbox classified as follows:

#ee	EDIFACT data
#eu	UN/TDI data
#e2	X12 data
#ec	UCS data

You can use a question mark as a wildcard to substitute for any character or characters. For example, to receive only the EDI files sent with the default class, type: **#E?**

To select all EDI files with a class ending in 9, type: **?9**

**archiveid**

Archive reference ID to be assigned to the files delivered in response to `RECEIVEEDI` requests. Use 1 to 8 alphanumeric characters.

**ediopt**

Specifies whether Expedite Base/400 should split EDI data at the end of EDI segments.

- y Split records at the end of EDI segments. If the CDH indicates that the information received is not EDI data, this option is ignored. This is the default.
- n Do not split records at the end of EDI segments. If the CDH indicates that the information received is not EDI data, this option is ignored.
- f Attempt to split records at the end of EDI segments, regardless of what the common data header (CDH) indicates. If the data is not valid EDI data, Expedite Base/400 issues a warning.

**allfiles**

Specifies whether Expedite Base/400 should receive all files that match the `RECEIVEEDI` specifications or just the first file in the Information Exchange mailbox that matches the `RECEIVEEDI` specifications.

- y Receive all files that match the `RECEIVEEDI` specifications. This is the default.
- n Receive only the first file that matches the `RECEIVEEDI` specifications.

**edionly**

Specifies that only EDI data should be received.

- y Receive only those messages from your mailbox that are identified in the CDH as having one of the EDI formats.
- n Receive all messages from your mailbox that satisfy your RECEIVEEDI request. If a received file's CDH indicates that the files do not contain EDI-formatted data, Expedite Base/400 receives the file into records using the format indicated in the CDH. If a file has no CDH and is not recognizable as EDI data, Expedite Base/400 receives the file without reformatting the records.

**msgkey**

Unique message key you can use to receive a specific file from the mailbox. Its value is taken from the AVAILABLE record in response to a QUERY command. Use 20 hexadecimal characters.

**startdate**

Starting date of a time range for the files you want to receive from Information Exchange. For a file to qualify to be received, the date the file was sent to Information Exchange must fall within this date range. The format is *YYMMDD* or *YYYYMMDD*. The default value is determined by Information Exchange, and is currently **000102** (January 2, 1900).

If you limit the range to one day and an error occurs while sending the file, files could be left undetected in the mailbox. To avoid missing any files, you should either move the starting date earlier, or periodically check your mailbox. See the STARTDATE parameter on the RECEIVE command (page 116) for an example.

**starttime**

Starting time of a time range for the files you want to receive from Information Exchange. For a file to qualify to be received, the time the file was sent to Information Exchange must fall within this time range. The format is *HHMMSS*. The default value is determined by Information Exchange, and is currently **000000** (0 hours).

**enddate**

Ending date of a time range for the files you want to receive from Information Exchange. For a file to qualify to be received, the date the file was sent to Information Exchange must fall within this time range. The format is *YYMMDD* or *YYYYMMDD*. The default value is determined by Information Exchange, and is currently **420916** (September 16, 2042).

**endtime**

Ending time of a time range for files you want to receive from Information Exchange. For a file to qualify to be received, the time the file was sent to Information Exchange must fall within this time range. Use the *HHMMSS* format for the ending time. The default value is determined by Information Exchange, and is currently **235347** (23:53:47).

**timezone**

Time reference in the STARTTIME and ENDTIME parameters.

- l Your local time, as specified by the TIMEZONE parameter of the IDENTIFY command. This is the default.
- g Greenwich mean time (GMT).

## RECEIVESTREAM command

Use the RECEIVESTREAM command to receive data directly into the message response file. When you receive data using the RECEIVESTREAM command, the data immediately follows the RECEIVED record in the response file. This command is valid only with session-level recovery.

### Syntax

#### receivestream

```
alias(alias) aliasname(alias name)
    or
sysid(system ID) account(account) userid(user ID)
    or
account(account) userid(user ID)
    or
listname(list name)
    or
requeued(n|y)
endstr(end string) class(class) archiveid(archive ID)
allfiles(y|n) msgkey(message key)
startdate(starting date) starttime(starting time)
enddate(ending date) endtime(ending time) timezone(l|g);
```

### Example

```
receivestream account(act3) userid(user005) endstr(xyzz) class(orders);
```

**Results:** Expedite Base/400 places all files from account *act3* and user ID *user005* with a class of *orders* in your mailbox. Expedite Base/400 places a delimiter of *xyzz* at the end of the received data.



**NOTE:** If you are using supported data compression software and receive compressed data, not all parameters are supported. See Appendix D, “Using data compression,” for more information.

### Parameters

#### alias

Table type and table name of an alias table.

blank	An alias name was not used. This is the default.
gxxx	Global alias table, where <i>xxx</i> identifies a 1- to 3-character table name.
oxxx	Organizational alias table, where <i>xxx</i> identifies a 1- to 3-character table name.
pxxx	Private alias table, where <i>xxx</i> identifies a 1- to 3-character table name.



**NOTE:** You create and maintain alias tables by using Information Exchange Administration Services (see *Using Information Exchange Administration Services*), or by using the DEFINEALIAS command (see “DEFINEALIAS command” on page 94).

#### aliasname

Alias name defined in the alias table. Use 1 to 16 alphanumeric characters.



**sysid**

System ID of a single-source user ID. You need the system ID only if you specify the ACCOUNT and USERID parameters for a user on another Information Exchange system. If you specify a SYSID parameter, you must specify the ACCOUNT and USERID parameters. Use 1 to 3 alphanumeric characters.

**account**

Account name of a single-source user ID from whom you are receiving data. Expedite Base/400 uses this field together with the USERID parameter to identify the user. If you specify an ACCOUNT parameter, you must also specify a USERID parameter. Use 1 to 8 alphanumeric characters.

**userid**

User ID of an Information Exchange user from whom you are receiving data. Expedite Base/400 uses this field together with the ACCOUNT parameter to identify the user. If you specify a USERID parameter, you must also specify an ACCOUNT parameter. Use 1 to 8 alphanumeric characters.

**listname**

Name of a previously defined list of accounts and user IDs. Use 1 to 8 alphanumeric characters.

**queued**

Option you can use to indicate whether or not Expedite Base/400 should receive only files that have been retrieved from archive.

- n Receive any file in your mailbox. This is the default.
- y Receive files retrieved from archive only. If you specify *y*, you cannot specify an Information Exchange destination.

**endstr**

Delimiter string that Expedite Base/400 places in the response file following the received data.

Begin the delimiter string with *s*: to place the delimiter after the received data, regardless of record boundaries. Begin the delimiter string with *r*: to place the delimiter at the beginning of the record following the received data.

The default is **r:**. Expedite Base/400 does not consider the *s*: and the *r*: part of the delimiter and does not place them in the response file. The ENDSTR parameter is required. Use 1 to 79 alphanumeric characters.

**class**

Message classification of the files to receive. You can limit the files received to those with the value specified in this field. You can use a question mark as a wildcard to substitute for any character or characters at the beginning or end of a classification that you do not know.

For example, to select all messages with a user classification that begins with *ABI*, you would type: **AB1?**. To select all those with a user classification that ends with *999*, you would type: **?999**.

Use 1 to 8 alphanumeric characters. The default is blank, which indicates all classes.

**archiveid**

Archive reference identifier Expedite Base/400 must assign to the data delivered in response to RECEIVESTREAM requests. Use 1 to 8 alphanumeric characters.

**allfiles**

Specifies whether Expedite Base/400 should receive all data that meets the RECEIVESTREAM specifications or only the first item that matches the RECEIVESTREAM specifications.

- y Receive all data in the mailbox that matches the RECEIVESTREAM specifications. This is the default.
- n Receive only the first item that matches the RECEIVESTREAM specifications.

**msgkey**

Unique message key to receive a specific file from the mailbox. Use the QUERY command to get this value. Use 20 hexadecimal characters.

**startdate**

Starting date of a time range for the files you want to receive from Information Exchange. For a file to qualify to be received, the date the file was sent to Information Exchange must fall within this date range. The format is *YYMMDD* or *YYYYMMDD*. The default value is determined by Information Exchange, and is currently **000102** (January 2, 1900).

If you limit the range to one day and an error occurs while sending the file, files could be left undetected in the mailbox. To avoid missing any files, you should either move the starting date earlier, or periodically check your mailbox. See the STARTDATE parameter on the RECEIVE command (page 116) for an example.

**starttime**

Starting time of a time range for the files you want to receive from Information Exchange. For a file to qualify to be received, the time the file was sent to Information Exchange must fall within this time range. The format is *HHMMSS*. The default value is determined by Information Exchange, and is currently **000000** (0 hours).

**enddate**

Ending date of a time range for the files you want to receive from Information Exchange. For a file to qualify to be received, the date the file was sent to Information Exchange must fall within this date range. The format is *YYMMDD* or *YYYYMMDD*. The default value is determined by Information Exchange, and is currently **420916** (September 16, 2042).

**endtime**

Ending time of a time range for the messages and files you want to receive from Information Exchange. For a file to qualify to be received, the time the file was sent to Information Exchange must fall within this time range. Use the *HHMMSS* format for the ending time. The default value is determined by Information Exchange, and is currently **235347** (23:53:47).

**timezone**

Time reference in the STARTTIME and ENDTIME parameters.

- l Your local time, as specified by the TIMEZONE parameter of the IDENTIFY command. This is the default.
- g Greenwich mean time (GMT).

## SEND command

Use the SEND command to send a file to Information Exchange.

### Syntax

#### send

```
alias(alias) aliasname(alias name)
    or
sysid(system ID) account(account) userid(user ID)
    or
account(account) userid(user ID)
    or
listname(list name)
fileid(file ID) format(n|y)
class(class) mode(blank|t) priority(blank|i|p)
charge(1|2|3|4|5|6) ack(blank|a|b|c|d|e|f|r) msgname(message name)
msgseqno(message sequence no) datatype(e|b) truncate(n|y)
delimit(c|l|n|u) crlfeof(crlfeof) verify(n|y|f)
description(description) retain(retain time) compress(n|y|t|
destfile(destination file) destloc(destination location)
selectrcv(n|f|blank);
```

### Example

```
send fileid(testfile) alias(ptb3) aliasname(name) class(question);
```

**Results:** Expedite Base/400 sends file *testfile* to Information Exchange for alias *ptb3* and aliasname *name*. The file is assigned a class of *question*.

### Parameters

#### alias

Table type and table name of an alias table.

blank	An alias table name was not used. This is the default.
gxxx	Global alias table, where xxx identifies a 1- to 3-character table name.
oxxx	Organizational alias table, where xxx identifies a 1- to 3-character table name.
pxxx	Private alias table, where xxx identifies a 1- to 3-character table name.



**NOTE:** You create and maintain alias tables by using Information Exchange Administration Services (see *Using Information Exchange Administration Services*), or by using the DEFINEALIAS command (see “DEFINEALIAS command” on page 94).

Alias name defined in the alias table. Use 1 to 16 alphanumeric characters.

#### sysid

System ID of an Information Exchange user to whom you are sending data. You need the system ID only if you specify the ACCOUNT and USERID parameters for a user on another Information Exchange system. If you specify a SYSID parameter, you must specify the ACCOUNT and USERID parameters. Use 1 to 3 alphanumeric characters.

**account**

Account of an Information Exchange user to whom you are sending data. Expedite Base/400 uses this field together with the `USERID` parameter to identify the user. If you specify an `ACCOUNT` parameter, you must also specify a `USERID` parameter. Use 1 to 8 alphanumeric characters.

**userid**

User ID of an Information Exchange user to whom you are sending data. Expedite Base/400 uses this field together with the `ACCOUNT` parameter to identify the user. If you specify a `USERID` parameter, you must also specify an `ACCOUNT` parameter. Use 1 to 8 alphanumeric characters.

**listname**

Name of a previously defined list of account IDs and user IDs. Use 1 to 8 alphanumeric characters.

**fileid**

File to be sent. Use 1 to 33 alphanumeric characters. Format is:

library/filename(member)

**format**

Specifies whether you want the data sent as a file or as a free-format message. See “Free-format messages” on page 67.

`y`            Format the data as a free-format message. This implies fixed 79-byte records. Expedite Base/400 pads records with blanks. You cannot specify `DATATYPE(B)`, `DELIMIT(C)`, or `DELIMIT(L)` with this option.

`n`            Send the data without free-format message formatting. This is the default.

**class**

Specific name for the data. A receiver can use this name to receive only files of this class.

If you specify `FORMAT(Y)`, this defaults to **ffmsg001**. Otherwise, it defaults to **blank**. Use 1 to 8 alphanumeric characters.

**mode**

Indicates whether the file is a test file or a normal file.

`blank`        Normal file

`t`            Test-mode file

The default is `blank`.

**priority**

Class of delivery service for this file.

`blank`        Normal priority. This is the default.

`i`            Express delivery to those users who have continuous receive capability and are currently in session with Information Exchange. (Expedite Base/400 does not support continuous receive capability.)

`p`            High priority.

**charge**

Indicates to Information Exchange how the sender wants the file charges paid.

- 1 The receiver pays all charges.
- 2 The receiver pays all charges, if agreed to by the receiver. Otherwise, the sender and receiver split the charges.
- 3 The receiver pays all charges, if agreed to by the receiver. If not, the sender and receiver split the charges, if agreed to by the receiver. Otherwise, the sender pays all charges. This is the default.
- 4 The sender and receiver split the charges, if agreed to by the receiver. Otherwise, the sender pays all charges.
- 5 The sender and receiver split the charges.
- 6 The sender pays all charges.

**ack**

Type of acknowledgments you want to receive from Information Exchange.

- |       |  |
|-------|--|
| blank | No acknowledgments created. This is the default.   |
| a     | Purge acknowledgments only.  |
| b     | Both receipt and delivery acknowledgments.   |
| c     | Both receipt and purge acknowledgments.  |
| d     | Delivery acknowledgments only. Acknowledgments are queued to the user ID when Information Exchange delivers the data to a destination user ID and reaches a subsequent recovery point with the same destination user ID. |
| e     | Either purge or delivery acknowledgments.  |
| f     | Receipt acknowledgments and either purge or delivery acknowledgments.  |
| r     | Receipt acknowledgments only when Expedite Base/400 finishes sending a file.   |

**msgname**

Name you specify for the file. Use 1 to 8 alphanumeric characters.

**msgseqno**

Number assigned by you as a message control number for the data. Use 1 to 5 alphanumeric characters.

**datatype**

Indicates whether the data is EBCDIC or binary. ASCII systems translate EBCDIC data to ASCII when they receive it from Information Exchange. For more information on EBCDIC and binary data, see Chapter 3, “Communicating with other operating systems.”

- |   |                                   |
|---|-----------------------------------|
| e | EBCDIC data. This is the default. |
| b | Binary data.                      |

**truncate**

Indicates whether you want Expedite Base/400 to remove trailing blanks from each record before sending them to Information Exchange.

- y Truncate trailing blanks.
- n Do not truncate trailing blanks. This is the default.

**delimit**

Indicates how to preserve the record structure of the file. Expedite Base/400 puts the appropriate delimiter type into the data and puts the appropriate information into the CDH for the receiver's use.

- c You want the records delimited by CRLF characters. Expedite Base/400 inserts X'0D0A' at the end of each record to indicate the record structure. Expedite Base/400 ends the file with X'1A'. You can change the CRLF characters with the CRLFEOF parameter.
- l You want the records delimited with delimiters 2 bytes long. Expedite Base/400 inserts a 2-byte length at the beginning of each record to delineate the record structure.
- n You want to send the records with no record delimiters. This is the default.
- u You do not want delimiting characters added; the records you are sending are already delimited with CRLF characters, and you want the CDH to indicate this.

**crlfeof**

Defines the carriage-return, line-feed, and end-of-file characters used with the DELIMIT(c) option. You need this parameter when sending to a PC that receives with a translate table other than the standard Information Exchange translate table. Use 6 hexadecimal characters. The default value is **0D0A1A**.

**verify**

Verifies that the receiver exists before sending the file.

- n Do not verify that the receiver exists. This is the default.
- y Verify that the receiver exists before sending the data.
- f Verify that the receiver exists. If Information Exchange cannot tell whether the receiver exists (for example, if the receiver is on another system), send the data anyway.

If you request verification, you might incur an Information Exchange message charge. See "Validating addresses, payment levels, and authorizations" on page 67 for more information on validations.

**description**

Free-format description of the file sent. Use 1 to 79 alphanumeric characters. The description is available only to receiving interfaces that support the CDH. For more information on the CDH, see Appendix B, "Common data header (CDH)."

**retain**

Number of days Information Exchange keeps the file in the mailbox if no one receives it. Valid values are **blank** and **0** through **180**.

The maximum retention period and the default retention period can be different on different Information Exchange systems. These periods are system-dependent. The default retention period is determined when Information Exchange is installed. For installations in the U.S., the default retention period is 30 days. Contact your marketing representative for more information on these values.

If you specify **0** or **blank**, Information Exchange retains the file for the default retention period. If you specify a retention period that is longer than the maximum retention period for your system, Information Exchange retains the file for the default retention period.

**compress**

Indicates whether the specified file should be compressed.

- n Do not compress the specified file. This is the default.
- y Compress the specified file.
- t For each sender/receiver pair, use the setting of the COMPRESS parameter (*y* or *n*) indicated in the CPLOOKUP table.

If the receiver/sender pair is not in the CPLOOKUP table, send the data, and write a warning to the response file, OUTMSG.

You must have the supported compression software installed to use data compression.

Some of the parameters of this command may not apply when sending compressed data; see Appendix D, "Using data compression," for more information.

**destfile**

File name to be used in the Common Data Header (CDH) as the original file name. If the receiver is using a workstation-based Expedite Base program and specifies ORIGFILE(Y) on the RECEIVE or RECEIVEEDI command, then Expedite Base uses this file name to store the data when it is received. If you specify a file name that is not valid on the receiver's system, then Expedite Base uses the file name specified by the receiver in the FILEID parameter on the RECEIVE or RECEIVEEDI command. By default, Expedite Base determines the original file name from the FILEID parameter on the SEND, SENDEDI, or PUTMEMBER command. Use 1 to 54 characters.

**destloc**

File location to use in the Common Data Header (CDH). When the file is received or the receiver's mailbox is queried, Expedite Base shows this value in the RECEIVED or AVAILABLE record in the SENDERLOC parameter. By default, Expedite Base uses the file location on the sender's system. Use 1 to 65 characters.

**Notes:**

- 1) Expedite Base/400 users who want to implement the DESTFILE and DESTLOC feature must be using an Expedite Base Version 4 Release 3 or later product on both the sending and receiving systems.

- 2) Expedite Base/400 does not support the ORIGFILE parameter on the RECEIVE or RECEIVEEDI commands. Users on this system must allocate the receive file before running Expedite, and must specify the file name on the RECEIVE or RECEIVEEDI commands.

**selectrcv**

Indicates whether the file should be marked in Information Exchange so that the file cannot be received by a blanket receive.

- |       |  |
|-------|--|
| n     | Allows the file to be received by a blanket receive.   |
| f     | Marks the file in Information Exchange so that the file cannot be received by a blanket receive. |
| blank | Allows the file to be received by a blanket receive. This is the default.                        |

When **f** is selected in SELECTRCV, the file being sent is marked in Information Exchange and cannot be received by a blanket receive. The file can only be received if the receiver specifies *one* of the following:

- Sender ID (account ID and user ID)
- Message class
- Message key



## SENDEDI command

Use the SENDEDI command to send an EDI-formatted file to Information Exchange. The file can contain X12, UCS, EDIFACT, UN/TDI data, or any combination of these. For more information on sending EDI data, see Chapter 4, “Sending and receiving EDI data.”

### Syntax

#### sendedi

**fileid**(*file ID*) mode(**blank**|**t**)  
 priority(**blank**|**i**|**p**) charge(**1**|**2**|**3**|**4**|**5**|**6**) ack(**blank**|**a**|**b**|**c**|**d**|**e**|**f**|**r**)  
 msgname(*message name*) msgseqno(*message sequence no*) class(*class*)  
 verify(**n**|**y**|**f**|**c**|**g**) description(*description*) retain(*retain time*)  
 compress(**n**|**y**|**t**) destfile(*destination file*) destloc(*destination location*)  
 selectrcv(*n*|**f**|**blank**);

### Example

```
sendedi fileid(edidata) class(question) retain(0);
```

**Results:** Expedite Base/400 sends file *edidata* to Information Exchange with a class of *question*. The default retention period applies.

### Parameters

#### fileid

File you want to send. Use 1 to 33 alphanumeric characters. Format is:

library/filename (member)

#### mode

Indicates whether a file is a test or a normal file.

**blank** Normal file. This is the default.

**t** Test-mode file.

#### priority

Class of delivery service for this file.

**blank** Normal priority. This is the default.

**i** Express delivery to those users who have the continuous receive capability and are currently in session with Information Exchange. (Expedite Base/400 does not support the continuous receive capability.)

**p** High priority.

**charge**

Specifies to Information Exchange how the sender wants the file charges paid.

- |   |  |
|---|--|
| 1 | Receiver pays all charges.   |
| 2 | Receiver pays all charges if agreed to. Otherwise, the sender and receiver split the charges.  |
| 3 | Receiver pays all charges if agreed to; otherwise, the sender and receiver split the charges, if agreed to by the receiver. Otherwise, the sender pays all charges. This is the default. |
| 4 | Sender and receiver split the charges, if agreed to by the receiver. Otherwise, the sender pays all charges.   |
| 5 | Sender and receiver split the charges.   |
| 6 | Sender pays all charges.   |

**ack**

Type of acknowledgments you want to receive from Information Exchange.

- |       |  |
|-------|--|
| blank | No acknowledgments are created. This is the default.   |
| a     | Purge acknowledgments only.  |
| b     | Both receipt and delivery acknowledgments.   |
| c     | Both receipt and purge acknowledgments.  |
| d     | Delivery acknowledgments only. Acknowledgments are queued to the user ID when Information Exchange delivers the file to a destination user ID and reaches a subsequent recovery point with that destination user ID. |
| e     | Either purge or delivery acknowledgments.  |
| f     | Receipt and either purge or delivery acknowledgments.  |
| r     | Receipt acknowledgments only when Expedite Base/400 finishes sending a file.   |

**msgname**

Name you specify for the file. See “Message name (MSGNAME) assignment” on page 40 for more information on the default values of MSGNAME. Use 1 to 8 alphanumeric characters.

**msgseqno**

Number assigned by you as a message control number for the data. See “Message sequence number (MSGSEQNO) assignment” on page 40 for more information on the default values of MSGSEQNO. Use 1 to 5 alphanumeric characters.

**class**

Unique name for this file. This name can be used by a receiver to receive only files of this class.

If you do not specify the class with either the CLASS parameter or the EDI envelope, the message class defaults as follows:

This EDI data type:	Uses this default message class:
EDIFACT	#ee
UCS	#ec
UN/TDI	#eu
X12	#e2

See “Message class (CLASS) assignment” on page 41 for more information on CLASS assignment. Use 1 to 8 alphanumeric characters.

### verify

Verifies that the receiver exists before sending the file.

- n Do not verify that the receiver exists. This is the default.
- y Verify that the receiver exists before sending the data. If the verification fails or the destination cannot be verified, the envelope is not sent. If there are multiple envelopes in the file, the envelopes following the one with the error are not sent.
- f Verify that the receiver exists. If Information Exchange cannot verify whether the receiver exists (for example, if the receiver is on another system), send the data anyway. If the verification fails for an envelope destined for the same Information Exchange system as the sender, the envelope is not sent. If there are multiple envelopes in the file, the envelopes following the one with the error are not sent.
- c Verify that the receiver exists before sending the data. If the verification fails or the destination cannot be verified, the envelope is not sent. If there are multiple envelopes in the file, those envelopes following the one in error are still processed.
- g Verify that the receiver exists before sending the data. If Information Exchange cannot verify whether the receiver exists (for example, if the receiver is on another system), send the data anyway. If verification fails for an envelope destined for the same Information Exchange system as the sender, the envelope is not sent. If there are multiple envelopes in the file, those envelopes following the one in error are still processed.

If you request verification, you might incur an Information Exchange message charge. See “Validating addresses, payment levels, and authorizations” on page 67 for more information.

### description

Free-format description of the file sent. The description is only available to receiving interfaces that support the CDH. For more information on the CDH, see Appendix B, “Common data header (CDH).” Use 1 to 79 alphanumeric characters.

**retain**

Number of days that Information Exchange keeps the file in the mailbox if no one receives it. Valid values are **blank** and **0** through **180**.

The maximum retention period and the default retention period can be different on different Information Exchange systems. These periods are system-dependent. The default retention period is determined when Information Exchange is installed. For installations in the U.S., the default retention period is 30 days. Contact your marketing representative for more information.

If you specify **0** or **blank**, Information Exchange retains the file for the default retention period. If you specify a retention period that is longer than the maximum retention period for your system, Information Exchange retains the file for the default retention period.

**compress**

Indicates whether the specified file should be compressed.

- n Do not compress the specified file. This is the default.
- y Compress the specified file.
- t For each sender/receiver pair, use the setting of the COMPRESS parameter (*y* or *n*) indicated in the CPLOOKUP table.

If the receiver/sender pair is not in the CPLOOKUP table, send the data, and write a warning to the response file, OUTMSG.

You must have the supported compression software installed to use data compression.

Some of the parameters of this command may not apply when sending compressed data; see Appendix D, "Using data compression," for more information.

**destfile**

File name to use in the CDH as the original file name. If the receiver is using a workstation-based Expedite Base program and specifies ORIGFILE(Y) on the RECEIVE or RECEIVEEDI command, then Expedite Base uses this file name to store the data when it is received. If you specify a file name that is not valid on the receiver's system, then Expedite Base uses the file name specified by the receiver in the FILEID parameter on the RECEIVE or RECEIVEEDI command. By default, Expedite Base determines the original file name from the FILEID parameter on the SEND, SENDEDI, or PUTMEMBER command. Use 1 to 54 characters.

**destloc**

File location to use in the CDH. When the file is received or the receiver's mailbox is queried, Expedite Base shows this value in the RECEIVED or AVAILABLE record in the SENDERLOC parameter. By default, Expedite Base uses the file location on the sender's system. Use 1 to 65 characters.

**Notes:**

- 1) Expedite Base/400 users who want to implement the DESTFILE and DESTLOC feature must be using an Expedite Base Version 4 Release 3 or later product on both the sending and receiving systems.
- 2) Expedite Base/400 does not support the ORIGFILE parameter on the RECEIVE or RECEIVEEDI commands. Users on this system must allocate the receive file before running Expedite, and must specify the file name on the RECEIVE or RECEIVEEDI commands.

**selectrcv**

Indicates whether the file should be marked in Information Exchange so that the file cannot be received by a blanket receive.

- n Allows the file to be received by a blanket receive.
- f Marks the file in Information Exchange so that the file cannot be received by a blanket receive.
- blank Allows the file to be received by a blanket receive. This is the default.

When **f** is selected in `SELECTRCV`, the file being sent is marked in Information Exchange and cannot be received by a blanket receive. The file can only be received if the receiver specifies *one* of the following:

- Sender ID (account ID and user ID)
- Message class
- Message key

## SENDSTREAM command

Use the SENDSTREAM command to send data from the message command file (INMSG) to Information Exchange. You can place the data immediately after the SENDSTREAM command in the command file. This command is valid only with session-level recovery.

### Syntax

#### sendstream

**alias**(*alias*) **aliasname**(*alias name*)

or

**sysid**(*system ID*) **account**(*account*) **userid**(*user ID*)

or

**account**(*account*) **userid**(*user ID*)

or

**listname**(*list name*)

**endstr**(*end string*) datatype(*e|b*) mode(*blank|t*)

priority(*blank|i|p*) charge(*1|2|3|4|5|6*) ack(*blank|a|b|c|d|e|f|r*)

msgname(*message name*) msgseqno(*message sequence no*) class(*class*)

verify(*n|y|f*) description(*description*) retain(*retain time*)

compress(*n|y|t*)

selectrev(*n|f|blank*);

### Example

```
sendstream listname(list3) class(testclas) endstr(xxx);
```

**Results:** This data is sent to everyone who is defined in list3.

### Parameters

#### alias

Table type and table name of an alias table.

**blank** An alias name was not used. This is the default.

**gxxx** Global alias table, where *xxx* identifies a 1- to 3-character table name.

**oxxx** Organizational alias table, where *xxx* identifies a 1- to 3-character table name.

**pxxx** Private alias table, where *xxx* identifies a 1- to 3-character table name.



**NOTE:** You create and maintain alias tables by using Information Exchange Administration Services (see *Information Exchange Administration Services User's Guide*), or by using the DEFINEALIAS command (see "DEFINEALIAS command" on page 94).

#### aliasname

Alias name defined in the alias table. Use 1 to 16 alphanumeric characters.

#### sysid

System ID of a single destination user ID. You need the system ID only if you specify the ACCOUNT and USERID parameters for a user on another Information Exchange system. If you specify a SYSID parameter, you must specify the ACCOUNT and USERID parameters. Use 1 to 3 alphanumeric characters.

**account**

Account ID of a single destination user ID to whom you are sending data. Expedite Base/400 uses this field together with the `USERID` parameter to identify the user. If you specify an `ACCOUNT` parameter, you must also specify a `USERID` parameter. Use 1 to 8 alphanumeric characters.

**userid**

User ID of an Information Exchange user to whom you are sending data. Expedite Base/400 uses this field together with the `ACCOUNT` parameter to identify the user. If you specify a `USERID` parameter, you must also specify an `ACCOUNT` parameter. Use 1 to 8 alphanumeric characters.

**listname**

Name of a previously defined list of account IDs and user IDs. Use 1 to 8 alphanumeric characters.

**endstr**

Character string used to mark the end of the data contained in the command stream. You can split the end string across record boundaries. Use 1 to 79 alphanumeric characters. This parameter is required.

**datatype**

Indicates whether the data is EBCDIC or binary. ASCII systems translate EBCDIC data to ASCII when they receive it from Information Exchange. For more information on EBCDIC and binary data, see Chapter 3, “Communicating with other operating systems.”

- e EBCDIC data. This is the default.
- b Binary data.

**mode**

Indicates whether the file is a test file or a normal file.

- blank Normal file. This is the default.
- t Test-mode file.

**priority**

Class of delivery service for this data.

- blank Normal priority. This is the default.
- i Express delivery to those users who have the continuous receive capability and are currently in session with Information Exchange. (Expedite Base/400 does not support the continuous receive capability.)
- p High priority.

**charge**

Indicates how you would like the charges paid.

- 1 Receiver pays all charges.
- 2 Receiver pays all charges if agreed to. Otherwise, the sender and receiver split the charges.

- 3 Receiver pays all charges if agreed to; otherwise, the sender and receiver split the charges, if agreed to by the receiver. Otherwise, the sender pays all charges. This is the default.
- 4 Sender and receiver split the charges, if agreed to by the receiver. Otherwise, the sender pays all charges.
- 5 Sender and receiver split the charges.
- 6 Sender pays all charges.

**ack**

Type of acknowledgments you want to receive from Information Exchange.

- blank No acknowledgments are created. This is the default.
- a Purge acknowledgments only.
- b Both receipt and delivery acknowledgments.
- c Both receipt and purge acknowledgments.
- d Delivery acknowledgments only. It queues the delivery acknowledgments to the user ID when Information Exchange delivers the data to a destination user ID and reaches a subsequent recovery point with that destination user ID.
- e Either a purge or a delivery acknowledgment.
- f Receipt and either a purge or a delivery acknowledgment.
- r Receipt acknowledgments only when Expedite Base/400 finishes sending a file.

**msgname**

User-specified name for the data. Use 1 to 8 alphanumeric characters.

**msgseqno**

User-specified message control number for the data. Use 1 to 5 alphanumeric characters.

**class**

Unique, user-specified name for the data. A receiver can then use this name to receive only files of this class. Use 1 to 8 alphanumeric characters.

**verify**

Verifies that the receiver exists before sending the data.

- n Do not verify that the receiver exists. This is the default.
- y Verify that the receiver exists before sending the data.
- f Verify that the receiver exists. If Information Exchange cannot tell whether the receiver exists (for example, if the receiver is on another system), send the data anyway.

If you request verification, you might incur an Information Exchange message charge. See “Validating addresses, payment levels, and authorizations” on page 67 for more information.

**description**

Free-format description of the data sent. Use 1 to 79 alphanumeric characters.



**retain**

Number of days Information Exchange keeps the file if the receiver does not receive it. Valid values are **blank** and **0** through **180**.

The maximum retention period and the default retention period can be different on different Information Exchange systems. These periods are system-dependent. The default retention period is determined when Information Exchange is installed. For installations in the U.S., the default retention period is 30 days. Contact your marketing representative for more information on these values.

If you specify **0** or **blank**, Information Exchange retains the file for the default retention period. If you specify a retention period that is longer than the maximum retention period for your system, Information Exchange retains the file for the default retention period.

**compress**

Indicates whether the specified file should be compressed.

- n** Do not compress the specified file. This is the default.
- y** Compress the specified file.
- t** For each sender/receiver pair, use the setting of the COMPRESS parameter (*y* or *n*) indicated in the CLOOKUP table.  
  
If the receiver/sender pair is not in the CLOOKUP table, send the data, and write a warning to the response file, OUTMSG.

You must have the supported compression software installed to use data compression.

Some of the parameters of this command may work differently when sending compressed data; see Appendix D, “Using data compression,” for more information.

**selectrcv**

Indicates whether the file should be marked in Information Exchange so that the file cannot be received by a blanket receive.

- n** Allows the file to be received by a blanket receive.
- f** Marks the file in Information Exchange so that the file cannot be received by a blanket receive.
- blank** Allows the file to be received by a blanket receive. This is the default.

When **f** is selected in SELECTRCV, the file being sent is marked in Information Exchange and cannot be received by a blanket receive. The file can only be received if the receiver specifies *one* of the following:

- Sender ID (account ID and user ID)
- Message class
- Message key

## SESSIONINFO command

Use the SESSIONINFO command to obtain information about your Information Exchange session. Expedite Base/400 places a SESSIONRESP record in the response file as a result of the SESSIONINFO command. It contains Information Exchange session information.

The SESSIONINFO command has no parameters.

### Syntax

**sessioninfo;**

## START command

Use the START command to begin an Information Exchange session. If Information Exchange successfully starts the session, Expedite Base/400 returns a session start key in the RETURN response record.

Three new parameters are added to the START command: KEYRINGFILE, KEYRINGPASSWORD, and APPLICID. These new parameters are only used if SSL is enabled on the new SSL command, and you specify AUTOSTART(N). If you use the APPLICID parameter, the KEYRINGFILE and KEYRINGPASSWORD parameters are ignored. Using the KEYRINGFILE parameter allows the user to specify the name (and path) of the KDB file that contains the user's public and private keys. Using the KEYRINGPASSWORD parameter allows the user to specify the KDB file password. Using the APPLICID allows the user to specify an application ID as an alternate tag into the default \*SYSTEM KDB file.

### Syntax

#### start

```
account(account) userid(user ID)
iepassword(IE password) niepassword(new IE password) check (y|n)

keyringfile(KDB file name) keyringpassword(key ring password)
OR
applicid(application ID);
```



**NOTE:** If the AUTOSTART value in the profile is y, issuing a START command results in an error condition.

### Example

```
start account(acct) userid(user01) iepassword(mypswd) keyringfile (keyfile.kdb)
keyringpassword(mykeypswd);
```

### Parameters

#### account

Account of an Information Exchange user to whom you are sending data. Expedite Base/400 uses this field together with the USERID parameter to identify the user. If you specify an ACCOUNT parameter, you must also specify a USERID parameter. Use 1 to 8 alphanumeric characters. The default is the IEACCOUNT value used in the IDENTIFY command.

#### userid

User ID of an Information Exchange user to whom you are sending data. Expedite Base/400 uses this field together with the ACCOUNT parameter to identify the user. If you specify a USERID parameter, you must also specify an ACCOUNT parameter. Use 1 to 8 alphanumeric characters. The default is the IEUSERID value used in the IDENTIFY command.

#### iepassword

Information Exchange password. Use 1 to 8 alphanumeric characters. The default is the IEPASSWORD value used in the IDENTIFY command.

**niepassword**

New Information Exchange password which will become effective at the next successful Information Exchange session end. Use 1 to 8 alphanumeric characters. The default is the NIEPASSWORD value used in the IDENTIFY command.



**NOTE:** Remove this parameter and change the IEPASSWORD parameter after you change your password.

**check**

Determines whether the status of the previous session is checked.

- y Check the status of the previous session. Do not specify any other commands except the END command in the input file (INMSG). The status of the previous session will be indicated on the STARTED response record.
- n Do not check the status of the previous session; start a session as usual. This is the default.

**keyringfile**

The name (and path) of the kdb file that contains the certificate. This applies to TCP/IP communication performed with SSL enabled. The keyringpassword is required with this parameter. This parameter is case sensitive. If you use this parameter, do not use the applicid parameter.

**keyringpassword**

The kdb file password. This value is not case sensitive. It can be from 1 to 128 characters in length. This parameter is case sensitive. If you use this parameter, do not use applicid.

**applicid**

The name of the application that you defined in the Digital Certificate Manager (DCM) application database, which is tied to a specific certificate. If you use this parameter, do not use the keyringfile and keyringpassword parameters.

The valid values are whatever are valid when you defined the application ID in DCM and it can be from 1 to 100 characters in length.

## TESTMSG command

The TESTMSG command tells Information Exchange to load one or more test messages into the mailbox. Test messages are from account \*SYSTEM\* and user ID \*TSTMSG\*. You must specify this account and user ID with the RECEIVE command to receive the test messages. If you do not specify the account and user ID, you cannot receive the test messages.

### Syntax

#### **testmsg**

```
startmsg(0|1|2|3|4|5) endmsg(0|1|2|3|4|5);
```

### Example

```
testmsg startmsg(0) endmsg(5);
```

### Parameters

#### **startmsg**

Number of the first test message that Expedite Base/400 must place in the mailbox. This parameter is required.

- 0 Return the numeric characters 0 through 9 in the test message.
- 1 Return the uppercase alphabetic characters in the test message.
- 2 Return the lowercase alphabetic characters in the test message.
- 3 Return the numeric characters followed by uppercase alphabetic characters, then lowercase alphabetic characters in the test message.
- 4 Return the special characters in the test message.
- 5 Return all 256 characters in ascending sequence.

#### **endmsg**

Number of the last test message Expedite Base/400 must place in the mailbox. Valid numbers are described under the STARTMSG parameter. The ENDMSG must be greater than or equal to the STARTMSG. If STARTMSG and ENDMSG are the same, Expedite Base/400 returns a single message. This parameter is required.



## Expedite Base/400 response records

---

A response file consists of echoed commands that Expedite Base/400 has processed, followed by response records associated with those commands. Expedite Base/400 provides you with two primary response files (OUTPRO and OUTMSG) that correspond to the two command files (INPRO and INMSG). If you are using checkpoint-level recovery, OUTWORK contains the commands and response records that were processed since the last checkpoint.

### OUTPRO response records

The response records contained in the OUTPRO file are listed here with the page numbers on which their detailed descriptions appear:

- PROFILERC, page 150  
Indicates the completion of an entire INPRO file.
- RETURN, page 151  
Indicates the completion of a command in INPRO.
- WARNING, page 152  
Indicates a minor problem that did not stop the command from finishing, but that should be noted.

## PROFILERC record

The PROFILERC record is the last record in OUTPRO. The PROFILERC record indicates that the profile commands have completed processing. A zero value indicates that all the profile commands to build or alter the profile completed normally.

### Format

PROFILERC(*return code*) ERRDESC(*error description*);



**NOTE:** There is only one PROFILERC record in each OUTPRO.

### Parameters

#### **RETCODE**

Indicates whether the profile commands processed successfully. If the return code is zero, the processing completed normally. If the return code is not zero, Expedite Base/400 displays an error code, and you may see an ERRDESC value. Expedite Base/400 displays the same return code on the SESSIONEND record in the OUTMSG file. This parameter contains 5 numeric characters.

#### **ERRDESC**

Short description of the error condition. If the return code is zero, Expedite Base/400 does not display this parameter. This parameter contains 1 to 80 alphanumeric characters.



## RETURN record

The RETURN record indicates the completion of a command in INPRO or INMSG. A zero value indicates that the command completed normally.

### Format

RETURN(*return*) ERRDESC(*error description*)  
REASON(*reason*) ERRTXT(*error text*);

### Parameters

#### **RETURN**

Indicates whether a command completed successfully. If the return code is zero, the command completed normally. If the return code is not zero, Expedite Base/400 displays an error number and may also display an ERRDESC, REASON, and ERRTXT value. This parameter contains 5 numeric characters.

#### **ERRDESC**

Short description of the error condition. If the return code is zero, Expedite Base/400 does not display this parameter. This parameter contains 1 to 80 alphanumeric characters.

#### **REASON**

Cause of the error. If Expedite Base/400 includes REASON in the record, it gives more information about what caused the error. Expedite Base/400 usually includes REASON when an error occurs opening a file. This parameter contains 1 to 240 alphanumeric characters.

#### **ERRTEXT**

Detailed description of an error which may suggest steps to correct the problem. Expedite Base/400 may include this parameter more than once in a record if the error text consists of multiple lines. This parameter contains 1 to 80 alphanumeric characters.

## WARNING record

The WARNING record indicates a low-severity problem occurred which did not prevent the profile command from completing.

### Format

WARNING(*warning*)  
ERRDESC(*error description*);

### Parameters

#### **WARNING**

Warning code. This parameter contains 5 numeric characters.

#### **ERRDESC**

Description of the problem. This parameter contains 1 to 80 alphanumeric characters.

## OUTMSG response records

Response records contained in the OUTMSG file are listed here with the page numbers on which their detailed descriptions appear:

- **AUTOEND**, page 155  
Indicates that an Information Exchange session ended automatically.
- **AUTOSTART**, page 156  
Indicates that an Information Exchange session started automatically.
- **AVAILABLE**, page 157  
Provides information about messages in your Information Exchange mailbox in response to a QUERY command.
- **LIBRARYLIST**, page 162  
Provides the results of the LISTLIBRARIES command.
- **MEMBERLIST**, page 164  
Provides the results of the LISTMEMBERS command.
- **MEMBERPUT**, page 166  
Returns information about the member placed in the library.
- **MOVED**, page 167  
Tells you how many files Information Exchange moved from the archive as a result of an ARCHIVEMOVE command.
- **MSGRESP**, page 168  
Contains information summarizing your Information Exchange mailbox contents. This record is produced in response to the MSGINFO command.
- **NOTSENT**, page 170  
Provides information on the data that could not be sent when a destination verification failure occurred with the SENDEDI command.
- **RECEIVED**, page 172  
Provides information on the data received with a RECEIVE, RECEIVEEDI, or RECEIVESTREAM command.
- **RETURN**, page 178  
Indicates the completion of a command in INMSG.
- **SENT**, page 179  
Provides information on the data sent with a SEND, SENDEDI, or SENDSTREAM command.
- **SESSIONEND**, page 181  
Indicates the return code for the last time you ran Expedite Base/400.

- **SESSIONRESP**, page 182  
Contains information summarizing the current Information Exchange session. This record is produced in response to the **SESSIONINFO** command.
- **STARTED**, page 185  
Contains information about the current Information Exchange session and the prior Information Exchange session.
- **WARNING**, page 186  
Indicates a minor problem that did not stop the command from finishing, but that should be noted.

## AUTOEND record

The AUTOEND record indicates that Expedite Base/400 automatically ended an Information Exchange session.

### Format

AUTOEND;

## AUTOSTART record

The AUTOSTART record indicates that Expedite Base/400 started an Information Exchange session automatically.

### Format

AUTOSTART SESSIONKEY(*session key*);

### Parameters

#### **SESSIONKEY**

Session access key provided by Information Exchange upon a successful session start. If the session start is not successful, Expedite Base/400 does not put the session key in the record.

## AVAILABLE record

Expedite Base/400 produces an AVAILABLE record for every message in the mailbox when you use the QUERY command. Each record contains information describing the message. Note that because parameters with blank values do not print, and because some parameters come from the CDH, all of the parameters described below may not be included with every AVAILABLE record. If you specify CDH(y) for a message or file that has no common data header, the CDH parameters adopt default values.

### Format

AVAILABLE

SYSID(*system ID*) ACCOUNT(*account*) USERID(*user ID*)

or

ACCOUNT(*account*) USERID(*user ID*)

MSGKEY(*message key*) CLASS(*class*) MODE(*t*) LENGTH(*length*)

MSGDATE(*message date*) MSGDATELONG(*message date long format*)

MSGTIME(*message time*) MSGNAME(*message name*)

MSGSEQNO(*message sequence no*)

SYSNAME(*system name*) SYSLEVEL(*system level*)

DATATYPE(*e|b*) EDITYPE(*EDI type*)

SENDERFILE(*sender file*) SENDERLOC(*sender location*)

FILEDATE(*file date*) FILEDATELONG(*file date long format*) FILETIME(*file time*)

RECFM(*record format*) RECLen(*record length*) RECDLM(*c|e|l|n*)

DESCRIPTION(*description*) UNIQUEID(*unique ID*) CODEPAGE(*code page*)

SYSTYPE(*01|10|11|12|14|15|16|17|19|20|21|22|30|31|33|40|44|61|62|71|80|90|91*)

SYSVER(*system version*) TRANSLATE(*xlate table*)

COMSW(*compression software name*) COMVER(*compression software version*)

COMFILE(*name of compressed file*);

### Parameters

The following parameters are not shown if you specify CDH(n) using the QUERY command:

CODEPAGE	COMFILE	COMSW	COMVER	DATATYPE
DESCRIPTION	EDITYPE	FILEDATE	FILETIME	RECDLM
RECFM	RECLen	SENDERFILE	SENDERLOC	SYSTYPE
SYSVER	TRANSLATE	UNIQUEID		

### SYSID

System ID of the user who sent the message. This parameter contains 1 to 3 alphanumeric characters.

### ACCOUNT

Account name of the user who sent the message. This parameter contains 1 to 8 alphanumeric characters.

### USERID

User ID of the user who sent the message. This parameter contains 1 to 8 alphanumeric characters.

**MSGKEY**

Unique identifier assigned to the message or file by Information Exchange. You can use this value for the MSGKEY parameter of the RECEIVE, RECEIVEEDI, or RECEIVESTREAM commands to receive only a specific message or file. This parameter contains 20 hexadecimal characters.

**CLASS**

User class of the data, specified by the sender. This parameter contains 1 to 8 alphanumeric characters.

**MODE**

Network data class for this data. The sender specified this value. If the file or message is not test mode (indicated by the value *T*), this parameter is omitted from the record.

**LENGTH**

Length of the data in the Information Exchange mailbox. The length of the file or message received may be different from the length of the file sent because of reformatting. This parameter contains 1 to 10 numeric characters.

**MSGDATE**

Date the message was placed into Information Exchange. The format is *YYMMDD*.

**MSGDATELONG**

Long format of MSGDATE. The format is *YYYYMMDD*. The *YY* value from MSGDATE is interpreted by Expedite Base/400 as *YYYY*.

**MSGTIME**

Time the message was placed into Information Exchange. The format is *HHMMSS*.

**MSGNAME**

Name of the message, specified by the sender. This parameter contains 1 to 8 alphanumeric characters.

**MSGSEQNO**

Sequence number assigned by the sender to specify a message control number for this data. This parameter contains 1 to 5 alphanumeric characters.

**SYSNAME**

Name of the system that sent the data. This parameter contains 1 to 8 alphanumeric characters.

**SYSLEVEL**

Level of the system that sent the data. This parameter contains 1 to 4 alphanumeric characters.

**DATATYPE**

Indicates whether the data is EBCDIC or binary.

e EBCDIC

b Binary

**EDITYPE**

Types of EDI data available: X12, UCS, UNTDI, EDIFACT, or unformatted. This parameter contains 3 to 11 alphanumeric characters.



**SENDERFILE**

File name the data had on the sender's system. This parameter contains 1 to 54 alphanumeric characters.

**SENDERLOC**

Location of the file on the sender's system. This parameter contains 1 to 65 alphanumeric characters.

**FILEDATE**

Date of the file on the sender's system. This parameter contains 1 to 6 alphanumeric characters. The format is *YYMMDD*.

**FILEDATELONG**

Long format of FILEDATE. The format is *YYYYMMDD*. The *YY* value from FILEDATE is interpreted by Expedite Base/400 as *YYYY*.

**FILETIME**

Time of the file on the sender's system. This parameter contains 1 to 6 alphanumeric characters. The format is *HHMMSS*.

**RECFM**

Record format of the file on the sender's system. If the record format is not appropriate for the sending machine (for example, if the sending machine is a PC), the value is *????*. This parameter contains 1 to 4 alphanumeric characters.

**RECLEN**

Record length of the file on the sender's system. This parameter contains 1 to 5 numeric characters.

**RECDLM**

Indicates the method is used to delimit the records.

- c CRLF characters delimit the records.
- e EDI characters delimit the records.
- l 2-byte delimiter precedes each record.
- n Either the records have no delimiters, or the CDH does not indicate the type of delimiter.

**DESCRIPTION**

Free-format description of the data given by the sender. This parameter contains 1 to 79 alphanumeric characters.

**UNIQUEID**

Random ID assigned to the message by the sending interface. It can help you identify the message and also help you match acknowledgments to the message sent. This parameter contains 8 alphanumeric characters.

**CODEPAGE**

Code page used by the sending system to determine the character representation of the data. This parameter contains 3 numeric characters.

**SYSTYPE**

Type of system that sent the data. The codes for the systems are:

01	Unknown system type
10	expEDItE/PC
11	Expedite Base/2
12	Expedite Base for AIX
14	Expedite Base for SCO UNIX
15	Expedite Base/DOS
16	Expedite Base for SCO XENIX
17	Expedite Base for Windows
19	Expedite for Windows
20	expEDItE/MVS Host
21	Expedite Base/MVS
22	Information Exchange File Transfer Protocol
30	Mail Exchange
31	Expedite Base/VM
33	X.400 Gateway
40	Expedite/Direct
44	EDI VAN Interconnect
61	EDI Server
62	IBM Exchange for Websphere MQ
71	Expedite Base/400
80	Expedite/CICS
90	Information Exchange Administration Services
91	Expedite/Async

This parameter contains 2 hexadecimal digits.

**SYSVER**

Software version of the system sending the data. This parameter contains 1 numeric character.

**TRANSLATE**

ASCII-to-EBCDIC translate table used to send this file to Information Exchange. This parameter contains 1 to 8 alphanumeric characters.

**COMSW**

Name of the software compression package used to compress the file. This parameter contains 10 alphanumeric characters.

**COMVER**

Version of the compression software package used to compress the file. This parameter contains 1 to 5 alphanumeric characters.

**COMFILE**

Name of the compressed file. This parameter contains 1 to 54 alphanumeric characters.

## LIBRARYLIST record

The LIBRARYLIST record returns information requested by the LISTLIBRARIES command. Each record contains information specific to that library. Parameters with blank values are not included.



**NOTE:** No LIBRARYLIST or other response records are written to the message response file if no libraries are found that match the criteria you specified.

### Format

#### LIBRARYLIST

OWNER(*library owner account*) OWNUSERID(*owner's user ID*)  
 LIBRARY(*library name*) MEMBERS(*number of members*) DESCRIPTION(*description*)  
 CREATEDATE(*creation date*) CREATEDATELONG(*creation date long format*)  
 CREATETIME(*creation time*)  
 UPDATEDBY(*ID of last user*) UPDATEDATE(*date of last update*)  
 UPDATEDATELONG(*date of last update long format*) UPDATETIME(*time of last update*)  
 WRITEAUTH(*p|o|g|l*) WRITELIST(*write distribution list*)  
 READAUTH(*p|o|g|l*) READLIST(*read distribution list*)  
 SEARCHABLE(*y|n*) OWNERPAYS(*y|n*);

### Parameters

#### **OWNER**

Account that owns the library. This parameter contains 1 to 8 alphanumeric characters.

#### **OWNUSERID**

User ID of the owner of the library.

#### **LIBRARY**

Name of the library. This parameter contains 1 to 8 alphanumeric characters.

#### **MEMBERS**

Number of members present in the library.

#### **DESCRIPTION**

Free-format description of the library. This parameter contains 1 to 79 alphanumeric characters.

#### **CREATEDATE**

Date the library was created. The format is *YYMMDD*. Information Exchange adjusts the date to that of your local time zone.

#### **CREATEDATELONG**

Long format of CREATEDATE. The format is *YYYYMMDD*. The *YY* value from CREATEDATE is interpreted by Expedite Base/400 as *YYYY*.

#### **CREATETIME**

Time the library was created. The format is *HHMMSS*. Information Exchange adjusts the time to that of your local time zone.

#### **UPDATEDBY**

Account and user ID, separated by at least one blank, of the user who last redefined this library.

**UPDATEDATE**

Date the library was last redefined. The format is *YYMMDD*. Information Exchange adjusts the date to that of your local time zone.

**UPDATEDATELONG**

Long format of **UPDATEDATE**. The format is *YYYYMMDD*. The *YY* value from **UPDATEDATE** is interpreted by Expedite Base/400 as *YYYY*.

**UPDATETIME**

Time the library was last redefined. The format is *HHMMSS*. Information Exchange adjusts the time to that of your local time zone.

**WRITEAUTH**

Authority type for update access to the library.

- p Only the owner can update this library.
- o Only users within the same account can update this library.
- g Any user can update this library.
- l Any user in the list named in the **WRITELIST** parameter can update this library.

**WRITELIST**

Name of a permanent distribution list that details the users who can update this library.

**READAUTH**

Authority type for read access to the library.

- p Only the owner can read this library.
- o Only users within the same account can read this library.
- g Any user can read this library.
- l Any user in the list named in the **READLIST** parameter can read this library.

**READLIST**

Name of a permanent distribution list that details the users who can read this library.

**SEARCHABLE**

Indicates whether the library can be searched.

- y The library is searchable.
- n The library is not searchable.

**OWNERPAYS**

Indicates whether the owner of the library is responsible for charges associated with transferring the library member from the user's mailbox to the user's system when the library member is retrieved. These are usually called *receive-side* charges.

- y The owner of the library pays the receive-side charges.
- n The receive-side charges will be charged to you.

## MEMBERLIST record

The MEMBERLIST record returns information requested by the LISTMEMBERS command. Each record contains information specific to that member. Parameters with blank values are not included.



**NOTE:** No MEMBERLIST or other response records are written to the message response file if the user has no authority to access the library.

### Format

```
MEMBERLIST
MEMBER(member name) DESCRIPTION(description)
CREATEDBY(user ID) CREATEDATE(creation date)
CREATEDATELONG(creation date long format) CREATETIME(creation time)
UPDATEDBY(ID of last user) UPDATEDATE(date of last update)
UPDATEDATELONG(date of last update long format) UPDATETIME(time of last update)
LENGTH(file length);
```

### Parameters

#### MEMBER

Name of a library member. This parameter contains 1 to 8 alphanumeric characters.

#### DESCRIPTION

Free-format description of the member. This parameter contains 1 to 79 numeric characters.

#### CREATEDBY

System ID, account ID, and user ID of the user that created this library member. The first 4 characters are the system ID (and may be blank), the next 8 characters are the account ID, and the last 8 characters are the user ID.

#### CREATEDATE

Date the member was created. The format is *YYMMDD*. Information Exchange adjusts the time to that of your local time zone.

#### CREATEDATELONG

Long format of CREATEDATE. The format is *YYYYMMDD*. The *YY* value from CREATEDATE is interpreted by Expedite Base/400 as *YYYY*.

#### CREATETIME

Time the member was created. The format is *HHMMSS*. Information Exchange adjusts the time to that of your local time zone.

#### UPDATEDBY

System ID, account ID, and user ID of the user that last updated this library member. The first 4 characters are the system ID (and may be blank), the next 8 characters are the account ID, and the last 8 characters are the user ID.

#### UPDATEDATE

Date the member was last updated. The format is *YYMMDD*. Information Exchange adjusts the date to that of your local time zone.

#### UPDATEDATELONG

Long format of UPDATEDATE. The format is *YYYYMMDD*. The *YY* value from UPDATEDATE is interpreted by Expedite Base/400 as *YYYY*.

**UPDATETIME**

Time the member was last updated. The format is *HHMMSS*. Information Exchange adjusts the time to that of your local time zone.

**LENGTH**

Length of the member in bytes.

## MEMBERPUT record

The MEMBERPUT record returns information about the member placed in the library. Since parameters with blank values are not written, all of the parameters indicated below may not appear with the MEMBERPUT record.

### Format

```
MEMBERPUT  
OWNER(owner) LIBRARY(library) MEMBER(member)  
UNIQUEID(uniqueid) LENGTH(length);
```

### Parameters

#### **OWNER**

Account that owns the library. This parameter contains 1 to 8 alphanumeric characters.

#### **LIBRARY**

Name of the library. This parameter contains 1 to 8 alphanumeric characters.

#### **MEMBER**

Name of the member.

#### **UNIQUEID**

Random ID assigned to the member by Expedite Base/400. It helps to identify the member. This parameter contains 8 alphanumeric characters.

#### **LENGTH**

Length of the file sent. This parameter contains 1 to 8 numeric characters.



## MOVED record

The MOVED record tells how many files Information Exchange moved from short-term archive to your Information Exchange mailbox as a result of an ARCHIVEMOVE command.

### Format

MOVED  
NUMBER(*number*);

### Parameters

#### **NUMBER**

Number of files moved from short-term archive to your Information Exchange mailbox as a result of an ARCHIVEMOVE command. If this value is zero, no files in the archive match the ARCHIVEID specified on the ARCHIVEMOVE command, or the files have already been retrieved and still reside in your mailbox. This parameter contains 1 to 5 numeric characters.

## MSGRESP record

Expedite Base/400 produces a MSGRESP record in response to the MSGINFO command. It contains information summarizing your Information Exchange mailbox contents. This record is found only in OUTMSG.

### Format

MSGRESP

DATE(*date*) DATELONG(*date long format*)

TIME(*time*) DATAMSGS(*data messages*)

DATADATE(*data date*) DATADATELONG(*data date long format*)

DATATIME(*data time*) SERVMSGs(*service messages*)

SERVDATE(*service date*) SERVDATELONG(*service date long format*)

SERVTIME(*service time*) TOTALDATA(*total data*) MAXMSG(*maximum message*);

### Parameters

#### DATE

Date that Information Exchange generated the summary information. The format is *YYMMDD*.

#### DATELONG

Long format of DATE. The format is *YYYYMMDD*. The *YY* value from DATE is interpreted by Expedite Base/400 as *YYYY*.

#### TIME

Time that Information Exchange generated the summary information. The format is *HHMMSS*.

#### DATAMSGS

Number of Information Exchange segments available as actual data. This parameter contains 1 to 6 characters.

#### DATADATE

Date of oldest data message available. The format is *YYMMDD*.

#### DATADATELONG

Long format of DATADATE. The format is *YYYYMMDD*. The *YY* value from DATADATE is interpreted by Expedite Base/400 as *YYYY*.

#### DATATIME

Time of the oldest data message available. The format is *HHMMSS*.

#### SERVMSGs

Number of Information Exchange segments available as service messages from Information Exchange. This parameter contains 1 to 6 numeric characters.

#### SERVDATE

Date of the oldest Information Exchange service message available. The format is *HHMMSS*.

#### SERVDATELONG

Long format of SERVDATE. The format is *YYYYMMDD*. The *YY* value from SERVDATE is interpreted by Expedite Base/400 as *YYYY*.

**SERVTIME**

Time of the oldest Information Exchange service message available. The format is *HHMMSS*.

**TOTALDATA**

Total number of Information Exchange segments available. It includes both service messages from Information Exchange and actual data messages. This parameter contains 1 to 6 numeric characters.

**MAXMSG**

Total number of Information Exchange segments contained in the largest Information Exchange message or message group in the mailbox. This parameter contains 1 to 6 numeric characters.

## NOTSENT record

Expedite Base/400 produces a NOTSENT record for every EDI envelope that could not be sent because of a destination verification failure. NOTSENT records are produced only when the VERIFY parameter of the SENDEDI command is set to *c* or *g*.

Because parameters with blank values do not print, Expedite Base/400 might not include all the parameters listed below with each NOTSENT record.

### Format

NOTSENT

ALIAS(*alias*) ALIASNAME(*alias name*)

or

SYSID(*system ID*) ACCOUNT(*account*) USERID(*user ID*)

or

ACCOUNT(*account*) USERID(*user ID*)

or

LISTNAME(*list name*)

EDITYPE(*datatype*) DESTINATION(*destination*) QUALIFIER(*qualifier*)

CONTROLNUM(*control number*) CLASS(*class*)

MSGNAME(*message name*) MSGSEQNO(*message sequence no*);

### Parameters

#### ALIAS

Alias table type and table name of the Information Exchange destination.

*gxxx* Global alias table, where *xxx* identifies a 1- to 3-character table name.

*oxxx* Organizational alias table, where *xxx* identifies a 1- to 3-character table name.

*pxxx* Private alias table, where *xxx* identifies a 1- to 3-character table name.

This parameter contains 1 to 4 characters.

#### ALIASNAME

Alias name defined in the alias table. This parameter contains 1 to 16 alphanumeric characters.

#### SYSID

System ID of the Information Exchange destination. This parameter contains 1 to 3 alphanumeric characters.

#### ACCOUNT

Account of the Information Exchange destination. This parameter contains 1 to 8 alphanumeric characters.

#### USERID

User ID of the Information Exchange destination. This parameter contains 1 to 8 alphanumeric characters.

**LISTNAME**

Name of a previously defined list of account and user IDs used as the Information Exchange destination. This parameter contains 1 to 8 alphanumeric characters.

**EDITYPE**

Type of EDI data that would have been sent, had the destination verification failure not occurred: X12, UCS, UN/TDI, or EDIFACT. This parameter contains 1 to 7 alphanumeric characters.

**DESTINATION**

Destination specified in the EDI data. This parameter contains 1 to 35 alphanumeric characters.

**QUALIFIER**

EDI qualifier for the destination. This parameter contains 1 to 4 alphanumeric characters.

**CONTROLNUM**

Interchange Control Number from the X12 or UCS data. For UN/TDI data, this is the SNRF element. For EDIFACT data, it is the data element 0020 (Interchange Control Reference). This parameter contains 1 to 14 alphanumeric characters.

**CLASS**

User class obtained from the EDI data, the CLASS parameter, or the default. This parameter contains 1 to 8 alphanumeric characters.

**MSGNAME**

Message name obtained from the EDI data or the MSGNAME parameter. This parameter contains 1 to 8 alphanumeric characters.

**MSGSEQNO**

Message sequence number obtained from the MSGSEQNO parameter or generated by Expedite Base/400. This parameter contains 1 to 5 alphanumeric characters.

## RECEIVED record

Expedite Base/400 produces a RECEIVED record for every file or EDI envelope received from Information Exchange. This record contains information describing the received data.

### Format

RECEIVED

ALIAS(alias) ALIASNAME(alias name)

or

SYSID(system ID) ACCOUNT(account) USERID(user ID)

or

ACCOUNT(account) USERID(user ID)

RECEIVER(receiver ID) RECVQUAL(receiver qualifier)

SENDER(sender ID) SENDQUAL(sender qualifier)

CONTROLNUM(control number) CLASS(class) MODE(t) PRIORITY(a|i|p)

CHARGE(1|5|6) ACK(d) LENGTH(x) FILEID(file ID)

MSGDATE(message date) MSGDATELONG(message date long format)

MSGTIME(message time)

MSGNAME(message name) MSGSEQNO(message sequence no)

SESSIONKEY(session key) DELIMITED(c|l|e|n)

SYSNAME(system name) SYSLEVEL(system level)

STARTDATE(starting date) STARTTIME(starting time)

ENDDATE(ending date) ENDTIME(ending time)

TIMEZONE(l|g) DATATYPE(e|b) EDITYPE(EDI type)

SENDERFILE(sender file) SENDERLOC(sender location)

FILEDATE(file date) FILEDATELONG(file date long format) FILETIME(file time)

RECFM(record format) RECLEN(record length) RECDLM(c|e|l|n)

DESCRIPTION(description) UNIQUEID(unique ID) CODEPAGE(code page)

SYSTYPE(01|10|11|12|14|15|16|17|19|20|21|22|30|31|33|40|44|61|62|71|80|90|91)

SYSVER(system version) TRANSLATE(xlate table)

COMSW(compression software name) COMVER(compression software version)

COMFILE(name of compressed file) DCMPRC(decompression return code);

### Parameters

#### ALIAS

Table type and table name of an alias table.

gxxx Global alias table, where xxx identifies a 1- to 3-character table name.

oxxx Organizational alias table, where xxx identifies a 1- to 3-character table name.

pxxx Private alias table, where xxx identifies a 1- to 3-character table name. This parameter contains 1 to 4 alphanumeric characters.

#### ALIASNAME

Alias name defined in the alias table. This parameter contains 1 to 16 alphanumeric characters.

**SYSID**

System ID of the user who sent the message. This parameter contains 1 to 3 alphanumeric characters.

**ACCOUNT**

Account of the user who sent the message. This parameter contains 1 to 8 alphanumeric characters.

**USERID**

User ID of the user who sent the message. This parameter contains 1 to 8 alphanumeric characters.

**RECEIVER**

Receiver ID specified in the EDI data. This parameter contains 1 to 35 characters.

**RECVQUAL**

EDI qualifier for the receiver specified in the EDI data. This parameter contains 1 to 4 characters.

**SENDER**

Sender ID specified in the EDI data. This parameter contains 1 to 35 characters.

**SENDQUAL**

EDI qualifier for the sender specified in the EDI data.

**CONTROLNUM**

Interchange Control Number from the X12 or UCS data. For UN/TDI data, this is the SNRF element. For EDIFACT data, this is the element 0020 (Interchange Control Reference). This parameter contains 1 to 14 characters.

**CLASS**

User class of the data specified by the sender to identify the data. This parameter contains 1 to 8 alphanumeric characters.

**MODE**

Sender-specified network data class field for this data. This parameter does not appear in the record for a message that is not test mode (indicated by the value *T*).

**PRIORITY**

Class of delivery service for this file. This parameter does not appear in the record for a normal-priority message.

- a Normal-priority requeued archive message
- i Express delivery to those users who have the continuous receive capability (Expedite Base/400 does not support the continuous receive capability.)
- p High priority

**CHARGE**

Indicates how the file charges will be paid.

- 1 Receiver pays all charges.
- 5 Sender and receiver split the charges.
- 6 Sender pays all charges.

**ACK**

Indicates whether delivery was successful. If the sender asked Information Exchange to send a delivery acknowledgment, this value is set to *D*. Otherwise, this parameter is omitted.

**LENGTH**

Length of the received file. This parameter contains 1 to 9 alphanumeric characters.

**FILEID**

Name of the file in which Expedite Base/400 placed the data received. This parameter contains 1 to 33 alphanumeric characters.

**MSGDATE**

Date the data received was placed into Information Exchange. The format of this parameter is *YYMMDD*.

**MSGDATELONG**

Long format of MSGDATE. The format is *YYYYMMDD*. The *YY* value from MSGDATE is interpreted by Expedite Base/400 as *YYYY*.

**MSGTIME**

Time the data received was placed into Information Exchange. The format of this parameter is *HHMMSS*.

**MSGSEQO**

Unique number assigned to the data by Information Exchange. This parameter contains 1 to 6 numeric characters.

**MSGNAME**

Name for the data specified by the sender. This parameter contains 1 to 8 alphanumeric characters.

**MSGSEQNO**

Number assigned by the sender as a file control number for this data. This parameter contains 1 to 5 alphanumeric characters.

**SESSIONKEY**

Session access key that Expedite Base/400 used when the file was received. This value is the archive ID for the file if you did not specify an *ARCHIVEID* parameter in the *RECEIVE*, *RECEIVEEDI*, or *RECEIVESTREAM* command. This parameter contains 1 to 8 alphanumeric characters.

**DELIMITED**

Indicates how Expedite Base/400 processed record delimiters when the file was received.

- c Expedite Base/400 split records at carriage-return/line-feed (CRLF) characters.
- l Expedite Base/400 split records according to the 2-byte length delimiters at the beginning of each record.
- e Expedite Base/400 split records according to EDI delimiters. If you specified *EDIOPT(n)*, the records are split according to the length of the data set.
- n Expedite Base/400 stored the data as it was received. The record length of the data depends on the record length of the data set allocated to receive the data, or on the format option.



**SYSNAME**

Name of the system that sent the data. This parameter contains 1 to 8 alphanumeric characters.

**SYSLEVEL**

Level of the system that sent the data. This parameter contains 1 to 4 alphanumeric characters.

**STARTDATE**

Starting date for files received from Information Exchange. The format is *YYMMDD* or *YYYYMMDD*. This parameter contains up to 8 numeric characters.

**STARTTIME**

Starting time for files received from Information Exchange. The format is *HHMMSS*. This parameter contains 6 numeric characters.

**ENDDATE**

Ending date for files received from Information Exchange. The format is *YYMMDD* or *YYYYMMDD*. This parameter contains up to 8 numeric characters. This parameter contains 6 numeric characters.

**ENDTIME**

Ending time for files received from Information Exchange. The format is *HHMMSS*. This parameter contains 6 numeric characters.

**TIMEZONE**

Time zone reference for the *STARTTIME* and *ENDTIME* parameters.

- l Local time, as defined on the *TIMEZONE* parameter of the *IDENTIFY* command.
- g Greenwich mean time (GMT)

**DATATYPE**

Type of data (EBCDIC or binary).

- e EBCDIC
- b Binary

**EDITYPE**

Type of EDI data received (X12, UCS, UNTDI, EDIFACT, or unformatted). This parameter contains 1 to 11 alphanumeric characters.

**SENDERFILE**

File name the data had on the sender's system. This parameter contains 1 to 33 alphanumeric characters.

**SENDERLOC**

Location of the file on the sender's system. This parameter contains 1 to 65 alphanumeric characters.

**FILEDATE**

Date of the file on the sender's system. The format is *YYMMDD*.

**FILEDATELONG**

Indicates the long format of *FILEDATE*. The format is *YYYYMMDD*. The *YY* value from *FILEDATE* is interpreted by Expedite Base/400 as *YYYY*.

**FILETIME**

Time of the file on the sender's system. The format is *HHMMSS*.

**RECFM**

Record format of the file on the sender's system. If the record format is not appropriate for the sending machine (for example, if the sending machine is a PC), the value is *????*. This parameter contains 1 to 4 alphanumeric characters.

**RECLEN**

Record length of the file on the sender's system. This parameter contains 1 to 5 numeric characters.

**RECDLM**

Method used to delimit the records.

- c CRLF characters delimit the records.
- e EDI characters delimit the records.
- l A 2-byte length preceding each record delimits the records.
- n The records contain no delimiters, or the sender did not indicate the type of delimiters.

**DESCRIPTION**

Free-format description of the data given by the sender. This parameter contains 1 to 79 alphanumeric characters.

**UNIQUEID**

Random ID assigned to the data by the sending interface. It helps you identify the data. This parameter contains 8 alphanumeric characters.

**CODEPAGE**

Code page used by the sending system to determine the character representation of a given byte. This parameter contains 3 numeric characters.

**SYSTYPE**

Type of system that sent the data. The codes for the systems are as follows:

- 01 Unknown system type
- 10 expEDItE/PC
- 11 Expedite Base/2
- 12 Expedite Base for AIX
- 14 Expedite Base for SCO UNIX
- 15 Expedite Base/DOS
- 16 Expedite Base for SCO XENIX
- 17 Expedite Base for Windows
- 19 Expedite for Windows
- 20 expEDItE/MVS Host
- 21 Expedite Base/MVS

---

22	Information Exchange File Transfer Protocol
30	Mail Exchange
31	Expedite Base/VM
33	X.400 Gateway
40	Expedite/Direct
44	EDI VAN Interconnect
61	EDI Server
62	IBM Exchange for Websphere MQ
71	Expedite Base/400
80	Expedite/CICS
90	Information Exchange Administration Services
91	Expedite/Async

**SYSVER**

Software version of the system sending the data. This parameter contains 1 to 3 numeric characters.

**TRANSLATE**

ASCII-to-EBCDIC translate table used when this file was sent to Information Exchange. This parameter contains 1 to 8 alphanumeric characters.

**COMSW**

Name of the software compression package used to compress the file. This parameter contains 10 alphanumeric characters.

**COMVER**

Version of the compression software package used to compress the file. This parameter contains 1 to 5 alphanumeric characters.

**COMFILE**

Name of the compressed file. This parameter contains 1 to 54 alphanumeric characters.

**DCMPRC**

Return code from decompression processing. This parameter contains 1 to 5 alphanumeric characters.

## RETURN record

The RETURN record indicates the completion of a command in INPRO or INMSG. A zero value indicates that the command completed normally.

### Format

RETURN(*return*) ERRDESC(*error description*)

REASON(*reason*) ERRTXT(*error text*)...

SESSIONKEY(*session key*)

### Parameters

#### **RETURN**

Completion code for the Expedite Base/400 command. If the return code is zero, the command completed normally. If the return code is not zero, Expedite Base/400 displays an error number and possibly ERRDESC, REASON, and ERRTXT values. This parameter contains 5 numeric characters.

#### **ERRDESC**

Short description of the error condition. If the return code is zero, Expedite Base/400 does not display this parameter. This parameter contains 1 to 80 alphanumeric characters.

#### **REASON**

Cause of the error. If Expedite Base/400 includes REASON in the record, it gives more information about what caused the error. Expedite Base/400 usually includes REASON when an error occurs opening a file. This parameter contains 1 to 240 alphanumeric characters.

#### **ERRTXT**

Detailed description of an error which might suggest steps to correct the problem. Expedite Base/400 might include this parameter more than once in a record if the error text consists of multiple lines. This parameter contains 1 to 80 alphanumeric characters.

#### **SESSIONKEY**

Session access key provided by Information Exchange upon a successful session start. The session access key is only provided after a START command. When the RETURN record contains a session access key, this is the only parameter included in the record.

## SENT record

The SENT record returns information about the data sent that might not be apparent from the command parameters. This record appears only in OUTMSG. Expedite Base/400 creates one SENT record for each file, stream, or EDI envelope transmitted. Because parameters with blank values do not print, Expedite Base/400 might not include all the parameters listed below with each SENT record.

When the SENT record follows a SEND or SENDSTREAM command, it includes only the UNIQUEID and the LENGTH parameters.

When the SENT record follows a SENDEDI command, it returns all the parameters shown, unless the parameter value is blank.

### Format

SENT

ALIAS(*alias*) ALIASNAME(*alias name*)

or

SYSID(*system ID*) ACCOUNT(*account*) USERID(*user ID*)

or

ACCOUNT(*account*) USERID(*user ID*)

or

LISTNAME(*list name*)

UNIQUEID(*unique ID*) LENGTH(*length*) EDITYPE(*datatype*)

DESTINATION(*destination*) QUALIFIER(*qualifier*)

CONTROLNUM(*control number*) CLASS(*class*)

MSGNAME(*message name*) MSGSEQNO(*message sequence no*);

### Parameters

#### ALIAS

Alias table type and table name of the Information Exchange destination.

**gxxx** Global alias table, where *xxx* identifies a 1- to 3-character table name.

**oxxx** Organizational alias table, where *xxx* identifies a 1- to 3-character table name.

**pxxx** Private alias table, where *xxx* identifies a 1- to 3-character table name.

This parameter contains 1 to 4 characters.

#### ALIASNAME

Alias name defined in the alias table. This parameter contains 1 to 16 alphanumeric characters.

#### SYSID

System ID of the Information Exchange destination. This parameter contains 1 to 3 alphanumeric characters.

**ACCOUNT**

Account of the Information Exchange destination. This parameter contains 1 to 8 alphanumeric characters.

**USERID**

User ID of the Information Exchange destination. This parameter contains 1 to 8 alphanumeric characters.

**LISTNAME**

Name of a previously defined list of account and user IDs used as the Information Exchange destination. This parameter contains 1 to 8 alphanumeric characters.

**UNIQUEID**

Random ID assigned to the message by Expedite Base/400 to help you identify the message. The first 8 characters of the MSGDESCR parameter in the Information Exchange acknowledgment contain the UNIQUEID. See the *Information Exchange Interface Programming Guide* for more details on the format of the acknowledgment. This parameter contains 8 alphanumeric characters.

**LENGTH**

Length of the file, stream, or EDI envelope sent. This parameter contains 1 to 9 numeric characters.

**EDITYPE**

Type of EDI data sent: X12, UCS, UNTDI, or EDIFACT. This parameter contains 1 to 7 alphanumeric characters.

**DESTINATION**

Destination specified in the EDI data. This parameter contains 1 to 35 alphanumeric characters.

**QUALIFIER**

EDI qualifier for the destination. This parameter contains 1 to 4 alphanumeric characters.

**CONTROLNUM**

Interchange control number from the X12 or UCS data. For UN/TDI data, this is the SNRF element. For EDIFACT data, it is the data element 0020 (Interchange control reference). This parameter contains 1 to 14 alphanumeric characters.

**CLASS**

User class obtained from the EDI data, the CLASS parameter, or the default. This parameter contains 1 to 8 alphanumeric characters.

**MSGNAME**

Message name obtained from the EDI data or the MSGNAME parameter. This parameter contains 1 to 8 alphanumeric characters.

**MSGSEQNO**

Message sequence number obtained from the MSGSEQNO parameter or generated by Expedite Base/400. This parameter contains 1 to 5 alphanumeric characters.

## SESSIONEND record

The SESSIONEND record is the last record in the response file. The SESSIONEND record indicates the completion of an entire INMSG file. A zero value indicates that all the commands in the file completed normally.

### Format

```
SESSIONEND(return code) REASON(reason)  
ERRDESC(error description) ERRTXT(error text);
```



**NOTE:** There is only one SESSIONEND record per response file, even if there are multiple START and END commands.

### Parameters

#### RETCODE

Completion code for the Expedite Base/400 command. If the return code is zero, the command completed normally. If the return code is not zero, Expedite Base/400 displays an error number and possibly an ERRDESC, REASON, and ERRTXT value. This parameter contains 5 numeric characters.

#### ERRDESC

Description of the error condition. If the return code is zero, Expedite Base/400 does not display this parameter. If the return code is not zero, Expedite Base/400 might display this parameter. This parameter contains 1 to 80 alphanumeric characters.

#### REASON

Cause of the error. If Expedite Base/400 includes REASON in the record, it gives more information about what caused the error. Expedite Base/400 usually includes REASON when an error occurs opening a file. This parameter contains 1 to 240 alphanumeric characters.

#### ERRTEXT

Detailed description of an error which might suggest steps to correct the problem. Expedite Base/400 can include this parameter more than once in a record if the error text consists of multiple lines. This parameter contains 1 to 80 alphanumeric characters.

## SESSIONRESP record

Expedite Base/400 produces a SESSIONRESP record in response to the SESSIONINFO command. It contains information summarizing the current Information Exchange session.

### Format

SESSIONRESP  
 DATE(*date*) TIME(*time*) DATELONG(*date long format*)  
 SESSDATE(*session date*) SESSDATELONG(*session date long format*)  
 SESSTIME(*session time*) SESSLEN(*session length*) SEGSENT(*segment sent*)  
 RCVACTIVE(*command received*) SEGRVCVD(*segment received*)  
 IDLETIME(*idle time*) SYSWAIT(*p|c*)  
 MSGSENT(*message sent*) MSGOK(*message ok*) MSGREJECT(*message reject*)  
 DATARCVD(*data received*) SERVRCVD(*service received*)  
 RESPRCVD(*response received*) SENDCKPT(*send checkpoint*)  
 RCVCKPT(*receive checkpoint*) SENDID(*send ID*) RCVID(*rcvid*);

### Parameters

#### DATE

Date that Information Exchange generated the session information. The format is *YYMMDD*.

#### TIME

Time that Information Exchange generated the session information. The format is *HHMMSS*.

#### DATELONG

Long format of DATE. The format is *YYYYMMDD*. The *YY* value from DATE is interpreted by Expedite Base/400 as *YYYY*.

#### SESSDATE

Date the current Information Exchange session started. The format is *YYMMDD*.

#### SESSDATELONG

Long format of SESSDATE. The format is *YYYYMMDD*. The *YY* value from SESSDATE is interpreted by Expedite Base/400 as *YYYY*.

#### SESSTIME

Time the current Information Exchange session started. The format is *HHMMSS*.

#### SESSLEN

Length of the current Information Exchange session. The format is *HHMMSS*.

#### SEGSENT

Number of segments sent to Information Exchange by Expedite Base/400 between the time the session started and the time the SESSIONINFO command processed. This parameter contains 1 to 6 numeric characters.

#### RCVACTIVE

Number of Information Exchange RECEIVE commands active at the time the SESSIONINFO command processed. This should be zero under normal conditions. This parameter contains 1 or 2 numeric characters.



**SEGRDVD**

Number of segments sent to Expedite Base/400 by Information Exchange between the time the session started and the time the SESSIONINFO command processed. This parameter contains 1 to 6 numeric characters.

**IDLETIME**

Time elapsed since the last Information Exchange message transfer activity. The format is *HHMMSS*.

**SYSWAIT**

Indicates whether Information Exchange is waiting for a commit or pace response from Expedite Base/400.

- p Pace response
- c Commit response

Under normal conditions, Expedite Base/400 does not include this parameter in the SESSIONRESP record.

**MSGSENT**

Number of messages sent to Information Exchange by Expedite Base/400 between the time the session started and the time the SESSIONINFO command processed. Because Expedite Base/400 uses message grouping, this value might not correspond to the number of files sent by Expedite Base/400. This parameter contains 1 to 6 numeric characters.

**MSGOK**

Number of messages sent to Information Exchange by Expedite Base/400 and accepted for delivery between the time the session started and the time the SESSIONINFO command processed. Because Expedite Base/400 uses message grouping, this value might not correspond to the number of files sent by Expedite Base/400. This parameter contains 1 to 6 numeric characters.

**MSGREJECT**

Number of messages sent to Information Exchange by Expedite Base/400, but rejected between the time the session started and the time the SESSIONINFO command processed. Because Expedite Base/400 uses message grouping, this value might not correspond to the number of files sent by Expedite Base/400. This parameter contains 1 to 6 numeric characters.

**DATARCVD**

Number of data messages sent to Expedite Base/400 by Information Exchange between the time the session started and the time the SESSIONINFO command processed. Because Expedite Base/400 uses message grouping, this value might not correspond to the number of files received. This parameter contains 1 to 6 numeric characters.

**SERVRCVD**

Number of service messages sent to Expedite Base/400 by Information Exchange between the time the session started and the time the SESSIONINFO command processed. This parameter contains 1 to 6 numeric characters.

**RESPRCVD**

Number of command responses sent to Expedite Base/400 by Information Exchange between the time the session started and the time the SESSIONINFO command processed. This parameter contains 1 to 6 numeric characters.

**SENDCKPT**

Last send checkpoint number completed at the time the SESSIONINFO command processed. This parameter contains 1 to 5 numeric characters.

**RCVCKPT**

Last receive checkpoint number completed at the time the SESSIONINFO command processed. This parameter contains 1 to 5 numeric characters.

**SENDID**

ID of the last SEND command completed. It contains the values specified in the ACCOUNT, USERID, MSGNAME, and MSGSEQNO parameters of the SEND command. This parameter contains 1 to 29 alphanumeric characters.

**RCVID**

ID of the last message received. This parameter contains 1 to 30 alphanumeric characters.

## STARTED record

The STARTED record, a session start response record, provides information about the current Information Exchange session and the prior Information Exchange session. This record is written to the output file as a result of a START or AUTOSTART command.

### Format

```
STARTED  
LASTSESS(0|1) SESSIONKEY (sesskey)  
IEVERSION(version) IERELEASE(release)  
RESPCODE(code);
```

### Parameters

#### LASTSESS

Status of the previous session.

- 0       Indicates the last session was successful.
- 1       Indicates the last session was not successful.

#### SESSIONKEY

Unique identifier for this Information Exchange session. This is also used as the archive ID for the files that are archived and for which you did not specify an ARCHIVEID on the RECEIVE command. SESSIONKEY is 8 alphanumeric characters.

#### IEVERSION

Version of Information Exchange. Levels of Information Exchange are tracked as *V.R* where *V* is the version, a major enhancement in the service, and where *R* is the release, a minor enhancement. IEVERSION is 2 digits, padded on the left with zeros.

#### IERELEASE

Release of Information Exchange. Levels of Information Exchange are tracked as *V.R* where *V* is the version, a major enhancement in the service, and where *R* is the release, a minor enhancement. IERELEASE is 2 digits, padded on the left with zeros.

#### RESPCODE

Information Exchange response code for the current session (not the previous session). These codes are interpreted for you by Expedite, so if the response code is not 0 and not 2, you will get a SESSIONEND return code from Expedite indicating the problem. RESPCODE is 5 digits, padded on the left with zeros.

## WARNING record

The WARNING record indicates a low-severity problem occurred which did not prevent the command from completing.

### Format

WARNING(*warning*)  
ERRDESC(*error description*);

### Parameters

#### **WARNING**

Warning number. This parameter contains 5 numeric characters.

#### **ERRDESC**

Description of the problem. This parameter contains 1 to 80 alphanumeric characters.

## Expedite Base/400 installation overview

---

Expedite Base/400 is capable of communicating using SNA LU 6.2 or TCP/IP communications. This chapter contains an overview of the installation procedures for both communications protocols. Most of the steps mentioned here must be performed by your system programmer.

To connect to Information Exchange, you must have an Information Exchange account, user ID, and password. These items must be defined in the Expedite Base/400 profile before you attempt to connect to Information Exchange.

### Downloading the Expedite Base/400 program

Before installing Expedite Base/400 4.6, verify that your system meets the system requirements described in “Understanding the Expedite Base/400 operating environment” on page 2, and then download the software files.

1. Download the software from the GXS website:  
[http://www.gxsolc.com/public/EDI/us/support/Downloads/downloads\\_index.html](http://www.gxsolc.com/public/EDI/us/support/Downloads/downloads_index.html).
2. Select the Expedite and Expedite Base option, and click **Continue**.  
The product page opens.
3. Select the **Expedite Base/400 4.6** radio button.
4. In the **Availability** field, select a country from the list.
5. Click **Continue**.  
The Expedite and Expedite Base logon page opens.
6. Type your user ID and password, and then click **Sign in**.

**NOTE:** If this is the first time that you are downloading software from this Web site, click **I have not registered before** to obtain a user ID. User IDs are issued quickly, and you can return to this page almost immediately.

7. Complete the survey questions, as needed, and accept the license agreement.  
The software begins downloading.

8. Save the zip files to your PC and unzip them.  
For example, if using PKZIP, type the following commands:  
pkunzip expblibr46.zip  
pkunzip expblibrm46.zip

## Installing the Expedite Base/400 program

To install the Expedite Base/400 program, do the following:

1. Create two destination files on your AS/400 system. If you do not create these files, the restore process will not recognize the files as being in the SAVF format. You must create both destination files in the same directory.

**NOTE:** You can name the downloaded files EXPBLIBR and EXPBLIBRM, or you can change the names to better match your environment. If you have an earlier version of Expedite Base/400 installed, these library names are probably already in use, and you should not overwrite them.

- a. Type CRTSAVF, and then press F4 (Prompt).
- b. Type the first save file name, then the library name, and then press Enter.

**NOTE:** The library name must be that of an existing library on your system. If the library does not already exist, use the CRTLIB command to create it; for example: CRTLIB MYEXPBLIB.

The following is an example of the Create Save File panel.

```

                                Create Save File (CRTSAVF)

Type choices, press Enter.

Save file . . . . . > EXPBLIBR Name

Library . . . . . > MYEXPBLIB Name, *CURLIB

Text 'description'. . > 'Expedite Base/400 V4R6 program and f:

```

- c. Type CRTSAVF, and then press F4 (Prompt).
- d. Type the second save file name, then type the library name, and press Enter.  
The following is an example of the Create Save File panel.

```

                                Create Save File (CRTSAVF)

Type choices, press Enter.

Save file . . . . . > EXPBLIBRM Name

Library . . . . . > MYEXPBLIB Name, *CURLIB

Text 'description'. . > 'Expedite Base/400 V4R6 modules'

```

2. Use FTP to send the files from your PC to the AS/400 system, as follows:
  - a. Open an FTP session from your PC to the AS/400 system.
  - b. Use the FTP binary subcommand to set the transmission mode to **BINARY**.
  - c. Verify that the current library is the library where you created the SAVF files. If it is not, use the FTP **cd** subcommand to change to the correct library, for example:
 

```
cd MYEXPBLIB.
```

**NOTE:** If you use the **dir** command to locate the library, the transmission mode is changed to ASCII. Make sure to reissue the binary subcommand before starting transmission.

- d. Using fully qualified file names, use the FTP subcommand to transfer both files to the AS/400 system, as follows:
 

```
- put expblibr [destination_AS/400_filename]
- put expblibrm [destination_AS/400_filename]
```

3. Restore the two Expedite Base/400 4.6 libraries by issuing the RSTLIB command for each save file, as follows:
  - a. To restore the expblibr file, type the following command, and then press Enter.

```
RSTLIB SAVLIB(EXPBLIBR) DEV(*SAVF) SAVF(library/fileid)
RSTLIB(EXPBLIBR)
```



**NOTES:** The fully qualified name of the save file that you created in step 1 is *library/fileid*.

The SAVLIB parameter is the name of the library where the files were originally saved. This parameter is required and should not change.

The RSTLIB parameter is the name of the destination library. The second RSTLIB parameter value, *expblibr*, is a suggested name. You may choose to use a different name for your environment.

- b. To restore the expblibrm file, type the following command, and then press Enter.

```
RSTLIB SAVLIB(EXPBLIBRM) DEV(*SAVF) SAVF(library/fileid)
RSTLIB(EXPBLIBRM)
```



**NOTES:** EXPBLIBR contains the programs and files. EXPBLIBRM contains the modules (objects).

The fully qualified name of the save file that you created in step 1 is *library/fileid*.

The SAVLIB parameter is required and should not change.

The RSTLIB parameter value (EXPBLIBRM) is required and should not change.

## Setting up SNA LU 6.2 communications

To enable SNA LU 6.2 communications with Expedite Base/400, you must order a logical unit (LU) name, define the LU name for your system, and configure the necessary communications descriptions. Detailed instructions are provided separately for NCP leased and for NCP dial connections. Values shown in the screen illustrations are only examples and users should verify whether the values are correct for their environment.

## Ordering an LU name

Expedite Base/400 communicates with the Information Exchange front end. Before you can use Expedite Base/400, you must order the LU name that Expedite Base/400 uses when communicating with Information Exchange. In the U.S., the Information Exchange system is named *ibm0rely*.

Your GXS Sales representative can help you with your order. The order must include your local network identifier (LCLNETID). If you use a dial connection, you must also include your local control point name (LCLCPNAME) and Exchange Identifier (XID), which consists of IDBLK=056 and IDNUM=XXXXXX (where XXXXX is a unique identifier for your AS/400).

## Setting up an NCP leased connection

The following is an overview of the steps required to connect your AS/400 to the Information Exchange system through an NCP leased connection.

### Defining the LU name to your system

After you have ordered the LU name for Expedite Base/400, you must define the LU name to your system per the following instructions.

1. Use the DSPNETA command to view the current network attributes of your system.

```

Display Network Attributes
System: MYOWNILU

Current system name.....:MYOWNILU
Pending system name.....:
Local network ID.....:APPN
Local control point name.....:MYOWNILU
Default local location.....:MYOWNILU
Default mode.....:BLANK
APPN node type.....: *ENDNODE
Data compression.....: *NONE
Intermediate data compression.....: *NONE
Maximum number of intermediate sessions .: 200
Route addition resistance.....:128
Server network ID/control point name.....: *LCLNETID *ANY

```

2. Create or update APPN location list entries (WRKCFGL) for either remote or local locations.

**To create a list, type 1 in the Opt field.** To view keywords for each field, press F4.

- a. For a local location list, type **QAPPNLCL** in the **List** field.



- b. For a remote location list, type **QAPPNRMT** in the **List** field.

```

Work with Configuration Lists

Position to . . . . . Starting characters

Type options, press Enter.

1=Create 2=Change 3=Copy 4=Delete 5=Display 6=Print 7=Rename
8=Work with entries

Opt List      Type      Text
1  QAPPNLCL

```

- c. Press Enter.

The system displays the Create Configuration List panel.

```

Create Configuration List (CRTCFGL)

Type choices, press Enter.

Configuration list type . . . . > *APPNLCL
Text 'description' . . . . . APPN LOCATION LIST

```

- d. In the **Configuration list type** field, type a configuration list type name or select one from the value list.
- e. In the **Text description** field, type a free-format description of the list type.
- f. Press Enter. The system displays the list.
- g. To update a list, type 2 in the **Opt** field, and then press Enter. The system displays the list.

```

Local
Location      Text

Luname      Expedite Base/400 Local Location

```

Remote Location	Remote Network ID	Local Location	Remote Control Point	Control Point Net ID	Secure Location	
IBM0RELY	IBMIN	luname	IBMCP	IBMIN	*NO	(USA)
D77Z62IE	GBIBM000	luname	GBIBM10S/W	APPN	*NO	(Europe)
QZQB62I2	NETNMS1	luname	IBMCP	IBMIN	*NO	(Japan)

3. Define the mode description (CRTMODD).

```

Mode description. . . . . LU62
    (*note: In Europe, this value should be LU62SYS1)
Maximum sessions. . . . . 8
Maximum conversations . . . . . 8
Locally controlled sessions. . . . . 4
Maximum inbound pacing value . . 7
Inbound pacing value . . . . . 7
Outbound pacing value. . . . . 7
Maximum length of request unit . *CALC
Data compression . . . . . *NETATR
Inbound data compression . . . . . *RLE
Outbound data compression. . . . . *RLE
Text 'description' . . . . . Exp/Base 400 Mode

    Additional Parameters

Class-of-service . . . . . #CONNECT
Authority . . . . . *LIBCRTAUT
    
```

4. Create the communications definitions.
  - a. Define the line description (CRTLINSDLC).



**NOTE:** If your system already has a line description, you may choose to skip this step.

```

                                Create Line Desc (SDLC) (CRTLINSDLC)

Type choices, press Enter.

Line description .....EXPBLINE
Resource names .....CMN02
                    + for more values.
Online at IPL .....*YES
Data link role .....*NEG
Physical interface .....*RS232V24
Connection type.....NONSWTPP
Switched network backup ... *NO
Exchange identifier. ....05600001
NRZI data encoding .....*NO
                    (*note: In Europe, this value should be set to *YES)

Line speed .....19200
Modem type supported .....*NORMAL
Maximum frame size .....521
Duplex .....*FULL
Inactivity timer .....300
Poll response delay .....0
Nonproductive receive timer..320
Idle timer.....30
Connect poll timer.....30

Poll cycle pause .....0
Frame retry .....7
Text 'description' .....Expedite/Base 400 SNA Leased Line

Additional Parameters should remain at Defaults.
  
```

b. Define the host controller description (CRTCTLHOST).



**NOTE:** If your system already has a controller description, you may choose to skip this step.

```

                                Create Ctl Desc (SNA Host) (CRTCTLHOST)

Type choices, press Enter.

Controller description ..... EXPBCTL
Link type ..... *SDLC
Online at IPL ..... *YES
Switched connection ..... *NO
Switched network backup. .... *NO
APPN-capable ..... *YES
Attached nonswitched line ..... EXPBLINE
Maximum frame size ..... *LINKTYPE
Remote network identifier ..... IBMIN
Remote control point ..... IBMCP
SSCP identifier ..... 052100000001
Local exchange identifier ..... *LIND
Station address. .... C0
APPN CP session support ..... *NO
APPN node type. .... *LENNODE
APPN/HPR capable ..... *YES
H/R path switching ..... *NO
APPN transmission group number 1
APPN minimum switched status ... *VRYONPND
Autocreate device ..... *ALL
Autodelete device ..... 1440
User-defined 1 ..... *LIND
User-defined 2 ..... *LIND
User-defined 3 ..... *LIND
Recontact on vary off ..... *YES
Text 'description' ..... Expedite Base 400 Leased Controller

Additional Parameters should remain at Defaults.
    
```

- c. Define the device description (CRTDEVAPPC).

Values supplied by network	Description
XX	Local location address
luname	Local location name

```

                                Create Device Desc (APPC) (CRTDEVAPPC)

Type choices, press Enter.

Device description . . . . . EXPBDEV
Remote location . . . . . IBM0RELY
Online at IPL . . . . . *NO
Local location . . . . . luname
Remote network identifier . . . . . IBMIN
Attached controller . . . . . EXPBCTL
Mode . . . . . LU62
                        + for more values

Message queue . . . . . QSYSOPR
  Library . . . . . *LIBL
APPN-capable . . . . . *YES
Single session:
  Single session capable . . . . . *YES
  Number of conversations . . . . . 4
Locally controlled session . . . . . *NO
Text 'description' . . . . . Expedite Base 400 Device

                        Additional Parameters

Local location address . . . . . XX
Authority . . . . . *LIBCRTAUT

(*Note: AS400 Local location address (hex value) must match NCP
Local Location Address (Decimal Value))

```

5. Include the LU name in the USERLUNAME parameter of the SNACOMM profile command as follows:

```
snacomm ieluname(IBM0RELY) userluname(luname) ielumode(LU62)
ielunetid( IBMIN) ;
```

6. Configure the Expedite Base/400 files to complete the installation process. See Chapter 2, “Setting up files,” for detailed information.

## Setting up an NCP dial connection

The following is an overview of the steps required to connect your AS/400 to the Information Exchange system through an NCP dial connection.

### Defining the LU name to your system

After you have ordered the LU name for Expedite Base/400, you must define the LU name to your system per the following instructions.

1. Use the DSPNETA command to view the current network attributes of your system.

```

Display Network Attributes
System: MYOWNILU

Current system name. . . . . :MYOWNILU
Pending system name. . . . . :
Local network ID . . . . . APPN
Local control point name . . . . . :MYOWNILU
Default local location. . . . . MYOWNILU
Default mode. . . . . BLANK
APPN node type. . . . . *ENDNODE
Data compression. . . . . *NONE
Intermediate data compression . . . . . : *NONE
Maximum number of intermediate sessions . : 200
Route addition resistance. . . . . :128
Server network ID/control point name. . . . . : *LCLNETID *ANY
    
```

2. Create or update APPN location list entries (WRKCFGL) for either remote or local locations.

To create a list, type **1** in the **Opt** field. To view keywords for each field, press F4.

- a. For a local location list, type **QAPPNLCL** in the **List** field.
- b. For a remote location list, type **QAPPNRMT** in the **List** field.

```

Work with Configuration Lists

Position to. . . . Starting characters

Type options, press Enter.

1=Create 2=Change 3=Copy 4=Delete 5=Display 6=Print 7=Rename
8=Work with entries

Opt List      Type      Text
1  QAPPNLCL
    
```

- c. Press Enter. The system displays the Create Configuration List panel.

```

                                Create Configuration List (CRTCFGL)

Type choices, press Enter.

Configuration list type . . . . > *APPNLCL
Text 'description' . . . . . APPN LOCATION LIST

```

- d. In the **Configuration list type** field, type a configuration list type name or select one from the value list.
- e. In the **Text description** field, type a free-format description of the list type.
- f. Press Enter. The system displays the list.
- g. To update a list, type 2 in the **Opt** field, and then press Enter. The system displays the list.

Local Location	Text
Luname	Expedite Base/400 Local Location

Remote Location	Remote Network ID	Local Location	Remote Control Point	Control Point Net ID	Secure Location	
IBM0RELY	IBMIN	luname	IBMCP	IBMIN	*NO	(USA)
D77Z62IE	GBIBM000	luname	GBIBM10S/W	APPN	*NO	(Europe)
QZQB62I2	NETNMS1	luname	IBMCP	IBMIN	*NO	(Japan)

3. Define the mode description (CRTMODD).

Mode description . . . . .	LU62
(*note: In Europe, this value should be LU62SYS1)	
Maximum sessions . . . . .	8
Maximum conversations. . . . .	8
Locally controlled sessions . . . . .	4
Maximum inbound pacing value . . .	7
Inbound pacing value . . . . .	7
Outbound pacing value . . . . .	7
Maximum length of request unit. . .	*CALC
Data compression . . . . .	*NETATR
Inbound data compression. . . . .	*RLE
Outbound data compression . . . . .	*RLE
Text 'description' . . . . .	Exp/Base 400 Mode
Additional Parameters	
Class-of-service . . . . .	#CONNECT
Authority . . . . .	*LIBCRTAUT



4. Create the communications definitions.
  - a. Define the line description (CRTLINSDLC).



**NOTE:** If your system already has a line description, you may choose to skip this step.

```

                                Create Line Desc (SDLC) (CRTLINSDLC)
Type choices, press Enter.

Line description ..... EXPB400LIN
Resource names ..... CMN02
      + for more values
Online at IPL ..... *NO
Data link role ..... *SEC
Physical interface ..... *RS232V24
Connection type ..... *SWTPP
Vary on wait ..... *NOWAIT
Autocall unit ..... *NO
Exchange identifier ..... 05600001
NRZI data encoding ..... *NO
      (*note: In Europe, this value should be set to *YES)
Maximum controllers ..... 1
Clocking ..... *MODEM
Line speed ..... 9600
Modem type supported ..... *NORMAL
Switched connection type .... *BOTH
Autoanswer ..... *NO
Autodial ..... *YES
Dial command type ..... *V25BIS
Calling number ..... *NONE
Station address ..... C1
Maximum frame size ..... 521
Duplex ..... *HALF
Inactivity timer ..... 300
Poll response delay ..... 0
Data Set Ready drop timer ... 6
Autoanswer type ..... *DTR
Remote answer timer ..... 60
Text 'description' ..... Expedite Base 400 line

Additional Parameters should remain at Defaults.
  
```

- b. Define the host controller description (CRTCTLHOST).



**NOTE:** If your system already has a controller description, you may choose to skip this step.

```

                                Create Ctl Desc (SNA Host) (CRTCTLHOST)

Type choices, press Enter.

Controller description . . . . . EXPBCTL
Link type . . . . . *SDLC
Online at IPL . . . . . *YES
Switched connection . . . . . *YES
Short hold mode . . . . . *NO
APPN-capable . . . . . *YES
Switched line list . . . . . EXPB400LIN
                                + for more values
Maximum frame size . . . . . *LINKTYPE
Remote network identifier . . . . . IBMIN
Remote control point . . . . . IBMCP
SSCP identifier . . . . . 05210000001
Local exchange identifier . . . . . *LIND
Initial Connection . . . . . *DIAL
Dial initiation . . . . . *LINKTYPE
Connection Number . . . . . '1-800-759-9575'
Station address . . . . . C0
APPN CP session support . . . . . *NO
APPN node type . . . . . *LENNODE
APPN/HPR capable . . . . . *YES
H/R path switching . . . . . *NO
APPN transmission group number . . . . . 1
APPN minimum switched status . . . . . *VRYONPND
Autocreate device . . . . . *ALL
Autodelete device . . . . . 1440
User-defined 1 . . . . . *LIND
User-defined 2 . . . . . *LIND
User-defined 3 . . . . . *LIND
Text 'description' . . . . . Expedite Base 400 Controller

Additional Parameters should remain at Defaults.
    
```

- c. Define the device description (CRTDEVAPPC).

Values supplied by network	Description
XX	Local location address
luname	Local location name

```

Create Device Desc (APPC) (CRTDEVAPPC)

Type choices, press Enter.

Device description ..... EXPBDEV
Remote location ..... IBMORELY
Online at IPL ..... *NO
Local location ..... luname
Remote network identifier ... IBMIN
Attached controller ..... EXPBCTL
Mode ..... LU62
    + for more values
Message queue ..... QSYSOPR
  Library ..... *LIBL
APPN-capable ..... *YES
Single session:
  Single session capable ... *YES
  Number of conversations ... 4
Locally controlled session .... *NO
Text 'description' ..... Expedite Base 400 Device

Additional Parameters

Local location address ..... XX
Authority ..... *LIBCRTAUT

(*Note: AS400 Local location address (hex value) must match NCP
Local Location Address (Decimal Value))

```

5. Include the LU name in the USERLUNAME parameter of the SNACOMM profile command as follows:

```
snacomm ieluname(IBM0RELY) userluname(luname) ielumode(LU62)
ielunetid( IBMIN) ;
```

6. Configure the Expedite Base/400 files to complete the installation process. See Chapter 2, “Setting up files,” for detailed information.

## Setting up TCP/IP communications

To enable TCP/IP communications, you must consider the size of your network, the other applications you will be accessing through the worldwide network, and your hardware configuration. Network personnel will work with you to allow communications with Information Exchange.

Detailed instructions are provided for TCP/IP dial connections. Values shown in the screen illustrations are only examples, and users should verify whether the values are correct for their environment.

TCP/IP leased connections must be set up in conjunction with network personnel. Your network representative can assist you.

### Setting up a TCP/IP dial connection

For best results, use the point-to-point (PPP) Internet protocol and OS/400 Version 4 Release 1 or Release 2. This protocol uses a PPP line description on the AS/400 system and is configured using the Operations Navigator component of Client Access for Windows 95/NT™. For more information, see *OS/400 TCP/IP Configuration and Reference* and *AS/400 Client Access for Windows 95/NT*.

The following section describes basic PPP configuration on the AS/400 for using TCP/IP communications with the worldwide network through a Local Interface Gateway (LIG).

Before you can use TCP/IP communications:

- The asynchronous modem that you will use for TCP/IP connectivity must be defined on your system.
- Your user ID must be authorized for TCP/IP Dial Services and for access to the Information Exchange/Expedite IP server. Contact the Help Desk if you need assistance with this step.

### Configuring Point-to-Point Network Connections

Using PPP and a PPP line description requires an asynchronous modem and one of the following adaptors:

2699	Two-line AN IOA
2720	PCI WAN, twin-axial IPA
2721	PCI two-line WAN IOA
7852-400	Asynchronous, bisynchronous, SDLC

To verify that the correct adaptor is present on your system, use the following command:

```
wrkhdwrs type(*cmn)
```

For testing purposes only, it may be possible to use the Electronic Customer Support modem on your AS/400.

### Setting up the connection profile and PPP line description

You must create and activate a connection profile for using PPP over a PPP line description. For additional information on any field, click on the item in question and press **F1**.

To create a connection profile:

1. In the main tree of Operations Navigator, click on your AS/400 server to expand the tree.
2. Click **Network**, and then **Point-to-Point**.
3. Right-click **Connection Profiles** to open a context menu.
4. Select **New Profile**.
5. Click the **General** tab and enter the following information:

In this field:	Type or select:
Profile name	The name of this connection profile. Use 1 to 9 alphanumeric characters.
Profile description	A free-form description of this connection profile.
Type	PPP
Mode	Switch line-dial

6. Click the **Connection** tab and enter the following information:

In this field:	Type or select:
Add remote phone number	Select this option.
LIG dial access number	Type the telephone number for your area. If needed for your telephone system, add a dial-out access code (such as 9). If you are unsure of the correct local LIG access telephone number, contact the Help Desk.
Line name	Select this option.
PPP line description	Select a line description from the <b>Line</b> field value list. To create a PPP line description, see “Creating or changing line descriptions” on page 204 for more information.
Redial on Disconnect	Select this option.
Line activity timeout	Type 300 seconds.

7. Click the **TCP/IP** tab and verify that the settings are correct. Both the Local IP and Remote IP addresses will be dynamically assigned by the LIG.

8. Click the **Routing** tab and enter the following information:

In this field:	Type or select:
Dynamic routing	Select None.
Static routing	Select this option.
Route	Type the Expedite server IP address (32.76.15.6 in the U.S.) and mask (255.255.255.255).

9. To set up system authentication, click the **Authentication** tab and enter the following information:

In this field:	Type or select:
Enable Local System Identification	Select this option.
Pap Only	Select this option.
User Name	Type your user name in the format: <i>secureip.acct.userid</i> , where <i>acct</i> is your 4-digit network account ID and <i>userid</i> is your network user ID.

10. At the password prompt, type your current network password.
11. Click **OK** repeatedly until you are returned to the list of Connection Profiles.
12. To initiate a dial connection, right-click on the appropriate PPP profile, and then click **Start**.
13. Run Expedite Base/400 to communicate with Information Exchange. When done, click **Stop** to end your dial session. This is the only time you will need to manually stop a dial session. From now on, Expedite Base/400 will end sessions automatically when processing has completed.

#### Creating or changing line descriptions

- Do one of the following:
  - To set up a line description, type a name for the line description in the **Line** field, and then click **New**.
  - To change the properties of a line description, select a line description in the Line field list and click **Open**.
- Click the **General** tab and highlight the appropriate hardware resource in the **Hardware Resource** list. The interface type must be RS232/V.24.
- Click the **Connection** tab and enter the following information:

In this field:	Type or select:
Dial Command type connection	Select AT command set.
CTS timeout	Type 60 seconds.
Dial Capability	Select this option.

4. Click the **Link** tab and specify the maximum line speed supported by your hardware in the **Line Speed** field.
5. Click the **Modem** tab and select your modem from the **Name** value list.
6. Click **OK** to save the new or changed properties. The system will return you to the PPP profile definition.

## Obtaining and managing certificates for use with SSL

Before you can begin using SSL communications with Expedite Base/400, make sure that the required software has been installed on your system. You must also install Digital Certificate Manager (DCM) to use SSL with TCP/IP communication to Information Exchange.

**NOTE:** You need SECOFR user authority to work with DCM.

### Obtaining a certificate

To obtain a certificate, do the following:

1. Using your browser, navigate to the PKI Web site (<https://pki.services.ibm.com>).
2. Obtain a client and PKI Services Root CA Certificate.
  - a. Create a certificate using the instructions on the Web site <https://pki.services.ibm.com/>.
  - b. Export your certificate to files using the instructions on the Web site under the heading **Method 2: Exporting your client certificate and CA as separate files**. When you export the certificate, make sure that **Include all certificates in the certificate path if possible** is not selected.
  - c. Download a copy of the PKI Services Root CA Certificate from the PKI Web site at: <http://pki.services.ibm.com/expedite/webdocs.shtml#ca>.
3. FTP the certificate files to the AS/400.
  - a. After logging in, change to a directory of your choice; for example, /tmp.
  - b. Change to binary mode.
  - c. Transfer the PKCS#12 client file, which was exported from your Internet Explorer browser, to the AS/400.
  - d. In the same way, transfer the PKI Services Root CA Certificate to the AS/400.

## Managing Certificates in DCM

To manage certificates in DCM, do the following:

1. Using Internet Explorer, go to <http://AS400HOST:2001>, where AS400HOST is the IP address of your AS400.  
The Enter Network Password window opens.



2. Type your AS/400 ID and password, and then click **OK**.  
The main AS/400 Tasks page opens.



3. Click **Digital Certificate Manager**.  
The Digital Certificate Manager page opens.
4. Click **Select a Certificate Store**.
5. To assign a certificate to an application ID, select **\*SYSTEM**, and click **Continue**.  
The confirmation message displays.  
**NOTE:** If the **\*SYSTEM** store does not exist, you can create one by selecting the **Create New Certificate Store** link in the left linklist.
6. In the left navigation bar, click **Manage Certificates**.  
The Manage Certificates page opens.
7. Select **Import Certificate**, and then click **Continue**.  
The Import Certificate page opens.

8. Select **Certificate Authority (CA)**, and click **Continue**.  
The Import Certificate Authority (CA) Certificate page opens.
9. In the **Import file** field, type the location of the PKI Services Root CA certificate, and then click **Continue**.  
The Import Certificate Authority (CA) Certificate page refreshes, showing the CA certificate label field.
10. In the **CA certificate label** field, type a label for the certificate, such as PKI Services Root CA, and then click **Continue**.  
The PKI Services Root CA is imported.
11. On the Import Certificate page, in the left navigation bar, click **Import certificate**.  
The Import Certificate page refreshes.
12. Select **Server or client**, and then click **Continue**.  
The Import Server or Client Certificate page opens.
13. Type the name of the PKCS#12 client file that you transferred to your AS/400, and click **Continue**.  
The certificate password page opens.

**NOTE:** **Include all certificates in the certificate path if possible** should **not** have been selected when you exported your certificate from Internet Explorer.

14. In the **Password** field, type the password for the PKCS#12 file, and click **Continue**.  
The Import Server or Client Certificate page refreshes, showing that the certificate was imported.  
  
**NOTE:** You do not need to click **OK**.
15. In the left navigation bar, click **Manage Applications** to expand it.
16. Click **Add application**, and click **Continue**.  
The Add Application page opens.
17. Select **Client**, and click **Continue**.
18. In the **Application ID** field, type the name of your application.  
This should be a value that you can associate easily with Expedite Base/400 and your Information Exchange user ID.
19. In the **CA trust list** field, select **Yes**.
20. In the **Application description** field, type `Expedite with SSL`, or the description of your choice.
21. Click **Add**.  
The Certificate is added.  
  
**NOTE:** You do not need to click **OK**.
22. In the left navigation bar, select **Update certificate assignment**.  
The Update Certificate Assignment page opens.

23. Select **Client**, and click **Continue**.  
The Update Certificate Assignment detail page opens.
24. Select the radio button for the application that you just added, and click **Update Certificate Assignment**.
25. Select the certificate that you just imported.  
  
**NOTE:** The Certificate Name may or may not be recognizable, but the Common name should be recognizable.
26. Click **Assign New Certificate**.
27. Select **Client**, and click **Continue**.  
The Application type page opens. The certificate configuration is now complete.
28. Make sure that the Account/Userid and Application ID information in INPRO and INMSG match the information in the certificate, and then run Expedite.

This process is complete.

## Expedite Base/400 objects

Expedite Base/400 contains the following objects in the main library, *EXPBLIBR*:

- **AUDITFMT**  
This sample COBOL language program can be used to format level 1 audit records from Information Exchange. *AUDITFMT* source code is located in the *SAMPLESRC* file.
- **CRTIEBASER**  
This program is supplied to rebind the *IEBASER* program. Use this program only when specified by the Help Desk. *CRTIEBASER* source code is located in the *SAMPLESRC* file.
- **ERRORCMP**  
This file contains the error messages issued when Comm-Press encounters an error.
- **ERRORMSG**  
This file contains the error messages issued when Expedite Base/400 encounters an error.
- **ERRORTXT**  
This file contains the explanations and appropriate user responses for the error messages in *ERRORMSG*.
- **IEBASE**  
This stand-alone program for Expedite Base/400 calls *IEBASEPR*, *INMSGP*, *IEBASER*, *IEBASEPO*, and *OUTMSGP*. The *INMSGP* and *OUTMSGP* programs are present only if you have the supported data compression software.



**NOTE:** All user programs should call this program.

- **IEBASER**  
This is the main program that provides all the functions of Expedite Base/400.



**NOTE:** User programs should *not* call this program. (See *IEBASE*.)

- **IEBASEPO**  
This program checks the message response file, *OUTMSG*, for compressed files, and verifies the presence of the *OUTMSGP* program required for data decompression.
- **IEBASEPR**  
This program checks the *COMPRESS()* parameter, and verifies the presence of the *INMSGP* program required for data compression.
- **SAMPLESRC**  
This file contains sample source code for Expedite Base/400 related programs.
- **STRIEBASE**  
This program can be used to run *IEBASE*; it replaces *CALL IEBASE PARM(RESET)*.
- **XINMSG**  
This file contains sample message commands that can be used to run sample sessions.
- **XINPRO**  
This file contains sample profile command files that can be used to run sample sessions.

- XQUALTBL  
This file contains QUALTBL examples that can be used for transmitting EDI data.
- XTCPIPADD  
This file contains the TCPCOMM commands that can be used to connect to Information Exchange using TCP/IP.
- XTTABLE01  
This file contains TTABLE01 examples that can be used for transmitting EDI data.



## Product examples

---

The following sections contain three Expedite Base/400 jobs. The jobs send and receive files, EDI data, and stream data. They also retrieve audit data, receive Information Exchange system error messages, and cancel previously sent files. The examples shown in this appendix are also included on the product tape.

### SEND and RECEIVE command example

The following example defines a list, sends a file, and receives a file. It also retrieves audit data and places it in the Information Exchange mailbox, cancels previously sent files, and receives Information Exchange system error messages. It uses session-level recovery.

```
Filename: XINMSG(EX1)
#Define a temporary list named mylist containing two users #
list listname(mylist)
account(act1) userid(user002) account(act1) userid(user003);

#Send a file named test to the list named mylist #
#use user class testclas.          #

send fileid(test) listname(mylist) class(testclas);

#Receive a file called test from account act1, #
#userid user002, and user class testclas #

receive fileid(testout) account(act1)
userid(user002) class(testclas);

#Cancel any files sent to account act3, userid user005, #
#with a class of testcls #

cancel account(act3) userid(user005) class(testcls);

#Load the audit records for any files sent to or received #
#from account act2, userid user005 #
#The audit data must be retrieved in a subsequent session #
```

```
audit account(act2) userid(user005);
#Receive IE system error messages into the file referenced by #
#File IESYSERR #
receive fileid(iesyserr) account(*system*) userid(*errmsg*);
```

**Filename: XINPRO(EX1)**

```
#Provide the IE account, userid, and password to be used for #
#the IE session start #
identify ieaccount(act1) ieuserid(user01) iepassword(paswrđ) timezone(est);
#Provide the IE LU Name and the LU name of Expedite Base/400 #
snacomm ieluname(ieapplid) userluname(myapplid) ielumode(LU62) ielunetid(IBMİN);
```

## SENDEDI and RECEIVEEDI command example

The following example sends an EDI file using both an EDI qualifier and EDI destination table. It also receives an EDI file and Information Exchange system error messages. It uses checkpoint-level recovery.

**Filename: XINMSG(EX2)**

```
#Send the EDI data in file edi test #
#Use a user class of editest #
sendedi fileid(edi) class(editest);
#Receive the EDI data sent from account act1, userid user002 #
#with a user class of editest #
#Place the data in file ediout #
receiveedi fileid(ediout)
account(act1) userid(user002) class(editest);
#Receive IE system error messages into the file referenced by #
#File IESYSERR #
receive fileid(iesyserr) account(*system*) userid(*errmsg*);
```

**Filename: XINPRO(EX2)**

```
#Provide the IE account, user ID, and password to be used for #
#the IE session start #
identify ieaccount(act1) ieuserid(user01) iepassword(paswrđ) timezone(est);
#Provide the IE LU name and the LU name of Expedite Base/400 #
snacomm ieluname(ieapplid) userluname(myapplid) ielumode(LU62) ielunetid(IBMİN);
#Ask for checkpoint-level recovery #
transmit recovery(c);
```



Filename: XQUALTBL  
 datatype(x) qualifier(01) alias(gx01) ttable(ttable01);

Filename: XTTABLE01  
 edidest(duns01) account(act2) userid(user001);  
 edidest(duns02) account(act3) userid(user005);

## SENDSTREAM and RECEIVESTREAM command example

The following example sends data from the message command file (INMSG) using the SENDSTREAM command. It also receives data and Information Exchange system error messages into the message response file (OUTMSG) using RECEIVESTREAM commands. It uses session-level recovery.

Filename: XINMSG(EX3)

```
#Send data following the SENDSTREAM command to #
#jones in alias table ptb1 with user class testclas #
#the data ends when the xxx string is found #

sendstream alias(ptb1) aliasname(jones)
class(testclas) endstr(xxx);
line one of data
line two of data
xxx

#Receive data from account act1, userid user002 with #
#user class class1 #
#End the data with the string xxxx #
#the RECEIVED record in OUTMSG #
receivestream account(act1) userid(user002) endstr(xxxx) class(class1);

#Receive IE system error messages #
receivestream account(*system*) userid(*errmsg*) endstr(?x$xx?);
```

Filename: XINPRO(EX3)

```
#Provide the IE account, userid, and password to be used for #
#the IE session start #

identify ieaccount(act1) ieuserid(user01) iepassword(paswrld) timezone(est);

#Provide the IE LU Name and the LU name of Expedite Base/400 #

snacomm ieluname(ieapplid) userluname(myapplid)
ielunetid(IBMIN);
```

## INPRO example using SSL

The following example shows how your INPRO file might look if you wanted to use SSL when communicating with Information Exchange.

```
Filename: XINPRO(EX4)
#Provide the IE account, userid, password and application id #
#to be used for the SSL IE session start #

IDENTIFY IEACCOUNT(act1) IEUSERID(user01) IEPASSWORD(paswrđ)
APPLICID(expeditessl) TIMEZONE(est);

#Provide the IP address and IP host of the Secure Front End #

TCPCOMM IETCPHOST1(204.146.24.85) IETCPPORT1(3667);

# Provide the TRANSMIT command and specify a TCP/IP session #

TRANSMIT COMMTYPE(T);

# Provide TRACE command parameters if you would like #

TRACE PROTOCOL(Y) BASE(Y) IOFILE(Y) LINK(Y);

# Provide new SSL command #

SSL ENABLESSL(Y);
```

## START and SEND commands using SSL example

The following example starts an SSL session using parameters from the START command, then sends data.

```
Filename: XINMSG(EX4)
#In INPRO, set COMMTYPE(T), AUTOSTART(N) and SSL ENABLESSL(Y). #
#This INMSG file will contain the START command with SSL parms.#

START ACCOUNT(act1) USERID(user1) IEPASSWORD(iepw)
APPLICID(expeditessl);

#Send a file to an IE user, with a user class of SSL. #

SEND FILEID(test) ACCOUNT(acct2) USERID(user2) CLASS(SSL);
```

## Common data header (CDH)

---

Information Exchange interfaces can use a common data header (CDH) to communicate detailed information about files and messages to other interfaces and to Information Exchange. Expedite Base/400 builds a CDH for every file sent, so you never need to build a CDH yourself. Expedite Base/400 also recognizes CDHs received from other interfaces.

The CDH provides details (such as file name and carriage-return and line-feed options) that let the receiving interface reconstruct a received message into its original format. It also makes more information available to the recipient of a file.

The actual location and format of the CDH is transparent to the Expedite Base/400 user. However, when looking at a trace of the actual Information Exchange message flow, you might notice an additional message at the beginning of each Information Exchange message group. This extra message contains the CDH.

The information in the CDH is presented to you in the form of parameters in the RECEIVED or AVAILABLE records.

### Construction of the CDH

The CDH is constructed during the SEND, SENDEDI, SENDSTREAM, and PUTMEMBER process and contains the following information:

- File name and location
- Record format and length
- Data type (EBCDIC or binary)
- Record delimiter (carriage-return line-feed characters or length indicators)
- Data format (EDIFACT, X12, UCS, or UN/TDI)
- Description (see “User data” on page 218)
- Unique ID
- Sending code page
- Sending system
- Sending version
- File date
- File time

- Carriage-return line-feed (CRLF) and end-of-file (EOF) characters
- EDI sender ID qualifier
- EDI sender ID
- EDI receiver ID qualifier
- EDI receiver ID
- EDI interchange control number
- Length of data
- Library owning account
- Library name
- Library member
- Library member replacement
- Compression software package name
- Compression software package version
- Name of compressed file

## User data

To further identify a file or to include extra information or instructions, you can supply additional user data in the CDH. This feature is optional, but you can add up to 79 bytes of user data by using the DESCRIPTION parameter of the SEND, SENDSTREAM, SENDEDI, and PUTMEMBER commands.



**NOTE:** For the SENDEDI commands, a CDH is constructed for each interchange contained in the file. If a DESCRIPTION parameter is present, the user data supplied is copied to the CDH for each interchange.

## Libraries

The Expedite interfaces use the CDH library fields to pass the library information to Information Exchange when processing a PUTMEMBER or GETMEMBER command. This information includes the library owner's account, the library name, the member name, and a replace indicator. For other commands, these fields will not be included in the CDH.

## Entry fields

For the layout of the CDH entry fields, see *Information Exchange Interface Programming Messages and Codes*.

## Sample audit report

---

With the `AUDIT` command, users can select specific criteria when they process the command (for example, a user might want to have a report only on messages received or only on messages sent). Because of this, the content of any given report depends on the criteria selected. The sample report shown at the end of this appendix illustrates the output that might result if all the criteria are selected. The program used to generate this report is included on the product tape.



**NOTE:** The procedures and the example presented in this appendix are only for level 1 message audit record formats. Modifications would be necessary to format a level 2 or level 3 audit record.

## Using the `AUDIT` command

To use the `AUDIT` command, follow these steps:

1. Retrieve the audit records by issuing the `AUDIT` command. This places the records in your mailbox.
2. When the audit information is available, issue the `RECEIVE` command to retrieve the records from your mailbox. The audit information comes from the `*SYSTEM*` account ID and the `*AUDITS*` user ID in message user class `#SAUDIT`.



**NOTE:** Audits are not available immediately; you cannot successfully issue the `AUDIT` command immediately followed by the `RECEIVE` command. Audits are usually available during the next session.

3. Run the sample audit report program, `AUDITFMT`.

The flowchart shown here illustrates how Expedite Base/400 processes an AUDIT command and a RECEIVE or RECEIVESTREAM command to produce an audit report.

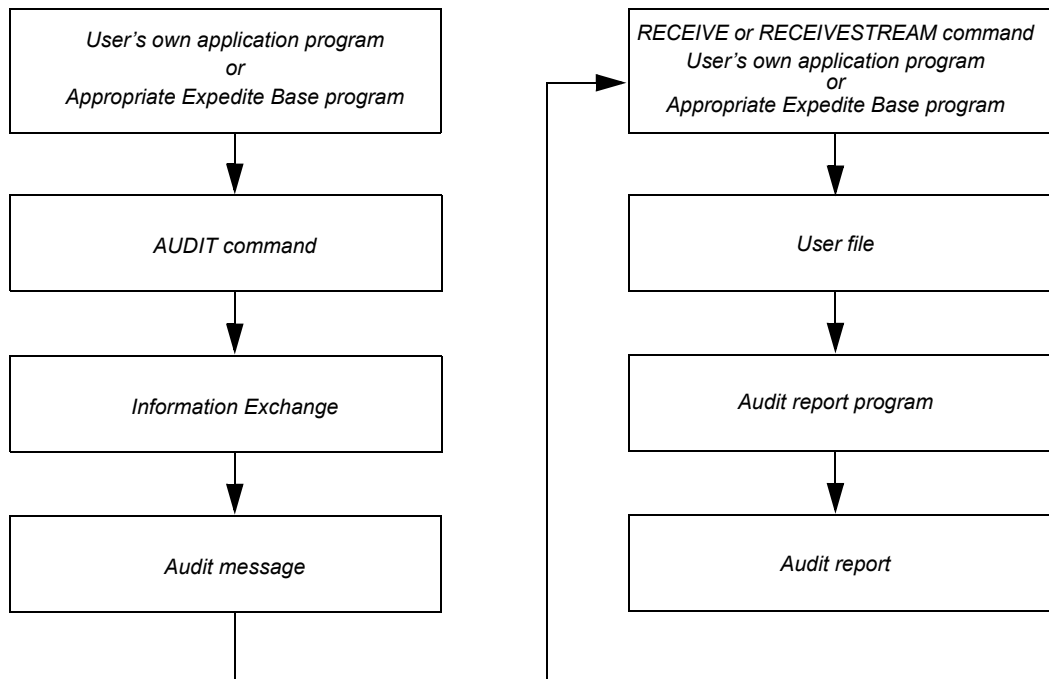


Figure 1. Producing an Audit Report

## Sample audit report layout

The following examples illustrate the layout of an audit report showing the status of messages sent and received by one user. The final two pages of the report show totals for each category.

DATE: 03 31 04										PAGE 000004
INFORMATION EXCHANGE SAMPLE AUDIT REPORT										
ACCOUNT: AAAA USERID: USERAA										
OUTBOUND TO: SYSID: ACCT: ZZZZ USERID: USERZZ ALIAS: EDI QUAL: EDI RECEIVER:										
DATE SENT	TIME SENT	DATE RCVD/PURGE	TIME RCVD/PURGE	MESSAGE NAME	MESSAGE SEQUENCE	MESSAGE CLASS	TEXT SIZE	STATUS	PURCE REASON	EDI CONTROL ID
03/19/04	14:02:00	03/19/04	14:03:52	MESSAGE1	12345	CLASS001	00000004	RECEIVED		
03/19/04	14:02:25	03/19/04	14:03:55	MESSAGE2	12350	CLASS002	00000009	RECEIVED		
03/19/04	14:04:42	03/24/04	17:40:06	MESSAGE3	12487	CLASS003	00000012	PURGED	CANCEL BY RCVR	
03/19/04	14:02:52	03/25/04	17:40:27	MESSAGE4	12489	CLASS004	00000025	PURGED	CANCEL BY RCVR	
03/19/04	10:12:19	03/20/04	12:10:11	X12MSG	X12	#E2	00000500	RECEIVED		0000123456789

DATE: 03 31 04										PAGE 000005	
INFORMATION EXCHANGE SAMPLE AUDIT REPORT											
ACCOUNT: AAAA USERID: USERAA											
OUTBOUND TOTALS: MSG TO SELF TOTALS: INBOUND TOTALS:											
IN TRANSIT	RECEIVED	PURGED	IN MAILBOX	IN TRANSIT	RECEIVED	PURGED	IN MAILBOX	IN TRANSIT	RECEIVED	PURGED	IN MAILBOX
0000000000	0000000001	0000000005	0000000001	0000000000	0000000000	0000000001	0000000002	0000000000	0000000003	0000000002	0000000001

DATE: 03 31 04										PAGE 000006	
INFORMATION EXCHANGE SAMPLE AUDIT REPORT											
TOTALS FOR ACCOUNT: AAAA											
OUTBOUND TOTALS: MSG TO SELF TOTALS: INBOUND TOTALS:											
IN TRANSIT	RECEIVED	PURGED	IN MAILBOX	IN TRANSIT	RECEIVED	PURGED	IN MAILBOX	IN TRANSIT	RECEIVED	PURGED	IN MAILBOX
0000000000	0000000001	0000000005	0000000001	0000000000	0000000000	0000000001	0000000002	0000000000	0000000003	0000000002	0000000001





## Expedite Base/400 messages and codes

---

This appendix describes the Expedite Base/400 messages and codes. The messages and codes are divided into the following categories:

- Expedite Base/400 completion codes (00-12), page 224
- Expedite Base/400 reason codes (1004-1060), page 225
- Expedite Base/400 return codes

This section includes the following codes:

- Successful completion code (00000), page 226
- Message command file syntax errors (02014 - 04988), page 227
- Profile command file syntax errors (05024 - 06014), page 259
- Parser errors (14000 - 15042), page 265
- Destination verification errors (16020 - 16060), page 267
- EDI errors (17102 - 19022), page 269
- File validation errors (20022 - 20052), page 279
- General environment errors (20365 - 23510), page 279
- Session start and end errors (24000 - 24620), page 284
- Session errors (28000 - 28200), page 287
- APPC communications support errors (29814 - 29999), page 290
- GSKIT errors (30006 - 30118), page 292
- Unexpected errors (31000 - 32000), page 292

## Expedite Base/400 completion codes

These completion codes appear in your job log as the return code for a step.

---

00

**Explanation:** Expedite Base/400 completed successfully with no errors.

**User Response:** No user action is required.

---

04

**Explanation:** Expedite Base/400 completed the session, but errors or warnings were detected.

**User Response:** Check the RETURN and WARNING records in the message response file, OUTMSG, to determine what problems occurred. Try the commands again that caused errors, if necessary.

---

08

**Explanation:** Expedite Base/400 could not complete the session due to an error.

**User Response:** Check the SESSIONEND record to determine the cause of the error. Correct the problem and try the program again.

---

12

**Explanation:** An invalid parameter was specified on the CALL statement.

**User Response:** Check output display for an error message. See “Calling IEBASE parameters” on page 20 for more information. Correct the problem and try the program again.

## Expedite Base/400 reason codes

Reason codes are generally provided when there are problems accessing a file. The reason code is accompanied by one of the error codes described in the subsequent sections of this appendix.

---

1044      File *x* not found.

**Explanation:** File *x* could not be found.

**User Response:** Check to be sure that the file is correctly defined in your command file. Be sure that there are no spelling errors and that the fully qualified file name is specified.

---

1050      The syntax for fileid *x* is invalid.

**Explanation:** The syntax of file *x* specified by the FILEID parameter is invalid.

**User Response:** Correct the FILEID parameter value in your command file.

---

1060      Unable to access file *x*, RC1=AAAA, RC2=YYYY.

**Explanation:** File *x* could not be opened. The values for RC1 and RC2 (reason codes) give additional information.

**User Response:** Check to be sure that the file is correctly defined in your command file and that your job can access the file. You may examine the operating system error messages for additional information. If the problem continues, report the problem to the Help Desk.

## Expedite Base/400 return codes

This section provides an explanation of the Expedite Base/400 return codes and tells you what action to take in response to each return code. The product displays these codes in the WARNING, RETURN, PROFILERC, and SESSIONEND records in the response files.

### Successful completion code

This section describes the return code for successful completion of the process.

---

00000      Session completed successfully.

**Explanation:** Expedite completed successfully.

**User Response:** No user action is required.

## Message command file syntax errors

The following section describes the return codes for message command file (INMSG) syntax errors.

---

02014      ALIAS invalid on LIST command.

**Explanation:** You specified an invalid value for an ALIAS parameter in the LIST command. The first character of the ALIAS is the destination table type and must be blank, g, o, or p. The last three characters of the ALIAS are the destination table ID and must be one to three alphanumeric characters. If the first character is blank, then the last three characters of the ALIAS parameter must also be blank.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which command produced the error. Correct the LIST command in the message command file, INMSG, and retry the program.

---

02032      LISTNAME missing on LIST command.

**Explanation:** The LISTNAME parameter in the LIST command is missing or blank. A LISTNAME must be specified.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which LIST command produced the error. Correct the LIST command in the message command file, INMSG, and retry the program.

---

02062      FUNCTION missing or invalid on LIST command.

**Explanation:** The FUNCTION parameter in the LIST command is missing, or you have specified an invalid value. The FUNCTION value must be a, d, e, or n.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which LIST command produced the error. Correct the LIST command in the message command file, INMSG, and retry the program.

---

02064      Invalid entries on LIST command.

**Explanation:** Either entries were specified in the LIST command with an 'E' specified for the FUNCTION parameter, or no entries were specified for one of the other FUNCTION values.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which LIST command produced the error. Correct the LIST command in the message command file, INMSG, and retry the program.

---

02066      LISTTYPE invalid on LIST command.

**Explanation:** The LISTTYPE value in the LIST command is invalid. The LISTTYPE value must be t, p, a, or g.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which command produced the error. Correct the LIST command in the message command file, INMSG, and retry the program.

---

02098 Invalid command sequence on LIST command.

**Explanation:** A LISTNAME, FUNCTION, or LISTTYPE parameter was specified more than once on the LIST command. You can only specify these parameters once in each LIST command.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which command produced the error. Correct the LIST command in the message command file, INMSG, and retry the program.

---

02099 Incomplete destination on LIST command.

**Explanation:** The list ended or a new destination was started before a previous destination was finished. This is usually caused by a missing parameter somewhere in the list. List entries are made up of either an ACCOUNT and USERID, or an ALIAS and ALIASNAME. Therefore, you must specify each list entry component next to its counterpart. For example, an ACCOUNT parameter should be entered next to a USERID parameter. If you specify the SYSID parameter, you must specify it either before the ACCOUNT and USERID parameters to which it belongs, or between them. It cannot follow the ACCOUNT and USERID parameters for that SYSID.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which command produced the error. Correct the LIST command in the message command file, INMSG, and retry the program.

---

2202 KEYRINGFILE missing on START command.

**Explanation:** You must specify the KEYRINGFILE parameter on the START command.

**User Response:** When making an SSL connection, there must be a KDB file containing the certificate to allow the connection. This filename is specified in the KEYRINGFILE parameter. It is a required parameter when SSL is enabled.

Correct the START command in the message command file, INMSG, to specify the filename of the KEYRINGFILE parameter and retry the program.

---

2204 KEYRINGFILE or APPLICID missing on START command.

**Explanation:** You must specify either the KEYRINGFILE or the APPLICID parameter on the START command.

**User Response:** When making an SSL connection, there must be a KDB file containing the certificate, or an application id defined that is tied to the certificate, to allow the connection. This filename or application id is specified in the KEYRINGFILE or APPLICID parameter, respectively. One of those parameters are required when SSL is enabled.

**User Response:** Correct the START command in the message command file, INMSG, to specify the filename of the KEYRINGFILE parameter or the application id of the APPLICID parameter and retry the program.

---

2205 KEYRINGPASSWORD missing on START command.

**Explanation:** The key database or Karin file specified in the KEYRINGFILE parameter, on the START command, requires a password.

**User Response:** Correct the START command in the message command file, INMSG, to specify the correct KEYRINGPASSWORD parameter associated with the key database specified in the KEYRINGFILE parameter and retry the program.

---

02206 Both KEYRINGFILE or APPLICID specified on START command.

**Explanation:** You specified both the KEYRINGFILE and the APPLICID parameters on the START command.

**User Response:** When making an SSL connection, there must be a KDB file containing the certificate, or an application id defined that is tied to the certificate, to allow the connection. This filename or application id is specified in the KEYRINGFILE or APPLICID parameter, respectively. Only one of those parameters can be specified when SSL is enabled.

Correct the START command in the message command file, INMSG, to specify either the filename of the KEYRINGFILE parameter or the application id of the APPLICID parameter and retry the program.

---

02207 Both APPLICID and KEYRINGPASSWORD specified on START command.

**Explanation:** You specified both the APPLICID and the KEYRINGPASSWORD parameters on the START command.

**User Response:** When making an SSL connection, there must be a KDB file containing the certificate and a password for that file, or an application id defined that is tied to the certificate, to allow the connection. If you specified the KEYRINGFILE parameter, then you need the KEYRINGPASSWORD parameter, and if you specified the APPLICID parameter, then you do not need the KEYRINGPASSWORD. Correct the START command in the message command file, INMSG, and retry the program.

---

02402 No destination on CANCEL command.

**Explanation:** You must specify a destination for this command. You can use ACCOUNT and USERID; ALIAS and ALIASNAME; or LISTNAME.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which CANCEL command produced the error. Correct the CANCEL command in the message command file, INMSG, and retry the program.

---

02404 Multiple destinations on CANCEL command.

**Explanation:** You can specify only one destination in the CANCEL command.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which CANCEL command produced the error. Correct the CANCEL command in the message command file, INMSG, and retry the program.

---

02406 Insufficient destination on CANCEL command.

**Explanation:** You must specify a complete destination for this command. You can use ACCOUNT and USERID; ALIAS and ALIASNAME; or LISTNAME.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which CANCEL command produced the error. Correct the CANCEL command in the message command file, INMSG, and retry the program.

---

02414 ALIAS invalid on CANCEL command.

**Explanation:** You specified an invalid value for the ALIAS parameter in the CANCEL command. The first character of the ALIAS parameter is the destination table type and must be blank, g, o, or p. The last three characters of the ALIAS parameter are the destination table ID and must be one to three alphanumeric characters. If the first character is blank, then the last three characters of the ALIAS parameter must also be blank.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which CANCEL command produced the error. Correct the CANCEL command in the message command file, INMSG, and retry the program.

---

02444 PRIORITY invalid on CANCEL command.

**Explanation:** You specified an invalid value for the PRIORITY parameter in the CANCEL command. The value of the PRIORITY parameter must be blank or p.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which CANCEL command produced the error. Correct the CANCEL command in the message command file, INMSG, and retry the program.

---

02450 ACK invalid on CANCEL command.

**Explanation:** You specified an invalid value for the ACK parameter in the CANCEL command. The value of the ACK parameter must be blank, h, or t.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which CANCEL command produced the error. Correct the CANCEL command in the message command file, INMSG, and retry the program.

---

02464 STARTDATE invalid on CANCEL command.

**Explanation:** You specified an invalid value for the STARTDATE parameter in the CANCEL command. The format must be YYMMDD, where YY is the last two digits of the year, MM is the two-digit month, and DD is the two-digit day; or YYYYMMDD where YYYY is a four-digit year.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which CANCEL command produced the error. Correct the CANCEL command in the message command file, INMSG, and retry the program.

---

02466 STARTTIME invalid on CANCEL command.

**Explanation:** You specified an invalid value for the STARTTIME parameter in the CANCEL command. The format must be HHMMSS, where HH is the two-digit hour, MM is the two-digit minutes, and SS is the two-digit seconds.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which CANCEL command produced the error. Correct the CANCEL command in the message command file, INMSG, and retry the program.



---

02468 ENDDATE invalid on CANCEL command.

**Explanation:** You specified an invalid value for the ENDDATE parameter in the CANCEL command. The format must be YYMMDD, where YY is the last two-digits of the year, MM is the two-digit month, and DD is the two-digit day; or YYYYMMDD where YYYY is a four-digit year.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which CANCEL command produced the error. Correct the CANCEL command in the message command file, INMSG, and retry the program.

---

02470 ENDTIME invalid on CANCEL command.

**Explanation:** You specified an invalid value for the ENDTIME parameter in the CANCEL command. The format must be HHMMSS, where HH is the two-digit hour, MM is the two-digit minutes, and SS is the two-digit seconds.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which CANCEL command produced the error. Correct the CANCEL command in the message command file, INMSG, and retry the program.

---

02472 TIMEZONE invalid on CANCEL command.

**Explanation:** You specified an invalid value for the TIMEZONE parameter in the CANCEL command. TIMEZONE must be either g or l.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which CANCEL command produced the error. Correct the CANCEL command in the message command file, INMSG, and retry the program.

---

02474 End date/time is before start date/time.

**Explanation:** You specified an end date and time in the CANCEL command that is before the start date and time. If you specified either date field with a two-digit year, a sliding window technique was used to decide the century indicator that is used when comparing the start and end dates. The two-digit year will be considered a 20xx year if the two digits are between 00 and the current year + 49. For example:

- In 1999, two-digit years starting with 00 through 48 will be considered 20xx years, while two-digit years starting with 49 through 99 will be considered 19xx years.
- In 2000, two-digit years starting with 00 through 49 will be considered 20xx years, while two-digit years starting with 50 through 99 will be considered 19xx years.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which CANCEL command produced the error. Correct the CANCEL command in the message command file, INMSG, and retry the program.

---

02604 Multiple destinations on AUDIT command.

**Explanation:** You can specify only one destination in the AUDIT command.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which AUDIT command produced the error. Correct the AUDIT command in the message command file, INMSG, and retry the program.

---

02606      Incomplete destination on AUDIT command.

**Explanation:** You specified an incomplete destination in the AUDIT command. If you specify a destination, you must use ACCOUNT and USERID; SYSID, ACCOUNT and USERID; or ALIAS and ALIASNAME.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which AUDIT command produced the error. Correct the AUDIT command in the message command file, INMSG, and retry the program.

---

02614      ALIAS invalid on AUDIT command.

**Explanation:** You specified an invalid value for the ALIAS parameter in the AUDIT command. The first character of the ALIAS parameter is the destination table type and must be blank, g, o, or p. The last three characters of the ALIAS parameter are the destination table ID and must be one to three alphanumeric characters. If the first character is blank, then the last three characters of the ALIAS parameter must also be blank.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which AUDIT command produced the error. Correct the AUDIT command in the message command file, INMSG, and retry the program.

---

02664      STARTDATE invalid on AUDIT command.

**Explanation:** You specified an invalid value for the STARTDATE parameter in the AUDIT command. The format must be YYMMDD, where YY is the last two digits of the year, MM is the two-digit month, and DD is the two-digit day; or YYYYMMDD where YYYY is a four-digit year.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which AUDIT command produced the error. Correct the AUDIT command in the message command file, INMSG, and retry the program.

---

02668      ENDDATE invalid on AUDIT command.

**Explanation:** You specified an invalid value for the ENDDATE parameter in the AUDIT command. The format must be YYMMDD, where YY is the last two digits of the year, MM is the two-digit month, and DD is the two-digit day; or YYYYMMDD where YYYY is a four-digit year.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which AUDIT command produced the error. Correct the AUDIT command in the message command file, INMSG, and retry the program.

---

02670      ALTACCT invalid on AUDIT command.

**Explanation:** You specified ALTACCT and did not specify ALTUSERID, or you specified ALTACCT and level 1 or 2. If you specify ALTACCT, you must specify ALTUSERID. ALTACCT is only supported for level 3 or higher.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which AUDIT command produced the error. Correct the AUDIT command in the message command file, INMSG, and retry the program.

02672      TIMEZONE invalid on AUDIT command.

**Explanation:** You specified an invalid value in the TIMEZONE parameter on the AUDIT command. TIMEZONE must be either g or l.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which AUDIT command produced the error. Correct the AUDIT command in the message command file, INMSG, and retry the program.

---

02674      STATUS invalid on AUDIT command.

**Explanation:** You specified an invalid value for the STATUS parameter in the AUDIT command. STATUS must be blank, u, p, or d.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which AUDIT command produced the error. Correct the AUDIT command in the message command file, INMSG, and retry the program.

---

02676      MSGTYPE invalid on AUDIT command.

**Explanation:** You specified an invalid value for the MSGTYPE parameter in the AUDIT command. MSGTYPE must be s, r, or b.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which AUDIT command produced the error. Correct the AUDIT command in the message command file, INMSG, and retry the program.

---

02678      End date is before start date.

**Explanation:** You specified an end date in the AUDIT command that is before the start date. If you specified either date field with a two-digit year, a sliding window technique was used to decide the century indicator that is used when comparing the start and end dates. The two-digit year will be considered a 20xx year if the two digits are between 00 and the current year + 49. For example:

- In 1999, two-digit years starting with 00 through 48 will be considered 20xx years, while two-digit years starting with 49 through 99 will be considered 19xx years.
- In 2000, two-digit years starting with 00 through 49 will be considered 20xx years, while two-digit years starting with 50 through 99 will be considered 19xx years.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which AUDIT command produced the error. Correct the AUDIT command in the message command file, INMSG, and retry the program.

---

02684      LEVEL invalid on AUDIT command.

**Explanation:** You specified an invalid value for the LEVEL parameter in the AUDIT command. LEVEL must be 1, 2 or 3.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which AUDIT command produced the error. Correct the AUDIT command in the message command file, INMSG, and retry the program.

---

02802      No destination specified on SEND command.

**Explanation:** You must specify a destination in the SEND command.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which SEND command produced the error. Correct the SEND command in the message command file, INMSG, and retry the program.

---

02804      Multiple destinations on SEND command.

**Explanation:** You specified multiple destinations in a SEND command. You can only specify one destination.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which SEND command produced the error. Correct the SEND command in the message command file, INMSG, and retry the program.

---

02806      Incomplete destination on SEND command.

**Explanation:** You specified an incomplete destination in the SEND command. The destination must be an ACCOUNT and USERID; SYSID, ACCOUNT and USERID; ALIAS and ALIASNAME; or LISTNAME.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which SEND command produced the error. Correct the SEND command in the message command file, INMSG, and retry the program.

---

02814      ALIAS invalid on SEND command.

**Explanation:** You specified an invalid value for the ALIAS parameter in the SEND command. The first character of the ALIAS parameter is the destination table type and must be blank, g, o, or p. The last three characters of the ALIAS parameter are the destination table ID and must be one to three alphanumeric characters. If the first character is blank, then the last three characters of the ALIAS parameter must also be blank.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which SEND command produced the error. Correct the SEND command in the message command file, INMSG, and retry the program.

---

02830      No FILEID specified on SEND command.

**Explanation:** You must specify a FILEID in the SEND command.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which SEND command produced the error. Correct the SEND command in the message command file, INMSG, and retry the program.

---

02836      FORMAT invalid on SEND command.

**Explanation:** You specified an invalid value for the FORMAT parameter in the SEND command. FORMAT must be y or n. You cannot specify 'Y' for FORMAT with 'B' for DATATYPE or 'Y' for DELIMIT.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which SEND command produced the error. Correct the SEND command in the message command file, INMSG, and retry the program.

---

02844      PRIORITY invalid on SEND command.

**Explanation:** You specified an invalid value for the PRIORITY parameter in the SEND command. PRIORITY must be blank, i, or p.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which SEND command produced the error. Correct the SEND command in the message command file, INMSG, and retry the program.

---

02846      MODE invalid on SEND command.

**Explanation:** You specified an invalid value for the MODE parameter in the SEND command. MODE must be blank or t.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which SEND command produced the error. Correct the SEND command in the message command file, INMSG, and retry the program.

---

02848      CHARGE invalid on SEND command.

**Explanation:** You specified an invalid value for the CHARGE parameter in the SEND command. CHARGE must be a numeric character from 1 to 6.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which SEND command produced the error. Correct the SEND command in the message command file, INMSG, and retry the program.

---

02850      ACK invalid on SEND command.

**Explanation:** You specified an invalid value for the ACK parameter in the SEND command. ACK must be blank, a, b, c, d, e, f, or r.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which SEND command produced the error. Correct the SEND command in the message command file, INMSG, and retry the program.

---

02872      VERIFY invalid on SEND command.

**Explanation:** You specified an invalid value for the VERIFY parameter in the SEND command. VERIFY must be y, n, or f.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which SEND command produced the error. Correct the SEND command in the message command file, INMSG, and retry the program.

---

02874      RETAIN invalid on SEND command.

**Explanation:** You specified an invalid value for the RETAIN parameter in the SEND command. RETAIN must be a numeric value from 0 to 180.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which SEND command produced the error. Correct the SEND command in the message command file, INPRO, and retry the program.

---

02876      DATATYPE invalid on SEND command.

**Explanation:** You specified an invalid value for the DATATYPE parameter in the SEND command. DATATYPE must be e or b.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which SEND command produced the error. Correct the SEND command in the message command file, INMSG, and retry the program.

---

02878      DELIMIT invalid on SEND command.

**Explanation:** You specified an invalid value for the DELIMIT parameter in the SEND command. DELIMIT must be c, l, u, or n.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which SEND command produced the error. Correct the SEND command in the message command file, INMSG, and retry the program.

---

02890      CRLFEOF invalid on SEND command.

**Explanation:** You specified an invalid value for the CRLFEOF parameter in the SEND command. CRLFEOF must contain 6 hexadecimal characters.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which SEND command produced the error. Correct the SEND command in the message command file, INMSG, and retry the program.

---

02892      TRUNCATE invalid on SEND command.

**Explanation:** You specified an invalid value for the TRUNCATE parameter in the SEND command. TRUNCATE must be y or n.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which SEND command produced the error. Correct the SEND command in the message command file, INMSG, and retry the program.

---

02898      COMPRESS invalid on SEND command.

**Explanation:** You specified an invalid value for the COMPRESS parameter in the SEND command. COMPRESS must be y, n, or t.

**User Response:** Check the message response file, OUTMSG, or the response work file, OUTWORK, to determine which SEND command produced the error. Correct the SEND command in the message command file, INMSG, and retry the program.

---

02899      SELECTRCV invalid on SEND command.

**Explanation:** You specified an invalid value for the SELECTRCV parameter in the SEND command. SELECTRCV must be f, n, or blank.

**User Response:** Check the message response file, OUTMSG, or the response work file, OUTWORK, to determine which SENDSTREAM command produced the error. Correct the SENDSTREAM command in the message command file, INMSG, and retry the program.

---

02902 No destination specified on SENDSTREAM command.

**Explanation:** You must specify a destination in the SENDSTREAM command.

**User Response:** Check the message response file, OUTMSG, or the response work file, OUTWORK, to determine which SENDSTREAM command produced the error. Correct the SENDSTREAM command in the message command file, INMSG, and retry the program.

---

02904 Multiple destinations on SENDSTREAM command.

**Explanation:** You specified multiple destinations in a SENDSTREAM command. You can specify only one destination.

**User Response:** Check the message response file, OUTMSG, or the response work file, OUTWORK, to determine which SENDSTREAM command produced the error. Correct the SENDSTREAM command in the message command file, INMSG, and retry the program.

---

02906 Incomplete destination on SENDSTREAM command.

**Explanation:** You specified an incomplete destination in the SENDSTREAM command. The destination must be ALIAS and ALIASNAME; ACCOUNT and USERID; SYSID with ACCOUNT and USERID; or LISTNAME.

**User Response:** Check the message response file, OUTMSG, or the response work file, OUTWORK, to determine which SENDSTREAM command produced the error. Correct the SENDSTREAM command in the message command file, INMSG, and retry the program.

---

02914 ALIAS invalid on SENDSTREAM command.

**Explanation:** You specified an invalid value for the ALIAS parameter in the SENDSTREAM command. The first character of the ALIAS parameter is the destination table type, and it must be blank, g, o, or p. The last three characters of the ALIAS parameter indicate the destination table, and they must be 1 to 3 alphanumeric characters. If the first character is blank, the last three characters of the ALIAS parameter must also be blank.

**User Response:** Check the message response file, OUTMSG, or the response work file, OUTWORK, to determine which SENDSTREAM command produced the error. Correct the SENDSTREAM command in the message command file, INMSG, and retry the program.

---

02944 PRIORITY invalid on SENDSTREAM command.

**Explanation:** You specified an invalid value for the PRIORITY parameter in the SENDSTREAM command. PRIORITY must be blank, i, or p.

**User Response:** Check the message response file, OUTMSG, or the response work file, OUTWORK, to determine which SENDSTREAM command produced the error. Correct the SENDSTREAM command in the message command file, INMSG, and retry the program.

---

02946      MODE invalid on SENDSTREAM command.

**Explanation:** You specified an invalid value for the MODE parameter in the SENDSTREAM command. MODE must be blank or t.

**User Response:** Check the message response file, OUTMSG, or the response work file, OUTWORK, to determine which SENDSTREAM command produced the error. Correct the SENDSTREAM command in the message command file, INMSG, and retry the program.

---

02948      CHARGE invalid on SENDSTREAM command.

**Explanation:** You specified an invalid value for the CHARGE parameter in the SENDSTREAM command. CHARGE must be a numeric character from 1 to 6.

**User Response:** Check the message response file, OUTMSG, or the response work file, OUTWORK, to determine which SENDSTREAM command produced the error. Correct the SENDSTREAM command in the message command file, INMSG, and retry the program.

---

02950      ACK invalid on SENDSTREAM command.

**Explanation:** You specified an invalid value for the ACK parameter in the SENDSTREAM command. ACK must be blank, a, b, c, d, e, f, or r.

**User Response:** Check the message response file, OUTMSG, or the response work file, OUTWORK, to determine which SENDSTREAM command produced the error. Correct the SENDSTREAM command in the message command file, INMSG, and retry the program.

---

02964      No ENDSTR parameter specified on SENDSTREAM command.

**Explanation:** You must specify a value for the ENDSTR parameter in the SENDSTREAM command.

**User Response:** Check the message response file, OUTMSG, or the response work file, OUTWORK, to determine which SENDSTREAM command produced the error. Correct the SENDSTREAM command in the message command file, INMSG, and retry the program.

---

02972      VERIFY invalid on SENDSTREAM command.

**Explanation:** You specified an invalid value for the VERIFY parameter in the SENDSTREAM command. VERIFY must be y, n, or f.

**User Response:** Check the message response file, OUTMSG, or the response work file, OUTWORK, to determine which SENDSTREAM command produced the error. Correct the SENDSTREAM command in the message command file, INMSG, and retry the program.

---

02974      RETAIN invalid on SENDSTREAM command.

**Explanation:** You specified an invalid value for the RETAIN parameter in the SENDSTREAM command. RETAIN must be a numeric value from 0 to 180.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which SENDSTREAM command produced the error. Correct the SENDSTREAM command in the message command file, INMSG, and retry the program.



02976      DATATYPE invalid on SENDSTREAM command.

**Explanation:** You specified an invalid value for the DATATYPE parameter in the SENDSTREAM command. DATATYPE must be e or b.

**User Response:** Check the message response file, OUTMSG, or the response work file, OUTWORK, to determine which SENDSTREAM command produced the error. Correct the SENDSTREAM command in the message command file, INMSG, and retry the program.

---

02993      EDITYPE invalid on SENDSTREAM command.

**Explanation:** You specified an invalid value for the EDITYPE parameter in the SENDSTREAM command. EDITYPE must be n, e, t, u, or x.

**User Response:** Check the message response file, OUTMSG, or the response work file, OUTWORK, to determine which SENDSTREAM command produced the error. Correct the SENDSTREAM command in the message command file, INMSG, and retry the program.

---

02994      SENDSTREAM command only valid for session-level recovery.

**Explanation:** The SENDSTREAM command cannot be used for checkpoint-level, file-level, or user-initiated recovery. It is only valid with session-level recovery.

**User Response:** Either use a SEND command instead of the SENDSTREAM command, and put data in a separate file, or change the RECOVERY parameter in the profile to use session-level recovery.

---

02996      End of data for SENDSTREAM command not found.

**Explanation:** The value specified by the ENDSTR parameter was not found in the data before the end of the file was reached. This value must be at the end of the data to indicate where the data ends.

**User Response:** Check the command to be sure that the ENDSTR value is correct and that it appears at the end of the data exactly as it appears in the parameter value. Note that the search for the ENDSTR value is sensitive to uppercase and lowercase differences. Correct the command file and try again.

---

02998      COMPRESS invalid on SENDSTREAM command.

**Explanation:** You specified an invalid value for the COMPRESS parameter on the SENDSTREAM command. COMPRESS must be y, n, or t.

**User Response:** Check the message response file, OUTMSG, or the response work file, OUTWORK, to determine which SENDSTREAM command produced the error. Correct the SENDSTREAM command in the message command file, INMSG, and retry the program.

---

02999      SELECTRCV invalid on SENDSTREAM command.

**Explanation:** You specified an invalid value for the SELECTRCV parameter in the SENDSTREAM command. SELECTRCV must be f, n, or blank.

**User Response:** Check the message response file, OUTMSG, or the response work file, OUTWORK, to determine which SENDSTREAM command produced the error. Correct the SENDSTREAM command in the message command file, INMSG, and retry the program.

---

03030 No FILEID specified on SENDEDI command.

**Explanation:** You must specify a FILEID in the SENDEDI command.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which SENDEDI command produced the error. Correct the SENDEDI command in the message command file, INMSG, and retry the program.

---

03044 PRIORITY invalid on SENDEDI command.

**Explanation:** You specified an invalid value for the PRIORITY parameter in the SENDEDI command. PRIORITY must be blank, i, or p.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which SENDEDI command produced the error. Correct the SENDEDI command in the message command file, INMSG, and retry the program.

---

03046 MODE invalid on SENDEDI command.

**Explanation:** You specified an invalid value for the MODE parameter in the SENDEDI command. MODE must be blank or t.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which SENDEDI command produced the error. Correct the SENDEDI command in the message command file, INMSG, and retry the program.

---

03048 CHARGE invalid on SENDEDI command.

**Explanation:** You specified an invalid value for the CHARGE parameter in the SENDEDI command. CHARGE must be a numeric character from 1 to 6.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which SENDEDI command produced the error. Correct the SENDEDI command in the message command file, INMSG, and retry the program.

---

03050 ACK invalid on SENDEDI command.

**Explanation:** You specified an invalid value for the ACK parameter in the SENDEDI command. ACK must be blank, a, b, c, d, e, f, or r.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which SENDEDI command produced the error. Correct the SENDEDI command in the message command file, INMSG, and retry the program.

---

03072 VERIFY invalid on SENDEDI command.

**Explanation:** You specified an invalid value for the VERIFY parameter in the SENDEDI command. VERIFY must be y, c, n, f, or g.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which SENDEDI command produced the error. Correct the SENDEDI command in the message command file, INMSG, and retry the program.

03074      RETAIN invalid on SENDEDI command.

**Explanation:** You specified an invalid value for the RETAIN parameter in the SENDEDI command. RETAIN must be a numeric value from 0 to 180.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which SENDEDI command produced the error. Correct the SENDEDI command in the message command file, INMSG, and retry the program.

---

03098      COMPRESS invalid on SENDEDI command.

**Explanation:** You specified an invalid value for the COMPRESS parameter on the SENDEDI command. COMPRESS must be y, n, or t.

**User Response:** Check the message response file, OUTMSG, or the response work file, OUTWORK, to determine which SENDEDI command produced the error. Correct the SENDEDI command in the message command file, INMSG, and retry the program.

---

03099      SELECTRCV invalid on SENDEDI command.

**Explanation:** You specified an invalid value for the SELECTRCV parameter in the SENDEDI command. SELECTRCV must be f, n, or blank.

**User Response:** Check the message response file, OUTMSG, or the response work file, OUTWORK, to determine which SENDEDI command produced the error. Correct the SENDEDI command in the message command file, INMSG, and retry the program.

---

03204      Multiple sources on RECEIVE command.

**Explanation:** You specified multiple sources in the RECEIVE command. You can specify only one source.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which RECEIVE command produced the error. Correct the RECEIVE command in the message command file, INMSG, and retry the program.

---

03206      Incomplete source specified on RECEIVE command.

**Explanation:** You specified an incomplete source for the RECEIVE command. If you specify a source, you must use ACCOUNT and USERID; SYSID, ACCOUNT and USERID; ALIAS and ALIASNAME; or LISTNAME.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which RECEIVE command produced the error. Correct the RECEIVE command in the message command file, INMSG, and retry the program.

---

03214      ALIAS invalid on RECEIVE command.

**Explanation:** You specified an invalid value for the ALIAS parameter in the RECEIVE command. The first character of the ALIAS parameter is the destination table type, and it must be blank, g, o, or p. The last three characters of the ALIAS parameter are the destination table ID, and they must be 1 to 3 alphanumeric characters. If the first character is blank, the last three characters of the ALIAS parameter must also be blank.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which RECEIVE command produced the error. Correct the RECEIVE command in the message command file, INMSG, and retry the program.

---

03230 No FILEID specified on RECEIVE command.

**Explanation:** You must specify a FILEID in the RECEIVE command.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which RECEIVE command produced the error. Correct the RECEIVE command in the message command file, INMSG, and retry the program.

---

03236 FORMAT invalid on RECEIVE command.

**Explanation:** You specified an invalid value for the FORMAT parameter in the RECEIVE command. FORMAT must be y or n.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which RECEIVE command produced the error. Correct the RECEIVE command in the message command file, INMSG, and retry the program.

---

03242 ALLFILES invalid on RECEIVE command.

**Explanation:** You specified an invalid value for the ALLFILES parameter in the RECEIVE command. ALLFILES must be y or n.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which RECEIVE command produced the error. Correct the RECEIVE command in the message command file, INMSG, and retry the program.

---

03246 EDIOPT invalid on RECEIVE command.

**Explanation:** You specified an invalid value for the EDIOPT parameter in the RECEIVE command. EDIOPT must be y or n.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which RECEIVE command produced the error. Correct the RECEIVE command in the message command file, INMSG, and retry the program.

---

03256 REQUEUED invalid on RECEIVE command.

**Explanation:** You specified an invalid value for the REQUEUED parameter in the RECEIVE command. REQUEUED must be y or n. If you specify a source, REQUEUED cannot be y.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which RECEIVE command produced the error. Correct the RECEIVE command in the message command file, INMSG, and retry the program.

---

03258 AUTOEDI invalid on RECEIVE command.

**Explanation:** You specified an invalid value for the AUTOEDI parameter in the RECEIVE command. AUTOEDI must be y or n.

---

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which RECEIVE command produced the error. Correct the RECEIVE command in the message command file, INMSG, and retry the program.

---

03260 DELIMITED invalid on RECEIVE command.

**Explanation:** You specified an invalid value for the DELIMITED parameter in the RECEIVE command. DELIMITED must be c, l, or n.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which RECEIVE command produced the error. Correct the RECEIVE command in the message command file, INMSG, and retry the program.

---

03262 DLMOVERRIDE invalid on RECEIVE command.

**Explanation:** You specified an invalid value for the DLMOVERRIDE parameter in the RECEIVE command. DLMOVERRIDE must be y or n.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which RECEIVE command produced the error. Correct the RECEIVE command in the message command file, INMSG, and retry the program.

---

03264 ENDSTR invalid on RECEIVE command.

**Explanation:** You specified an invalid value for the ENDSTR parameter in the RECEIVE command. The ENDSTR parameter must be 1 to 79 characters. If it includes an s: or r: to indicate whether it starts on a new record, these characters can be added to the 79 characters for a maximum length of 81.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which RECEIVE command produced the error. Correct the RECEIVE command in the message command file, INMSG, and retry the program.

---

03266 MSGKEY invalid on RECEIVE command.

**Explanation:** You specified an invalid value for the MSGKEY parameter in the RECEIVE command. MSGKEY must be 20 hex characters.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which RECEIVE command produced the error. Correct the message command file, INMSG, and retry the program.

---

03270 STARTDATE invalid on RECEIVE command.

**Explanation:** You specified an invalid value for the STARTDATE parameter in the RECEIVE command. The format must be *YYMMDD*, where *YY* is the last two digits of the year, *MM* is the two-digit month, and *DD* is the two-digit day; or *YYYYMMDD* where *YYYY* is a four-digit year.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which RECEIVE command produced the error. Correct the message command file, INMSG, and retry the program.

---

03271      STARTTIME invalid on RECEIVE command.

**Explanation:** You specified an invalid value for the STARTTIME parameter in the RECEIVE command. The format must be *HHMMSS*, where *hh* is the two-digit hour, *mm* is the two-digit minutes, and *ss* is the two-digit seconds.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which RECEIVE command produced the error. Correct the message command file, INMSG, and retry the program.

---

03272      ENDDATE invalid on RECEIVE command.

**Explanation:** You specified an invalid value for the ENDDATE parameter in the RECEIVE command. The format must be *YYMMDD*, where *yy* is the last two digits of the year, *mm* is the two-digit month, and *dd* is the two-digit day, or *YYYYMMDD* where *yyyy* is a four-digit year.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which RECEIVE command produced the error. Correct the message command file, INMSG, and retry the program.

---

03273      ENDTIME invalid on RECEIVE command.

**Explanation:** You specified an invalid value for the ENDTIME parameter in the RECEIVE command. The format must be *HHMMSS*, where *hh* is the two-digit hour, *mm* is the two-digit minutes, and *ss* is the two-digit seconds.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which RECEIVE command produced the error. Correct the message command file, INMSG, and retry the program.

---

03274      TIMEZONE invalid on RECEIVE command.

**Explanation:** You specified an invalid value for the TIMEZONE parameter in the RECEIVE command. TIMEZONE must be either g or l.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which RECEIVE command produced the error. Correct the message command file, INMSG, and retry the program.

---

03275      End date/time is before start date/time.

**Explanation:** You specified an end date and time in the RECEIVE command that is before the start date and time. If you specified either date field with a two-digit year, a sliding window technique was used to decide the century indicator that is used when comparing the start and end dates. The two-digit year will be considered a 20xx year if the two digits are between 00 and the current year + 49. For example:

- In 1999, two-digit years starting with 00 through 48 will be considered 20xx years, while two-digit years starting with 49 through 99 will be considered 19xx years.
- In 2000, two-digit years starting with 00 through 49 will be considered 20xx years, while two-digit years starting with 50 through 99 will be considered 19xx years.

---

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which RECEIVE command produced the error. Correct the message command file, INMSG, and retry the program.

---

03276 RESRECL invalid on RECEIVE command.

**Explanation:** You specified an invalid value for the RESRECL parameter in the RECEIVE command. RESRECL must be s or e. If you are using checkpoint-level recovery, only s is valid.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which RECEIVE command produced the error. Correct the message command file, INMSG, and retry the program.

---

03284 NONEDIONLY invalid on RECEIVE command.

**Explanation:** You specified an invalid value for the NONEDIONLY parameter in the RECEIVE command. NONEDIONLY must be y or n.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which RECEIVE command produced the error. Correct the RECEIVE command in the message command file, INMSG, and retry the program.

---

03304 Multiple sources on RECEIVESTREAM command.

**Explanation:** You specified multiple sources in the RECEIVESTREAM command. You can specify only one source.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which RECEIVESTREAM command produced the error. Correct the RECEIVESTREAM command in the message command file, INMSG, and retry the program.

---

03306 Incomplete source specified on RECEIVESTREAM command.

**Explanation:** You specified an incomplete source for this command. If you specify a source, you must use the parameters ACCOUNT and USERID; SYSID, ACCOUNT, and USERID; ALIAS and ALIASNAME; or LISTNAME.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which RECEIVESTREAM command produced the error. Correct the RECEIVESTREAM command in the message command file, INMSG, and retry the program.

---

03314 ALIAS invalid on RECEIVESTREAM command.

**Explanation:** You specified an invalid value for the ALIAS parameter in the RECEIVESTREAM command. The first character of the ALIAS parameter is the destination table type, and it must be blank, g, o, or p. The last three characters of the ALIAS parameter are the destination table ID, and they must be 1 to 3 alphanumeric characters. If the first character is blank, the last three characters of the ALIAS parameter must also be blank.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which RECEIVESTREAM command produced the error. Correct the RECEIVESTREAM command in the message command file, INMSG, and retry the program.

---

03342 ALLFILES invalid on RECEIVESTREAM command.

**Explanation:** You specified an invalid value for the ALLFILES parameter in the RECEIVESTREAM command. ALLFILES must be y or n.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which RECEIVESTREAM command produced the error. Correct the RECEIVESTREAM command in the message command file, INMSG, and retry the program.

---

03356 REQUEUED invalid on RECEIVESTREAM command.

**Explanation:** You specified an invalid value for the REQUEUED parameter in the RECEIVESTREAM command. REQUEUED must be y or n. If you specify a source, REQUEUED cannot be y.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which RECEIVESTREAM command produced the error. Correct the RECEIVESTREAM command in the message command file, INMSG, and retry the program.

---

03364 ENDSTR missing or invalid on RECEIVESTREAM command.

**Explanation:** You either did not specify the ENDSTR parameter in the RECEIVESTREAM command, or you specified an invalid value. The ENDSTR parameter must be 1 to 79 characters. If it includes s: or r: to indicate whether it starts on a new record, these characters can be added to the 79 characters for a maximum length of 81.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which RECEIVESTREAM command produced the error. Correct the RECEIVESTREAM command in the message command file, INMSG, and retry the program.

---

03366 MSGKEY invalid on RECEIVESTREAM command.

**Explanation:** You specified an invalid value for the MSGKEY parameter in the RECEIVESTREAM command. MSGKEY must either be blank or contain 20 hexadecimal characters.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which RECEIVESTREAM command produced the error. Correct the RECEIVESTREAM command in the message command file, INMSG, and retry the program.

---

03370 STARTDATE invalid on RECEIVESTREAM command.

**Explanation:** You specified an invalid value for the STARTDATE parameter in the RECEIVESTREAM command. The format must be *YYMMDD*, where *yy* is the last two digits of the year, *mm* is the two-digit month, and *dd* is the two-digit day; or *YYYYMMDD* where *yyyy* is a four-digit year.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which RECEIVESTREAM command produced the error. Correct the RECEIVESTREAM command in the message command file, INMSG, and retry the program.

---

03371 STARTTIME invalid on RECEIVESTREAM command.

**Explanation:** You specified an invalid value for the STARTTIME parameter in the RECEIVESTREAM command. The format must be *HHMMSS*, where *hh* is the two-digit hour, *mm* is the two-digit minutes, and *ss* is the two-digit seconds.



**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which RECEIVESTREAM command produced the error. Correct the RECEIVESTREAM command in the message command file, INMSG, and retry the program.

---

03372      ENDDATE invalid on RECEIVESTREAM command.

**Explanation:** You specified an invalid value for the ENDDATE parameter in the RECEIVESTREAM command. The format must be *YYMMDD*, where *yy* is the last two digits of the year, *mm* is the two-digit month, and *dd* is the two-digit day; or *YYYYMMDD* where *yyyy* is a four-digit year.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which RECEIVESTREAM command produced the error. Correct the RECEIVESTREAM command in the message command file, INMSG, and retry the program.

---

03373      ENDTIME invalid on RECEIVESTREAM command.

**Explanation:** You specified an invalid value for the ENDTIME parameter in the RECEIVESTREAM command. The format must be *HHMMSS*, where *hh* is the two-digit hour, *mm* is the two-digit minutes, and *ss* is the two-digit seconds.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which RECEIVESTREAM command produced the error. Correct the RECEIVESTREAM command in the message command file, INMSG, and retry the program.

---

03374      TIMEZONE invalid on RECEIVESTREAM command.

**Explanation:** You specified an invalid value for the TIMEZONE parameter in the RECEIVESTREAM command. TIMEZONE must be either *g* or *l*.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which RECEIVESTREAM command produced the error. Correct the RECEIVESTREAM command in the message command file, INMSG, and retry the program.

---

03375      End date/time is before start date/time.

**Explanation:** You specified an end date and time in the RECEIVESTREAM command that is before the start date and time. If you specified either date field with a two-digit year, a sliding window technique was used to decide the century indicator that is used when comparing the start and end dates. The two-digit year will be considered a 20xx year if the two digits are between 00 and the current year + 49. For example:

- In 1999, two-digit years starting with 00 through 48 will be considered 20xx years, while two-digit years starting with 49 through 99 will be considered 19xx years.
- In 2000, two-digit years starting with 00 through 49 will be considered 20xx years, while two-digit years starting with 50 through 99 will be considered 19xx years.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which RECEIVESTREAM command produced the error. Correct the RECEIVESTREAM command in the message command file, INMSG, and retry the program.

---

03394      RECEIVESTREAM command only valid for session-level recovery.

**Explanation:** The RECEIVESTREAM command cannot be used for checkpoint-level, file-level, or user-initiated recovery. It is only valid for session-level recovery.

**User Response:** Either use a RECEIVE instead of the RECEIVESTREAM command and receive data in a separate file, or change the RECOVERY parameter in the profile to use session-level recovery.

---

03404      Multiple sources on RECEIVEEDI command.

**Explanation:** You specified multiple sources in the RECEIVEEDI command. You can specify only one source.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which RECEIVEEDI command produced the error. Correct the RECEIVEEDI command in the message command file, INMSG, and retry the program.

---

03406      Incomplete source on RECEIVEEDI command.

**Explanation:** You specified an incomplete source in the RECEIVEEDI command. If you specify a source, you must use ACCOUNT and USERID; SYSID, ACCOUNT and USERID; ALIAS and ALIASNAME; or LISTNAME.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which RECEIVEEDI command produced the error. Correct the RECEIVEEDI command in the message command file, INMSG, and retry the program.

---

03414      ALIAS invalid on RECEIVEEDI command.

**Explanation:** You specified an invalid value for the ALIAS parameter in the RECEIVEEDI command. The first character of the ALIAS parameter is the destination table type, and it must be blank, g, o, or p. The last three characters of the ALIAS parameter are the destination table ID, and they must be one to three alphanumeric characters. If the first character is blank, then the last three characters of the ALIAS parameter must also be blank.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which RECEIVEEDI command produced the error. Correct the RECEIVEEDI command in the message command file, INMSG, and retry the program.

---

03430      No FILEID specified on RECEIVEEDI command.

**Explanation:** You must specify a FILEID parameter in the RECEIVEEDI command.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which RECEIVEEDI command produced the error. Correct the RECEIVEEDI command in the message command file, INMSG, and retry the program.

---

03442      ALLFILES invalid on RECEIVEEDI command.

**Explanation:** You specified an invalid value for the ALLFILES parameter in the RECEIVEEDI command. ALLFILES must be y or n.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which RECEIVEEDI command produced the error. Correct the RECEIVEEDI command in the message command file, INMSG, and retry the program.

---

03446 EDIOPT invalid on RECEIVEEDI command.

**Explanation:** You specified an invalid value for the EDIOPT parameter in the RECEIVEEDI command. EDIOPT must be y, n, or f.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which RECEIVEEDI command produced the error. Correct the RECEIVEEDI command in the message command file, INMSG, and retry the program.

---

03456 REQUEUED invalid on RECEIVEEDI command.

**Explanation:** You specified an invalid value for the REQUEUED parameter in the RECEIVEEDI command. REQUEUED must be y or n. If you specify a source, REQUEUED cannot be y.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which RECEIVEEDI command produced the error. Correct the RECEIVEEDI command in the message command file, INMSG, and retry the program.

---

03466 MSGKEY invalid on RECEIVEEDI command.

**Explanation:** You specified an invalid value for the MSGKEY parameter in the RECEIVEEDI command. MSGKEY must be 20 hex characters.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which RECEIVEEDI command produced the error. Correct the message command file, INMSG, and retry the program.

---

03470 STARTDATE invalid on RECEIVEEDI command.

**Explanation:** You specified an invalid value for the STARTDATE parameter in the RECEIVEEDI command. The format must be YYMMDD, where YY is the last two digits of the year, MM is the two-digit month, and DD is the two-digit day; or YYYYMMDD where YYYY is a four-digit year.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which RECEIVEEDI command produced the error. Correct the RECEIVEEDI command in the message command file, INMSG, and retry the program.

---

03471 STARTTIME invalid on RECEIVEEDI command.

**Explanation:** You specified an invalid value for the STARTTIME parameter in the RECEIVEEDI command. The format must be HHMMSS, where hh is the two-digit hour, mm is the two-digit minutes, and ss is the two-digit seconds.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which RECEIVEEDI command produced the error. Correct the message command file, INMSG, and retry the program.

---

03472 ENDDATE invalid on RECEIVEEDI command.

**Explanation:** You specified an invalid value for the ENDDATE parameter in the RECEIVEEDI command. The format must be YYMMDD, where YY is the last two digits of the year, MM is the two-digit month, and DD is the two-digit day; or YYYYMMDD where YYYY is a four-digit year.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which RECEIVEEDI command produced the error. Correct the message command file, INMSG, and retry the program.

---

03473 ENDTIME invalid on RECEIVEEDI command.

**Explanation:** You specified an invalid value for the ENDTIME parameter in the RECEIVEEDI command. The format must be HHMMSS, where hh is the two-digit hour, mm is the two-digit minutes, and ss is the two-digit seconds.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which RECEIVEEDI command produced the error. Correct the message command file, INMSG, and retry the program.

---

03474 TIMEZONE invalid on RECEIVEEDI command.

**Explanation:** You specified an invalid value for the TIMEZONE parameter in the RECEIVEEDI command. TIMEZONE must be either g or l.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which RECEIVEEDI command produced the error. Correct the message command file, INMSG, and retry the program.

---

03475 End date/time is before start date/time.

**Explanation:** You specified an end date and time in the RECEIVEEDI command that is before the start date and time. If you specified either date field with a two-digit year, a sliding window technique was used to decide the century indicator that is used when comparing the start and end dates. The two-digit year will be considered a 20xx year if the two digits are between 00 and the current year + 49. For example:

- In 1999, two-digit years starting with 00 through 48 will be considered 20xx years, while two-digit years starting with 49 through 99 will be considered 19xx years.
- In 2000, two-digit years starting with 00 through 49 will be considered 20xx years, while two-digit years starting with 50 through 99 will be considered 19xx years.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which RECEIVEEDI command produced the error. Correct the message command file, INMSG, and retry the program.

---

03484 EDIONLY invalid on RECEIVEEDI command.

**Explanation:** You specified an invalid value for the EDIONLY parameter in the RECEIVEEDI command. EDIONLY must be y or n.

---

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which RECEIVEEDI command produced the error. Correct the RECEIVEEDI command in the message command file, INMSG, and retry the program.

---

03600 Requested a session start when already in session.

**Explanation:** You requested that Expedite Base/400 do a session start when it was already in an Information Exchange session. If you use multiple START commands in a command file, you must end the previous Information Exchange session before you start the next one.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which command produced the error. Correct the message command file, INMSG, and retry the program.

---

03610 START command invalid with automatic session start.

**Explanation:** You specified the START command in the command file, but your profile, INPRO, indicates that the system should start the session automatically. If the session starts automatically, the START command is invalid.

**User Response:** Specify 'N' for AUTOSTART on the TRANSMIT command in the profile command file, INPRO, if you do not want the session started automatically. Otherwise, remove the START command from the message command file, INMSG, and retry the program.

---

03620 END command invalid with automatic session end.

**Explanation:** You specified the END command in the command file, but your profile indicates that the session should end automatically. If the system ends the session automatically, the END command is invalid.

**User Response:** Specify 'N' for AUTOEND on the TRANSMIT command in the profile command file, INPRO, if you do not want the session ended automatically. Otherwise, remove the END command from the message command file, INMSG, and retry the program.

---

03630 Session not started before first command.

**Explanation:** You specified a command in the message command file, INMSG, other than START, but were not currently in session.

**User Response:** Add a START command to the message command file, INMSG, or specify 'Y' for AUTOSTART on the TRANSMIT command in the profile command file, INPRO, and retry the program.

---

03640 No END command or automatic end for session.

**Explanation:** If you specify 'N' for AUTOEND on the TRANSMIT command, you must specify an END command in the command file.

**User Response:** Add an END command in the message command file, INMSG, or specify 'Y' for AUTOEND on the TRANSMIT command in the profile command file, INPRO. Retry the program.

---

03650 IE account/userid already in use.

**Explanation:** The account ID and user ID specified are already being used for another session.

**User Response:** When using checkpoint-level recovery, only one job should use an Information Exchange account ID and user ID combination at a time. The first job should be completed with a valid session end before any other jobs are submitted using that Information Exchange account ID and user ID combination.

---

03860      CDH invalid on QUERY command.

**Explanation:** You specified an invalid value for the CDH parameter in the QUERY command. CDH must be y or n.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which QUERY command produced the error. Correct the QUERY command in the message command file, INMSG, and retry the program.

---

04060      Missing ARCHIVEID on ARCHIVEMOVE command.

**Explanation:** You did not specify an ARCHIVEID value on the ARCHIVEMOVE command. This is a required parameter.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which ARCHIVEMOVE command produced the error. Correct the ARCHIVEMOVE command in the message command file, INMSG, and retry the program.

---

04110      MSGKEY missing on PURGE command.

**Explanation:** You did not specify the MSGKEY parameter on the PURGE command. This is a required parameter.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which PURGE command produced the error. Correct the PURGE command in the message command file, INMSG, and retry the program. The message key can be found on the AVAILABLE record in response to a QUERY command.

---

04232      LISTNAME missing or invalid on LISTVERIFY command.

**Explanation:** The LISTNAME value on the LISTVERIFY command must be specified if the FUNCTION value is b, c, l, r or s, but cannot be specified if the FUNCTION value is a or d.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which LISTVERIFY command produced the error. Correct the LISTVERIFY command in the message command file, INMSG, and retry the program.

---

04248      Invalid CHARGE on LISTVERIFY command.

**Explanation:** You specified an invalid value for the CHARGE parameter on the LISTVERIFY command. CHARGE must be a numeric character from 1 to 6.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which LISTVERIFY command produced the error. Correct the LISTVERIFY command in the message command file, INMSG, and retry the program.

---

04266      FUNCTION missing or invalid on LISTVERIFY command.

**Explanation:** The FUNCTION value on the LISTVERIFY command is invalid or was not specified. A FUNCTION value is required and must be a, b, c, d, l, r, or s.

---

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which LISTVERIFY command produced the error. Correct the LISTVERIFY command in the message command file, INMSG, and retry the program.

---

04460      STARTMSG missing or invalid on TESTMSG command.

**Explanation:** The STARTMSG value on the TESTMSG command is invalid or was not specified. A STARTMSG value is required and must be a numeric value from 0 to 5.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which TESTMSG command produced the error. Correct the TESTMSG command in the message command file, INMSG, and retry the program.

---

04462      ENDMSG missing or invalid on TESTMSG command.

**Explanation:** The ENDMSG value on the TESTMSG command is invalid or was not specified. An ENDMSG value is required. It must be a numeric value from 0 to 5 and be greater than or equal to the STARTMSG value.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which TESTMSG command produced the error. Correct the TESTMSG command in the message command file, INMSG, and retry the program.

---

04502      Missing destination on DEFINEALIAS command.

**Explanation:** You did not specify a destination on the DEFINEALIAS command. A destination consists of an ACCOUNT and USERID; SYSID, ACCOUNT, and USERID; or ALIAS and ALIASNAME.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which DEFINEALIAS command produced the error. Correct the DEFINEALIAS command in the message command file, INMSG, and retry the program.

---

04514      Invalid table type on DEFINEALIAS command.

**Explanation:** You specified an invalid value for an ALIAS parameter in the DEFINEALIAS command. The first character of the ALIAS parameter is the destination table type, and it must be blank, g, o, or p. The last three characters of the ALIAS parameter are the destination table ID, and they must be 1 to 3 alphanumeric characters. If the first character is blank, then the last three characters of the ALIAS must be blank.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which DEFINEALIAS command produced the error. Correct the DEFINEALIAS command in the message command file, INMSG, and retry the program.

---

04532      Invalid table name on DEFINEALIAS command.

**Explanation:** The ALIASTABLE parameter in the DEFINEALIAS command is missing, blank, or invalid. An ALIASTABLE name must be specified. The first character of the ALIASTABLE parameter is the destination table type, and it must be g, o, or p. The last three characters of the ALIASTABLE parameter are the destination table ID, and they must be 1 to 3 alphanumeric characters.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which DEFINEALIAS command produced the error. Correct the DEFINEALIAS command in the message command file, INMSG, and retry the program.

---

04562      FUNCTION invalid on DEFINEALIAS command.

**Explanation:** The value for the FUNCTION parameter in the DEFINEALIAS command is invalid. The FUNCTION parameter value must be a, c, d, e, or n.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which DEFINEALIAS command produced the error. Correct the DEFINEALIAS command in the message command file, INMSG, and retry the program.

---

04564      Invalid use of FUNCTION parameter on DEFINEALIAS command.

**Explanation:** Either entries were specified in the DEFINEALIAS command with 'E' for FUNCTION, or no entries were specified for one of the other FUNCTION values.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which DEFINEALIAS command produced the error. Correct the DEFINEALIAS command in the message command file, INMSG, and retry the program.

---

04566      AUTHORITY invalid on DEFINEALIAS command.

**Explanation:** Use of the AUTHORITY parameter in the DEFINEALIAS command is invalid or you specified an invalid value for the AUTHORITY parameter. You may use the AUTHORITY parameter only if you are defining a new alias table and the FUNCTION parameter is set to 'N'. Valid values for the AUTHORITY parameter are p, a, or g. If you are defining a private alias table, where the first character of the alias table name is P, or an organizational table, where the first character of the alias table name is O, then you cannot specify 'G' for AUTHORITY.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which DEFINEALIAS command produced the error. Correct the DEFINEALIAS command in the message command file, INMSG, and retry the program.

---

04598      Invalid parameter duplication on DEFINEALIAS command.

**Explanation:** An ALIASTABLE, FUNCTION, or AUTHORITY parameter is duplicated. These parameters can be specified only once in each DEFINEALIAS command.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which DEFINEALIAS command produced the error. Correct the DEFINEALIAS command in the message command file, INMSG, and retry the program.

---

04599      Incomplete destination on DEFINEALIAS command.

**Explanation:** A DEFINEALIAS command ended before a destination was completed, or a new type of destination started before the previous destination entry was completed. This is usually caused by a missing destination parameter in a DEFINEALIAS command. Destination entries are made up of an ACCOUNT and USERID; SYSID, ACCOUNT, and USERID; or an ALIAS and ALIASNAME. Therefore, you must specify each destination component next to its counterpart. For example, an ACCOUNT parameter should be entered next to a USERID parameter. If you specify a SYSID parameter, you must specify it next to its associated ACCOUNT and USERID parameters. An ALIAS parameter must have an associated ALIASNAME parameter. A SYSID next to an ALIASNAME is invalid.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which DEFINEALIAS command produced the error. Correct the DEFINEALIAS command in the message command file, INMSG, and retry the program.



---

04630 FILEID missing on PUTMEMBER command.

**Explanation:** You must specify a FILEID in the PUTMEMBER command.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which PUTMEMBER command produced the error. Correct the PUTMEMBER command in the message command file, INMSG, and retry the program.

---

04636 FORMAT invalid on PUTMEMBER command.

**Explanation:** You specified an invalid value for the FORMAT parameter in the PUTMEMBER command. FORMAT must be y or n. You cannot specify 'Y' for FORMAT with 'B' for DATATYPE or 'Y' for DELIMIT.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which PUTMEMBER command produced the error. Correct the PUTMEMBER command in the message command file, INMSG, and retry the program.

---

04650 ACK invalid on PUTMEMBER command.

**Explanation:** You specified an invalid value for the ACK parameter in the PUTMEMBER command. ACK must be blank or d.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which PUTMEMBER command produced the error. Correct the PUTMEMBER command in the message command file, INMSG, and retry the program.

---

04672 VERIFY invalid on PUTMEMBER command.

**Explanation:** You specified an invalid value for the VERIFY parameter in the PUTMEMBER command. VERIFY must be y or n.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which PUTMEMBER command produced the error. Correct the PUTMEMBER command in the message command file, INMSG, and retry the program.

---

04676 DATATYPE invalid on PUTMEMBER command.

**Explanation:** You specified an invalid value for the DATATYPE parameter in the PUTMEMBER command. DATATYPE must be a or b.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which PUTMEMBER command produced the error. Correct the PUTMEMBER command in the message command file, INMSG, and retry the program.

---

04678 DELIMIT invalid on PUTMEMBER command.

**Explanation:** You specified an invalid value for the DELIMIT parameter in the PUTMEMBER command. DELIMIT must be y or n.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which PUTMEMBER command produced the error. Correct the PUTMEMBER command in the message command file, INMSG, and retry the program.

---

---

04686 MEMBER missing on PUTMEMBER command.

**Explanation:** You must specify a MEMBER name on the PUTMEMBER command.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which PUTMEMBER command produced the error. Correct the PUTMEMBER command in the message command file, INMSG, and retry the program.

---

04688 LIBRARY missing on PUTMEMBER command.

**Explanation:** You must specify a LIBRARY name on the PUTMEMBER command.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which PUTMEMBER command produced the error. Correct the PUTMEMBER command in the message command file, INMSG, and retry the program.

---

04690 CRLFEOF invalid on PUTMEMBER command.

**Explanation:** You specified an invalid value for the CRLFEOF parameter in the SEND command. The CRLFEOF parameter must contain 6 hexadecimal characters.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which SEND command produced the error. Correct the SEND command in the message command file, INMSG, and retry the program.

---

04692 TRUNCATE invalid on PUTMEMBER command.

**Explanation:** You specified an invalid value for the TRUNCATE parameter in the PUTMEMBER command. The TRUNCATE parameter must be y or n.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which PUTMEMBER command produced the error. Correct the PUTMEMBER command in the message command file, INMSG, and retry the program.

---

04694 REPLACE invalid on PUTMEMBER command.

**Explanation:** You specified an invalid value for the REPLACE parameter in the PUTMEMBER command. REPLACE must be y or n.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which PUTMEMBER command produced the error. Correct the PUTMEMBER command in the message command file, INMSG, and retry the program.

---

04704 Multiple destinations on GETMEMBER command.

**Explanation:** You specified multiple destinations in a GETMEMBER command. You can specify only one destination.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which GETMEMBER command produced the error. Correct the GETMEMBER command in the message command file, INMSG, and retry the program.

---

04706      Incomplete destinations on GETMEMBER command.

**Explanation:** The destination must be ALIAS and ALIASNAME; ACCOUNT and USERID; SYSID, ACCOUNT and USERID; or LISTNAME.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which GETMEMBER command produced the error. Correct the GETMEMBER command in the message command file, INMSG, and retry the program.

---

04714      Invalid ALIAS on GETMEMBER command.

**Explanation:** You specified an invalid value for an ALIAS parameter in the GETMEMBER command. The first character of the ALIAS parameter is the destination table type, and it must be blank, g, o, or p. The last three characters of the ALIAS parameter are the destination table ID, and they must be 1 to 3 alphanumeric characters. If the first character is blank, the last three characters of the ALIAS parameter must also be blank.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which GETMEMBER command produced the error. Correct the GETMEMBER command in the message command file, INMSG, and retry the program.

---

04748      CHARGE invalid on GETMEMBER command.

**Explanation:** You specified an invalid value for the CHARGE parameter in the GETMEMBER command. CHARGE must be 1, 3, 5, or 6.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which GETMEMBER command produced the error. Correct the GETMEMBER command in the message command file, INMSG, and retry the program.

---

04750      ACK invalid on GETMEMBER command.

**Explanation:** You specified an invalid value for the ACK parameter in the GETMEMBER command. ACK must be blank, a, b, c, d, e, f, or r.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which GETMEMBER command produced the error. Correct the GETMEMBER command in the message command file, INMSG, and retry the program.

---

04774      RETAIN invalid on GETMEMBER command.

**Explanation:** You specified an invalid value for the RETAIN parameter in the GETMEMBER command. RETAIN must be a numeric value from 0 to 180.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which GETMEMBER command produced the error. Correct the GETMEMBER command in the message command file, INMSG, and retry the program.

---

04786      MEMBER name missing on GETMEMBER command.

**Explanation:** You must specify a MEMBER name on the GETMEMBER command.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which GETMEMBER command produced the error. Correct the GETMEMBER command in the message command file, INMSG, and retry the program.

---

04788      LIBRARY name missing on GETMEMBER command.

**Explanation:** You must specify a LIBRARY name on the GETMEMBER command.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which GETMEMBER command produced the error. Correct the GETMEMBER command in the message command file, INMSG, and retry the program.

---

04866      AUTHORITY invalid on LISTLIBRARIES command.

**Explanation:** You specified an invalid value for the AUTHORITY parameter in the LISTLIBRARIES command. AUTHORITY must be r or w.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which command produced the error. Correct the message command file, INMSG, and retry the command.

---

04868      SELECTION invalid on LISTLIBRARIES command.

**Explanation:** You specified an invalid value for the SELECTION parameter in the LISTLIBRARIES command. SELECTION must be a or c.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which LISTLIBRARIES command produced the error. Correct the message command file, INMSG, and try again.

---

04988      LIBRARY name missing on LISTMEMBERS command.

**Explanation:** You must specify a LIBRARY name on the LISTMEMBERS command.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which LISTMEMBERS command produced the error. Correct the LISTMEMBERS command in the message command file, INMSG, and retry the command.

## Profile command file syntax errors

The following describes the return codes for profile command file (INPRO) syntax errors.

---

05024      IEACCOUNT missing on IDENTIFY command.

**Explanation:** You must specify an Information Exchange account in your profile when Expedite Base/400 starts the session automatically.

**User Response:** Specify the account using the IEACCOUNT parameter in the IDENTIFY command in the profile command file, INPRO. Retry the program.

---

05026      IEUSERID missing on IDENTIFY command.

**Explanation:** You must specify an Information Exchange user ID in your profile when Expedite Base/400 starts the session automatically.

**User Response:** Specify the user ID using the IEUSERID parameter in the IDENTIFY command in the profile command file, INPRO. Retry the program.

---

05028      IEPASSWORD missing on IDENTIFY command.

**Explanation:** You must specify an Information Exchange password in your profile when Expedite Base/400 starts the session automatically.

**User Response:** Specify the password using the IEPASSWORD parameter in the IDENTIFY command in the profile command file, INPRO. Retry the program.

---

5030      Missing KEYRINGFILE on IDENTIFY command.

**Explanation:** The KEYRINGFILE was not specified. This is a required parameter. You must specify the name of the KEYRINGFILE.

**User Response:** When making an SSL connection, there must be a KDB file containing the certificate to allow the connection. This filename is specified in the KEYRINGFILE parameter. It is a required parameter when SSL is enabled. Correct the IDENTIFY command to specify the KDB filename containing the KEYRINGFILE parameter and retry the program

---

5031      IDENTIFY keyring file parameters not valid when AUTOSTRT is N.

**Explanation:** The keyring parameters cannot be specified on the IDENTIFY command when the AUTOSTART parameter is set to N on the TRANSMIT command.

**User Response:** Remove the KEYRINGFILE and KEYRINGPASSWORD or APPLICID parameters in the IDENTIFY command or set the AUTOSTART parameter to Y on the TRANSMIT command and retry the program.

---

5032      Missing KEYRINGPASSWORD on IDENTIFY command.

**Explanation:** The key database or key ring specified in the KEYRINGFILE parameter requires a password.

**User Response:** Provide either the correct KEYRINGPASSWORD parameter or KEYRINGSTASHFILE associated with the key database specified in the KEYRINGFILE parameter and retry the program

---

5033 Missing KEYRINGFILE or APPLICID on IDENTIFY command.

**Explanation:** You must specify either the KEYRINGFILE or the APPLICID parameter on the IDENTIFY command.

**User Response:** When making an SSL connection, there must be a KDB file containing the certificate, or an application id defined that is tied to the certificate, to allow the connection. This filename or application id is specified in the KEYRINGFILE or APPLICID parameter, respectively. One of those parameters are required when SSL is enabled.

Correct the IDENTIFY command in the message command file, INMSG, to specify the filename of the KEYRINGFILE parameter or the application id of the APPLICID parameter and retry the program.

---

5034 Missing KEYRINGPASSWORD on IDENTIFY command.

**Explanation:** The key database or key ring specified in the KEYRINGFILE parameter on the IDENTIFY command requires a password.

**User Response:** Provide either the correct KEYRINGPASSWORD parameter or KEYRINGSTASHFILE associated with the key database specified in the KEYRINGFILE parameter, on the IDENTIFY command and retry the program

---

5035 Both APPLICID and KEYRINGFILE on IDENTIFY command.

**Explanation:** You specified both the KEYRINGFILE and the APPLICID parameters on the IDENTIFY command.

**User Response:** When making an SSL connection, there must be a KDB file containing the certificate, or an application id defined that is tied to the certificate, to allow the connection. This filename or application id is specified in the KEYRINGFILE or APPLICID parameter, respectively. Only one of those parameters can be specified when SSL is enabled.

Correct the IDENTIFY command in the message command file, INMSG, to specify either the filename of the KEYRINGFILE parameter or the application id of the APPLICID parameter and retry the program.

---

5036 Both APPLICID and KEYRINGPASSWORD on IDENTIFY command.

**Explanation:** You specified both the APPLICID and the KEYRINGPASSWORD parameters on the IDENTIFY command.

**User Response:** When making an SSL connection, there must be a KDB file containing the certificate and a password for that file, or an application id defined that is tied to the certificate, to allow the connection. If you specified the KEYRINGFILE parameter, then you need the KEYRINGPASSWORD parameter, and if you specified the APPLICID parameter, then you do not need the KEYRINGPASSWORD.

Correct the IDENTIFY command in the message command file, INMSG, and retry the program.

---

05072      TIMEZONE invalid on IDENTIFY command.

**Explanation:** You specified an invalid value for the TIMEZONE parameter in the IDENTIFY command. TIMEZONE must be 1 to 5 alphanumeric characters. Valid values are ast, ahd, ahs, bst, cdt, cst, ead, edt, emt, est, gmt, jst, mdt, mst, pdt, pst, wed, wes, ydt, and yst. You can also specify the time zone as hours and minutes east or west of Greenwich Mean Time, for example, w0400 or e0400.

**User Response:** Correct the TIMEZONE parameter in the IDENTIFY command in the profile command file, INPRO. Retry the program.

---

05678      PROTOCOL invalid on TRACE command.

**Explanation:** You specified an invalid value for the PROTOCOL parameter in the TRACE command. PROTOCOL must be y or n.

**User Response:** Correct the PROTOCOL parameter in the TRACE command in the profile command file, INPRO. Retry the program.

---

05680      LINK invalid on TRACE command.

**Explanation:** You specified an invalid value for the LINK parameter in the TRACE command. LINK must be y or n.

**User Response:** Correct the LINK parameter in the TRACE command in the profile command file, INPRO. Retry the program.

---

05682      BASE invalid on TRACE command.

**Explanation:** You specified an invalid value for the BASE parameter in the TRACE command. BASE must be y or n.

**User Response:** Correct the BASE parameter in the TRACE command in the profile command file, INPRO. Retry the program.

---

05684      IOFILE invalid on TRACE command.

**Explanation:** You specified an invalid value for the IOFILE parameter in the TRACE command. IOFILE must be y or n.

**User Response:** Correct the IOFILE parameter in the TRACE command in the profile command file, INPRO. Retry the program.

---

05872      AUTOSTART invalid on TRANSMIT command.

**Explanation:** You specified an invalid value for the AUTOSTART parameter in the TRANSMIT command. AUTOSTART must be y or n.

**User Response:** Correct the AUTOSTART parameter in the TRANSMIT command in the profile command file, INPRO. Retry the program.

---

05876      AUTOEND invalid on TRANSMIT command.

**Explanation:** You specified an invalid value for the AUTOEND parameter in the TRANSMIT command. AUTOEND must be y or n.

**User Response:** Correct the AUTOEND parameter in the TRANSMIT command in the profile command file, INPRO. Retry the program.

---

05878 COMMITDATA invalid on TRANSMIT command.

**Explanation:** You specified an invalid value for the COMMITDATA parameter in the TRANSMIT command. COMMITDATA must be a numeric value from 1000 to 9999999 and must be an even multiple of, greater than, or equal to the MSGSIZE parameter value.

**User Response:** Correct the COMMITDATA parameter in the TRANSMIT command in the profile command file, INPRO. Retry the program.

---

05882 COMMTYPE invalid on TRANSMIT command.

**Explanation:** You specified an invalid value for the COMMTYPE parameter in the TRANSMIT command. COMMTYPE must be s or t.

**User Response:** Correct the COMMTYPE parameter in the TRANSMIT command in the profile command file, INPRO. Retry the program.

---

05884 MAXMSGs invalid on TRANSMIT command.

**Explanation:** You specified an invalid value for the MAXMSGs parameter in the TRANSMIT command. MAXMSGs must be a numeric value from 1 to 10.

**User Response:** Correct the MAXMSGs parameter in the TRANSMIT command in the profile command file, INPRO. Retry the program.

---

05892 RECOVERY invalid on TRANSMIT command.

**Explanation:** You specified an invalid value for the RECOVERY parameter in the TRANSMIT command. RECOVERY must be c, s, f, or u.

**User Response:** Correct the RECOVERY parameter of the TRANSMIT command in the profile command file, INPRO. Retry the program.

---

05894 Session-level recovery not valid with session in progress.

**Explanation:** You specified session-level recovery, but the session file indicates that a session is in progress.

**User Response:** Continue the present session using checkpoint-level recovery. If you want to use session-level recovery without completing the present session, either specify the RESET parameter on the IEBASE call command, or delete the session file, SESSION.

**Warning:** If you reset the session using the RESET parameter, you will no longer be able to continue the previous session. Failure to modify the message command file, INMSG, before resetting the session may result in some data being lost or duplicated.

---

05896 MSGSIZE invalid on TRANSMIT command.

**Explanation:** You specified an invalid value for the MSGSIZE parameter in the TRANSMIT command. MSGSIZE must be a numeric value from 1000 to 99999.

**User Response:** Correct the MSGSIZE parameter in the TRANSMIT command in the profile command file, INPRO. Retry the program.



---

05898      TIMEOUT invalid on TRANSMIT command.

**Explanation:** You specified an invalid value for the TIMEOUT parameter in the TRANSMIT command. TIMEOUT must be a number from 0 to 99.

**User Response:** Correct the TIMEOUT parameter in the TRANSMIT command in the profile command file, INPRO. Retry the program.

---

006010      USERLU missing on snacomm command.

**Explanation:** You must specify at least one USERLU value in the SNACOMM command.

**User Response:** Correct the USERLU parameter in the SNACOMM command in the profile command file, INPRO. This message also displays if you specified COMMTYPE (S) and used the TCPCOMM command instead of the SNACOMM command. Correct the error. Retry the program.

---

06012      Too many USERLU parameters on SNACOMM command.

**Explanation:** You can specify a maximum of 99 USERLU parameters in the SNACOMM command.

**User Response:** Correct the USERLU parameter in the SNACOMM command in the profile command file, INPRO. Retry the program.

---

06014      RUSIZE invalid on SNACOMM command.

**Explanation:** You specified an invalid value for the RUSIZE parameter in the SNACOMM command. RUSIZE must be a numeric value from 256 through 3840.

**User Response:** Correct the RUSIZE parameter in the SNACOMM command in the profile command file, INPRO. Retry the program.

---

06200      ENABLESSL invalid on SSL command.

**Explanation:** You specified an invalid value for the ENABLESSL parameter in the SSL command. Valid values are y or n.

**User Response:** Correct the ENABLESSL parameter in the SSL command in the profile command file, INPRO. Retry the program.

---

06201      CIPHERSUITES invalid on SSL command.

**Explanation:** You specified an invalid value for the CIPHERSUITES parameter in the SSL command. The value of CIPHERSUITES is a string consisting of 1 or more 2-character values. Valid characters are 0 through 9 or a through f.

**User Response:** Correct the CIPHERSUITES parameter in the SSL command in the profile command file, INPRO. Retry the program. Valid values for CIPHERSUITES are determined when your Cryptographic software (AC3) is installed on your AS/400 system and a default value is established. You should never change this value unless requested to do so by IBM Global Services personnel.

---

06202        SSLVERSION invalid on SSL command.

**Explanation:** You specified an invalid value for the SSLVERSION parameter in the SSL command. Valid values for SSLVERSION are 3031, 30 or 31.

**User Response:** Correct the SSLVERSION parameter in the SSL command in the profile command file, INPRO. Retry the program. 3031 is the default value for this parameter. You should never change this value unless requested to do so by IBM Global Services personnel.

---

06203        SSL only valid with TCP/IP communication types.

**Explanation:** You specified Y for the ENABLESSL parameter, but the communication type is not TCP/IP. SSL is only valid when communicating via TCP/IP.

**User Response:** Set your communication type to TCP/IP (COMMTYPE(T)) on the TRANSMIT command in the profile command file, INPRO, and retry the program again.

## Parser errors

This section describes the return codes for parser errors.

---

14000 Null character in input file.

**Explanation:** There is a null character in either the profile command file, INPRO, or the message command file, INMSG. Null characters are not permitted in INMSG or INPRO.

**User Response:** Check the message response file, OUTMSG, the profile response file, OUTPRO, or the response work file, OUTWORK, to determine which command produced the error. Correct the command file and retry the program.

---

14020 Command or parameter name too long.

**Explanation:** A command or parameter name in the command file is invalid because it is too long.

**User Response:** Check the message response file, OUTMSG, profile response file, OUTPRO, or response work file, OUTWORK, to determine which command produced the error. Correct the command file and retry the program.

---

14030 Parameter value too long.

**Explanation:** A parameter value in the command file is invalid because it is longer than the maximum length permitted for the parameter. This is sometimes caused by unbalanced parentheses.

**User Response:** Check the message response file, OUTMSG, profile response file, OUTPRO, or response work file, OUTWORK, to determine which command produced the error. Correct the appropriate command file and retry the program.

---

14040 Command expected but not found.

**Explanation:** You did not specify an expected command in the command file. It is possible that you specified a parameter before a command.

**User Response:** Check the message response file, OUTMSG, profile response file, OUTPRO, or response work file, OUTWORK, to determine which command produced the error. Correct the appropriate command file and retry the program.

---

14050 Parameter and value or semicolon expected but not found.

**Explanation:** There is a syntax error in the profile command file, INPRO, or message command file, INMSG. Either a semicolon or a parameter and value was expected but was not found. This is usually caused by omitting the semicolon from a command, omitting the value of a parameter, or leaving space between the parameter name and value.

**User Response:** Check the message response file, OUTMSG, profile response file, OUTPRO, or response work file, OUTWORK, to determine which command produced the error. Correct the appropriate command file and retry the program.

---

15010 Parameter value too long.

**Explanation:** The parameter value for one of the commands in the command file is longer than the maximum allowed for that parameter.

**User Response:** Check the message response file, OUTMSG, profile response file, OUTPRO, or response work file, OUTWORK, to determine which command produced the error. Correct the appropriate command file and retry the program.

---

15020 Duplicate parameter found.

**Explanation:** The same parameter was specified more than once in a command.

**User Response:** Check the message response file, OUTMSG, profile response file, OUTPRO, or response work file, OUTWORK, to determine which command produced the error. Correct the appropriate command file and retry the program.

---

15030 Invalid parameter found.

**Explanation:** You specified an invalid parameter in the command file.

**User Response:** Check the message response file, OUTMSG, profile response file, OUTPRO, or response work file, OUTWORK, to determine which command produced the error. Correct the appropriate command file and retry the program.

---

15040 Command not recognized.

**Explanation:** You specified an unrecognized command in the command file.

**User Response:** Check the message response file, OUTMSG, profile response file, OUTPRO, or response work file, OUTWORK, to determine which command produced the error. Correct the appropriate command file and retry the program.

---

15042 COMMIT command only valid with user-initiated recovery.

**Explanation:** You specified a COMMIT command with the RECOVERY parameter on the TRANSMIT command specified as C, S, or F. COMMIT commands are processed with 'U' for RECOVERY only.

**User Response:** Update the RECOVERY parameter on the TRANSMIT command to 'U', or remove all COMMIT commands from the message command file, INMSG. Retry the program.

**Warning:** If you reset the session using the RESET parameter on the IEBASE call command, you will no longer be able to continue the session. If it is necessary to reset the session, modify the previous message command file, INMSG, deleting any commands that have already been processed. Failure to do this may result in some data being lost or duplicated.

## Destination verification errors

This section describes the return codes for errors that occur verifying the Information Exchange destination.

---

16020      The destination specified is not a valid IE destination.

**Explanation:** The destination specified in the SEND, SENDSTREAM, or SENDEDI command does not exist. The data was not sent.

**User Response:** Correct the destination in the SEND or SENDSTREAM command, EDI data, EDI destination table, or EDI qualifier table. Retry the command.

---

16030      IE destination is blocked by trading partner list or payment levels.

**Explanation:** The destination in the SEND, SENDSTREAM, or SENDEDI command exists. However, the message cannot be sent because it is blocked by the payment level specified or by the trading partner list. The data was not sent.

**User Response:** Make sure the Information Exchange destination and payment levels are correct. Check with your service administrator to change a trading partner list or payment levels. Retry the command.

---

16040      IE was unable to verify the destination immediately.

**Explanation:** The destination in the SEND, SENDSTREAM, or SENDEDI command could not be verified immediately because it is on another Information Exchange system. The data was not sent.

**User Response:** Retry the command using an f, g, or n value for the VERIFY parameter. Information Exchange cannot immediately verify a destination on another system.

---

16050      No update access to library.

**Explanation:** You specified 'Y' for VERIFY on the PUTMEMBER command and either the library does not exist or you do not have update access to the library. The data was not sent.

**User Response:** Use Information Exchange Administration Services to verify that the library you are trying to update exists and that you have update authority for it. Retry the command.

---

16052      File with specified message key does not exist.

**Explanation:** The message key you specified does not match any of the files in your mailbox.

**User Response:** Verify that you specified the correct message key. The message key is shown on the AVAILABLE record in response to a QUERY command. Correct the message key and retry the command.

---

16054      File to purge is being received.

**Explanation:** The message key you specified is for a file that is in the process of being received, and cannot be purged.

**User Response:** If you do not want to receive the file, then discontinue the receive process and reset the Information Exchange session. Retry the command.

---

16056 Information Exchange profile does not allow files to be purged.

**Explanation:** Your Information Exchange profile does not allow purging of files from your mailbox.

**User Response:** If you want to be able to purge files from your mailbox, use Information Exchange Administration Services or ask your System Administrator to change your profile and allow this action. Retry the command.

---

16060 Unable to retrieve library member.

**Explanation:** Information Exchange could not retrieve the library member because either the library does not exist, you do not have read access to the library, or the destination you specified is invalid.

**User Response:** Receive the system error message from your mailbox. The system error message explains why the GETMEMBER failed. Correct the problem and retry the command.

## EDI errors

This section describes the return codes for EDI errors.

---

17102 Invalid X12 header in file xxxxxxxx.

**Explanation:** The X12 header in the data you attempted to send is invalid. This envelope and the envelopes following it in the file were not sent.

**User Response:** Correct the X12 ISA header that caused the error in the envelope, and retry the command. If there were multiple envelopes in this file, check the message response file, OUTMSG, for SENT records to determine which envelopes in the file were sent.

---

17106 Missing X12 destination in file xxxxxxxx.

**Explanation:** The X12 header in the data you attempted to send does not contain an X12 receiver ID. This envelope and the envelopes following it in the file were not sent.

**User Response:** Correct the X12 ISA header that caused the error in the envelope, and retry the command. If there were multiple envelopes in this file, check the message response file, OUTMSG, for SENT records to determine which envelopes in the file were sent.

---

17108 Invalid X12 destination in file xxxxxxxx.

**Explanation:** The X12 header in the data you attempted to send contains an X12 receiver ID that could not be resolved. This envelope and the envelopes following it in the file were not sent.

**User Response:** Correct the EDI qualifier table, the EDI destination table, or the X12 ISA header that caused the error in the envelope. Retry the command. If there were multiple envelopes in this file, check the message response file, OUTMSG, for SENT records to determine which envelopes in the file were sent.

---

17110 Invalid X12 binary or encrypted segment in file xxxxxxxx.

**Explanation:** The X12 data contains an invalid binary or security segment. This envelope and the envelopes following it in the file were not sent.

**User Response:** Correct the X12 data and retry the command. If there were multiple envelopes in this file, check the message response file, OUTMSG, for SENT records to determine which envelopes in the file were sent.

---

17112 Invalid X12 binary or encrypted length in file xxxxxxxx.

**Explanation:** The length element in an X12 binary or security segment is invalid. This envelope and the envelopes following it in the file were not sent.

**User Response:** Correct the X12 data and retry the command. If there were multiple envelopes in this file, check the message response file, OUTMSG, for SENT records to determine which envelopes in the file were sent.

---

17114 X12 control number is missing or invalid.

**Explanation:** The X12 header in the data you attempted to send does not contain an interchange control number, or the interchange control number is not numeric. This envelope and the envelopes following it in the file were not sent.

**User Response:** Correct the X12 ISA header that caused the error in the envelope and retry the command. If there were multiple envelopes in this file, check the message response file, OUTMSG, for SENT records to determine which envelopes in the file were sent.

---

17116 X12 control number error.

**Explanation:** The X12 control number in the IEA is missing or does not match the control number in the ISA. This envelope and the envelopes following it in the file were not sent.

**User Response:** Correct the X12 data that caused the error and retry the command. If there were multiple envelopes in this file, check the message response file, OUTMSG, for SENT records to determine which envelopes in the file were sent.

---

17118 Missing IEA in X12 data.

**Explanation:** The X12 IEA segment was not found in the data. This envelope and the envelopes following it in the file were not sent.

**User Response:** Correct the X12 data that caused the error and retry the command. If there were multiple envelopes in this file, check the message response file, OUTMSG, for SENT records to determine which envelopes in the file were sent.

---

17202 Invalid UCS header in file xxxxxxxx.

**Explanation:** The UCS BG header in the data you attempted to send is invalid. The segment terminator must be hex 15. This envelope and the envelopes following it in the file were not sent.

**User Response:** Correct the UCS BG header that caused the error and retry the command. If there were multiple envelopes in this file, check the message response file, OUTMSG, for SENT records to determine which envelopes in the file were sent.

---

17206 Missing UCS destination in file xxxxxxxx.

**Explanation:** The UCS header in the data you attempted to send does not contain a UCS receiver ID. This envelope and the envelopes following it in the file were not sent.

**User Response:** Correct the UCS BG header that caused the error and retry the command. If there were multiple envelopes in this file, check the message response file, OUTMSG, for SENT records to determine which envelopes in the file were sent.

---

17208 Invalid UCS destination in file xxxxxxxx.

**Explanation:** The UCS BG header in the data you attempted to send contains a UCS receiver ID that could not be resolved. This envelope and the envelopes following it in the file were not sent.

**User Response:** Correct the EDI qualifier table, the EDI destination table, or the UCS BG header that caused the error, and retry the command. If there were multiple envelopes in this file, check the message response file, OUTMSG, for SENT records to determine which envelopes in the file were sent.



---

17302 Invalid EDIFACT header in file xxxxxxxx.

**Explanation:** The EDIFACT UNB header in the data you attempted to send is invalid. The segment terminator was not found. This envelope and the envelopes following it in the file were not sent.

**User Response:** Correct the EDIFACT UNB header in the envelope that caused the error and retry the command. If there were multiple envelopes in this file, check the message response file, OUTMSG, for SENT records to determine which envelopes in the file were sent.

---

17306 Missing EDIFACT destination in file xxxxxxxx.

**Explanation:** The EDIFACT UNB header in the data you attempted to send does not contain an EDIFACT receiver ID. This envelope and the envelopes following it in the file were not sent.

**User Response:** Correct the EDIFACT UNB header that caused the error, and retry the command. If there were multiple envelopes in this file, check the message response file, OUTMSG, for SENT records to determine which envelopes in the file were sent.

---

17308 Invalid EDIFACT destination in file xxxxxxxx.

**Explanation:** The EDIFACT UNB header in the data you attempted to send contains an EDIFACT receiver ID that could not be resolved. This envelope and the envelopes following it in the file were not sent.

**User Response:** Correct the EDI qualifier table, the EDI destination table, or the EDIFACT UNB header in the envelope that caused the error. Retry the command. If there were multiple envelopes in this file, check the message response file, OUTMSG, for SENT records to determine which envelopes in the file were sent.

---

17316 EDIFACT control number error.

**Explanation:** The EDIFACT control number in the UNZ is missing or does not match the control number in the UNB. This envelope and the envelopes following it in the file were not sent.

**User Response:** Correct the EDIFACT data in the envelope that caused the error and retry the command. If there were multiple envelopes in this file, check the response file for SENT records to determine which envelopes in the file were sent.

---

17318 Missing UNZ in EDIFACT data.

**Explanation:** A UNA or UNB was found before the preceding EDIFACT envelope ended. This envelope and envelopes following it in the file were not sent.

**User Response:** Correct the EDIFACT data that caused the error and retry the command. If there were multiple envelopes in this file, check the response file for SENT records to determine which envelopes in the file were sent.

---

17402 Invalid UN/TDI header in file xxxxxxxx.

**Explanation:** The UN/TDI STX header in the data you attempted to send is invalid. The segment terminator was not found. This envelope and the envelopes following it in the file were not sent.

**User Response:** Correct the UN/TDI STX header that caused the error, and retry the command. If there were multiple envelopes in this file, check the message response file, OUTMSG, for SENT records to determine which envelopes in the file were sent.

---

17406 Missing UN/TDI destination in file xxxxxxxx.

**Explanation:** The UN/TDI STX header in the data you attempted to send does not contain a UN/TDI receiver ID. This envelope and the envelopes following it in the file were not sent.

**User Response:** Correct the UN/TDI STX header that caused the error, and retry the command. If there were multiple envelopes in this file, check the message response file, OUTMSG, for SENT records to determine which envelopes in the file were sent.

---

17408 Invalid UN/TDI destination in file xxxxxxxx.

**Explanation:** The UN/TDI STX header in the data you attempted to send contains a UN/TDI receiver ID that could not be resolved. This envelope and the envelopes following it in the file were not sent.

**User Response:** Correct the EDI qualifier table, the EDI destination table, or the UN/TDI STX header in the envelope that caused the error. Retry the command. If there were multiple envelopes in this file, check the message response file, OUTMSG, for SENT records to determine which envelopes in the file were sent.

---

18010 Null character found in EDI table "xxxxxxx".

**Explanation:** You specified a null character in the EDI destination table or the EDI qualifier table specified.

**User Response:** Correct the table and retry the command.

---

18020 EDI table xxxxxxxx has an invalid parameter "zzzzzzzz".

**Explanation:** The parameter name in the EDI destination table or EDI qualifier table is not a valid parameter name or does not have an associated value.

**User Response:** Correct the table and retry the command.

---

18030 EDI table xxxxxxxx has an invalid parameter value "zzzzzzzz".

**Explanation:** The parameter value in the EDI destination table or EDI qualifier table is longer than the maximum length allowed.

**User Response:** Correct the table and retry the command.

---

18040 EDI table xxxxxxxx contains a duplicate parameter "zzzzzzzz".

**Explanation:** You specified the same parameter more than once for an entry in the EDI destination table or EDI qualifier table. This may be caused by a missing semicolon between entries.

**User Response:** Correct the table and retry the program.

---

18110 End of file found before end of EDI envelope in file "xxxxxxx".

**Explanation:** Expedite Base/400 encountered the end of the file before the end of the EDI envelope. This envelope was not sent.

---

**User Response:** Correct the EDI data and retry the command. If there were multiple envelopes in this file, check the message response file, OUTMSG, for SENT records to determine which envelopes in the file were sent.

---

18120      Unable to determine EDI type in file xxxxxxxx.

**Explanation:** Expedite Base/400 could not determine the type of EDI data you attempted to send. This envelope and the envelopes following it in the file were not sent.

**User Response:** Correct the EDI data and retry the command. If there were multiple envelopes in this file, check the message response file, OUTMSG, for SENT records to determine which envelopes in the file were sent.

---

18130      Error processing EDI data in file xxxxxxxx.

**Explanation:** There was an error processing the EDI data you attempted to send. This envelope and the envelopes following it in the file were not sent.

**User Response:** Correct the EDI data and retry the command. If there were multiple envelopes in this file, check the message response file, OUTMSG, for SENT records to determine which envelopes in the file were sent.

---

18210      Qualifier table xxxxxxxx contains an invalid alias "aaaa".

**Explanation:** You specified an invalid value for the default alias in the qualifier table. The first character of the ALIAS is the destination table type and must be blank, g, o, or p. The last three characters of the ALIAS are the destination table ID and must be one to three alphanumeric characters.

**User Response:** Correct the EDI destination table and retry the program.

---

18230      Invalid IE destination for entry "aaaaaaaaaaaaaa".

**Explanation:** You specified an invalid Information Exchange destination for the EDI destination in the EDI destination table. This indicates that the destination is missing, incomplete, contains more than one destination type, or uses an invalid alias type.

**User Response:** Correct the EDI qualifier table and retry the program.

---

18250      End of file found before end of destination in file xxxxxxxx.

**Explanation:** Expedite Base/400 encountered an end-of-file in the middle of either a qualifier table entry or EDI destination table entry. Make sure the last table entry ends with a semicolon.

**User Response:** Correct the EDI qualifier table and retry the program.

---

18300      Invalid EDI data received.

**Explanation:** The data received with the RECEIVEEDI command was not valid EDI data. Once Expedite Base/400 determines that the data was not valid EDI data, the rest of the data in the message is not reformatted. Therefore the records may not be separated as you want them to be.

**User Response:** No response is needed. This is only an informational message.

---

19006      Unable to create socket. %s.

**Explanation:** Expedite Base was unable to create a socket for communication with Information Exchange.

**User Response:** Verify that TCP/IP has been configured properly on your system. Retry the program.

---

19007      Unable to connect to Information Exchange.

**Explanation:** Expedite Base was unable to connect with Information Exchange.

**User Response:** Information Exchange or the Domain Name Server may be temporarily unavailable. Verify that IETCPHOSTn and IETCPPORTn are valid. Retry the program.

---

19009      Unable to resolve host name.

**Explanation:** Expedite Base was unable to resolve the host name.

**User Response:** Expedite Base tried to resolve the host name specified in IETCPHOSTn but was not successful. The host may be temporarily unavailable. Retry the program. If the problem persists, contact the Help Desk.

---

19011      Unable to connect to host.

**Explanation:** Expedite Base was unable to connect to the Information Exchange Common Front End using TCP/IP. If you have never successfully connected to the Information Exchange Common Front End using TCP/IP, your connection may not be set up properly.

**User Response:** If you have connected before, verify that IETCPHOSTn and IETCPPORTn are valid. The host may be temporarily unavailable. Retry the program. If the problem persists, contact the Help Desk.

---

19012      TCP/IP subsystem is not running.

**Explanation:** TCP/IP subsystem on your system is not running.

**User Response:** Verify that TCP/IP has been installed, configured and started properly. Retry the program. If the problem persists, contact the Help Desk.

---

19014      Invalid Information Exchange account and/or user ID specified.

**Explanation:** You specified an account and user ID that is not valid.

**User Response:** Enter the correct account and user ID and run the session again.

---

19015      Invalid Information Exchange password specified.

**Explanation:** You specified a password that is not valid.

**User Response:** Enter the correct password and run the session again.

---

19016      Unable to receive from host.

**Explanation:** Expedite Base was unable to receive data from Information Exchange.

---

---

**User Response:** Verify that the link has not been dropped, your Information Exchange ID is not being used by someone else, and the inactivity timeout has not been reached. Retry the program. If the problem persists, contact the Help Desk.

---

19017      Unable to send to host.

**Explanation:** Expedite Base was unable to send data to Information Exchange.

**User Response:** Verify that the link has not been dropped, your Information Exchange ID is not being used by someone else, and the inactivity timeout has not been reached. Retry the program. If the problem persists, contact the Help Desk.

---

19019      IETCPHOSTn and IETCPPORTn missing on TCPCOMM command.

**Explanation:** You must specify both an IETCPHOST and an IETCPPORTn parameter on the TCPCOMM command.

**User Response:** Verify that you have a valid TCPCOMM command with at least one IETCPHOSTn and one IETCPPORTn parameter specified in the profile command file, INPRO. If you do not know the correct values for these parameters, contact the Help Desk. Retry the program.

---

19020      IETCPHOSTn or IETCPPORTn missing on TCPCOMM command.

**Explanation:** You must specify both an IETCPHOST and an IETCPPORTn parameter on the TCPCOMM command. Only one was specified.

**User Response:** Verify that you have a valid TCPCOMM command with at least one IETCPHOSTn and one IETCPPORTn parameter specified in the profile command file, INPRO. If you do not know the correct values for these parameters, contact the Help Desk. Retry the program.

---

19021      Invalid IP address.

**Explanation:** You specified an invalid IP address for the IETCPHOSTn parameter in the TCPCOMM command. A valid IP address uses the format nnn.nnn.nnn.nnn, where each *nnn* is a number from 1 to 255.

**User Response:** Correct the IP address specified in the IETCPHOSTn parameter on the TCPCOMM command in the profile command file, INPRO. If you do not know the correct IP address number for Information Exchange, contact the Help Desk. Retry the program.

---

19022      Invalid IP port number.

**Explanation:** You specified an invalid IP port number for the IETCPPORTn parameter in the TCPCOMM command. A valid IP port number must be a number from 1 to 65535.

**User Response:** Correct the IP port number specified in the IETCPPORTn parameter specified in the profile command file, INPRO. If you do not know the correct IP port number for Information Exchange, contact the Help Desk. Retry the program.

---

19023      Socket has been lost.

**Explanation:** Expedite Base determined that the socket connection has been lost.

**User Response:** Expedite Base received an error from Information Exchange that the socket connection has been lost. Retry the program. If the problem persists, contact the Help Desk.

---

19024 TCP/IP activity timeout.

**Explanation:** Expedite Base determined there was a TCP/IP activity timeout.

**User Response:** Retry the program. If the problem persists, contact the Help Desk.

---

19025 The Information Exchange session is out of sync.

**Explanation:** Expedite Base received a response that the Information Exchange session is out of sync.

**User Response:** Retry the program. If the problem persists, contact the Help Desk.

---

19026 An active session was already started for user.

**Explanation:** Expedite Base received a response that an active session was already started for the user.

**User Response:** Retry the program. If the problem persists, contact the Help Desk.

---

19027 Communication with the Information Exchange AOR has been lost.

**Explanation:** Expedite Base received a response that communication with the Information Exchange AOR has been lost.

**User Response:** The host may be temporarily unavailable. Retry the program. If the problem persists, contact the Help Desk.

---

19028 Fatal system error.

**Explanation:** Expedite Base received a response that there was a fatal system error with Information Exchange.

**User Response:** Retry the program. If the problem persists, contact the Help Desk.

---

19029 All hosts are too busy to accept new traffic.

**Explanation:** Expedite Base received a response that all hosts are too busy to accept new traffic.

**User Response:** Retry the program. If the problem persists, contact the Help Desk.

---

19030 Data from Information Exchange is bigger than TCP\_start blocksize.

**Explanation:** Expedite Base has determined that the data being sent from Information Exchange is bigger than the TCP\_start blocksize.

**User Response:** Retry the program. If the problem persists, contact the Help Desk.

---

19031 Invalid data received.

**Explanation:** Expedite Base received invalid data during session initialization.

**User Response:** Check to make sure that you have used the correct IP address for a non-SSL connection and retry the program. If you wanted an SSL connection ensure that you are using a correct IP address for the SFE. If the problem persists, contact the Help Desk. You will be asked to FAX for send the trace files, LINKTRC and BASETRC from the failed session, to the Help Desk for problem determination.

## File validation errors

This section describes the return codes for file validation errors.

---

20022 Invalid record length for INPRO file.

**Explanation:** The record length for the profile command file, INPRO, must be less than or equal to 255.

**User Response:** Correct your profile command file and try the program again.

---

20030 Invalid record length for OUTPRO file.

**Explanation:** The record length for the profile response file, OUTPRO, must be less than or equal to 255, greater than or equal to the record length of the profile command file, INPRO, and greater than or equal to 72.

**User Response:** Correct your profile response file, OUTPRO, and try the program again.

---

20038 Invalid record format for OUTPRO file.

**Explanation:** The record format for the profile response file, OUTPRO, indicates that the file contains ISO/ANSI device control characters or machine control characters; for example, FBA or FBM. This record format is not supported for the profile response file.

**User Response:** Correct your profile response file, OUTPRO, and try the program again.

---

20040 Invalid record length for OUTMSG file.

**Explanation:** The record length for the message response file, OUTMSG, must be less than or equal to 255 and greater than or equal to the record length of the message command file, INMSG.

**User Response:** Correct your message response file, OUTMSG, and try the program again.

---

20048 Invalid record format for OUTMSG file.

**Explanation:** The record format for the message response file, OUTMSG, indicates that the file contains ISO/ANSI device control characters or machine code control characters; for example, FBA or FBM. This record format is not supported for the message response file.

**User Response:** Correct your message response file, OUTMSG, and try the program again.

---

20052 Invalid record length for INMSG.

**Explanation:** The record length for the message command file, INMSG, must be less than or equal to 255.

**User Response:** Correct your message command file, OUTMSG, and try the program again.



## General environment errors

This section describes the return codes for general environment errors.

---

20365      Unable to allocate storage - location ID *ZZZZZZZZZZZZZZZZ*.

**Explanation:** Expedite Base/400 was unable to get the memory it needed. The location ID indicates the location within the program where the memory allocation failed. This ID can be used by the Help Desk if the memory allocation failure is caused by an internal error.

**User Response:** Make sure there is adequate memory and retry the program. If the problem persists, call the Help Desk.

---

20410      Profile not found.

**Explanation:** Expedite Base/400 could not find the profile information. You must have a profile command file, INPRO.

**User Response:** Create the profile command file, INPRO, containing all of the required information. Then retry the program again.

---

20611      Error opening input file.

**Explanation:** Expedite Base/400 could not open the message command file, INMSG. Display the OUTMSG file used. There might be a REASON parameter that gives more information about why the file could not be opened.

**User Response:** If there is a REASON parameter in the OUTMSG file, check it for more information about the error. Check the INMSG file for errors and try the program again.

---

20620      Unable to restart at checkpoint due to error in INMSG.

**Explanation:** Commands in the message command file, INMSG, that were processed before the last checkpoint have been changed. Expedite Base/400 is unable to continue the current session at the latest checkpoint. Commands in INMSG that have already been processed, echoed to OUTMSG, should not be changed if you want to restart the session at the last checkpoint.

**User Response:** Reset the session using the RESET command line parameter on the IEBASE call command. Before starting the next session, review the message response file, OUTMSG, to see which commands were processed successfully. Remove these commands from the message command file, INMSG, so they are not processed again.

**Warning:** If you reset the session using the RESET command line parameter you will no longer be able to continue the previous session. Failure to modify the message command file, INMSG, before resetting the session may result in some data being lost or duplicated.

---

21000      Expedite Base/400 encountered an unexpected condition.

**Explanation:** Expedite Base/400 encountered an unexpected condition during execution.

**User Response:** Retry the program. If the problem persists, contact the Help Desk. You will be asked to FAX or send the trace file BASETRC from the failed session to the Help Desk for problem determination.

---

21013 Error reading EDI table "xxxxxxx".

**Explanation:** Expedite Base/400 could not read the EDI qualifier table file or EDI destination table file.

**User Response:** Make sure there are no input or output problems with the file. Retry the program.

---

21606 Receiver not found in lookup table.

**Explanation:** You specified 'T' for the COMPRESS parameter, which indicates you want the lookup table to be consulted before your data is sent compressed, but the receiver was not in the lookup table; therefore, the data was not compressed before sending. The COMPRESS parameter in your OUTMSG will be set to 'W' for warning.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine which receiver was not found in the table. Either specify 'Y' for COMPRESS, or add the receiver to your lookup table, and retry the program.

---

21810 File operation zzzzzzz failed on file xxxxxxx.

**Explanation:** Expedite Base/400 could not access a file in the requested mode.

**User Response:** Check your file and make sure there are no I/O problems. Check the file permissions to make sure Expedite Base/400 has access to the file. Retry the program. If the problem persists, contact the Help Desk. You will be asked to FAX or send the trace file BASETRC to the Help Desk for problem determination.

---

22240 Error looking up message.

**Explanation:** An error occurred while looking up an error message. This is caused by an error in the error message description file, ERRORMSG, or the extended error text file, ERRORTXT.

**User Response:** If you modified either the error message description file, ERRORMSG, or the extended error text file, ERRORTXT, retry the program with the original files. If the problem continues, contact the Help Desk.

---

22411 Error opening receive file.

**Explanation:** Expedite Base/400 could not open the file for the data being received. There might be a REASON parameter that gives more information about why the file could not be opened.

**User Response:** If there is a REASON parameter, check it for more information about the error. Make sure that you specified a valid FILEID for the RECEIVE or RECEIVEEDI command. Also, check the receive file and the RCVWORK file for errors. Try the program again.

---

22413 Invalid record length in the receive work file, RCVWORK.

**Explanation:** The record length of the RCVWORK file is not the same as it was in the session that is being restarted. The work files should not be modified while a session is in progress.

**User Response:** Make sure the RCVWORK file refers to the same data set that it did in the session that is being restarted and that the file has not been modified. If the problem continues, reset the session using the RESET parameter on the IEBASE call command.



**NOTE:** If you reset the session using the RESET parameter, you can no longer continue the previous session, and some data might be lost.

---

22420 Record length mismatch in receive file.

**Explanation:** The records in the data received, as indicated by CRLF characters or 2-byte length indicators, were too long for the receiving file, and you specified 'E' for RESRECL in the RECEIVE command.

**User Response:** Receive the data into a file with a larger record length. If you want Expedite Base/400 to split the records that are too long, specify 'S' for RESRECL in the RECEIVE command.

---

22440 Length delimiters in data invalid.

**Explanation:** The common data header or DELIMITED parameter indicated that the data contained two-byte length delimiters to separate records, but the lengths indicated by the delimiters did not match the length of the data. The length delimiters were processed, but the records may not be separated as you want them to be.

**User Response:** No response is needed. This is only an information message.

---

22460 Some of the records in the received file were segmented.

**Explanation:** Some of the records delimited by two-byte length indicators or by CRLF characters in the received file were segmented into multiple records because they were too long.

**User Response:** No response is necessary. This is an information-only message.

---

22484 Invalid record length on receive.

**Explanation:** The record length of the receive file is not the same as it was in the session that is being restarted. The file for the receive in progress should not be modified while a session is in progress.

**User Response:** Make sure that the file ID used when receiving refers to the same file that it did in the session being restarted and that the file has not been modified. If the problem continues, reset the session using the RESET parameter on the IEBASE call command.



**NOTE:** If you reset the session using the RESET parameter, you can no longer continue the previous session, and some data might be lost.

---

22610 Send or Putmember file not found.

**Explanation:** Expedite could not open the file indicated in the SEND or PUTMEMBER command.

**User Response:** Check the FILEID parameter on the SEND or PUTMEMBER command and retry the command.

---

22615      Send or Putmember file was empty.

**Explanation:** The file indicated in the SEND or PUTMEMBER command was empty.

**User Response:** Correct the message command file, INMSG, or send or putmember file and retry the command.

---

23410      EDI send file xxxxxxxx not found.

**Explanation:** Expedite Base/400 could not open the file indicated in the SENDEDI command. There might be a REASON parameter that gives more information about why the file could not be opened.

**User Response:** If there is a REASON parameter, check it for more information about the error. Also, make sure the file specified is not empty. Check the FILEID parameter for errors and try the command again.

---

23415      EDI send file was empty.

**Explanation:** The file indicated in the SENDEDI command did not contain any EDI data. It was empty or contained only blanks.

**User Response:** Correct the message command file, INMSG, or EDI send file and retry the command.

---

23500      No libraries found to list.

**Explanation:** There were no libraries found for the parameters specified on the LISTLIBRARIES command.

**User Response:** Check the parameters specified in the LISTLIBRARIES command. If the AUTHORITY, SELECTION, or OWNER parameters are incorrect, correct them and retry the command.

---

23502      Owing account for libraries invalid.

**Explanation:** The owing account ID specified by the OWNER parameter of the LISTLIBRARIES command is not recognized by Information Exchange.

**User Response:** Check that the owing account ID specified in the LISTLIBRARIES command is correct. If not, correct the OWNER parameter and retry the command.

---

23504      Library does not contain any members.

**Explanation:** The library specified in the LISTMEMBERS command does not contain any members. The command was not processed.

**User Response:** Check the library specified in the LISTMEMBERS command. Correct the LIBRARY parameter and try again.

---

23506      Library does not exist.

**Explanation:** The library specified in the LISTMEMBERS command does not exist on Information Exchange.

**User Response:** Check the library specified in the LISTMEMBERS command. If it is correct, use Information Exchange Administration Services to verify that the library exists.

---

23508      Library owning account invalid.

**Explanation:** The library owning account specified in the OWNER parameter of the LISTMEMBERS command is not recognized by Information Exchange.

**User Response:** Check the library owning account specified in the LISTMEMBERS command. Correct the OWNER parameter and retry the command.

---

23510      Read access not permitted for library.

**Explanation:** The ACCOUNT or USERID does not have read access to the library specified in the LISTMEMBERS command.

**User Response:** Use Information Exchange Administration Services to verify that you have read access to the library. Retry the command.

## Session start and end errors

This section describes the return codes for session start and end errors.

---

24000      Error in restart processing.

**Explanation:** Expedite was not able to restart the session with Information Exchange. The session file, SESSION, may be damaged.

**User Response:** Reset the session using the RESET command line parameter on the IEBASE CALL command. Also, make sure there is not another user using this user ID. If the problem persists, contact the Help Desk. Before starting the next session, review the message response file, OUTMSG, to see which commands were processed successfully. Remove these commands from the message command file, INMSG, so they are not processed again.

**Warning:** If you reset the session using the RESET parameter you will no longer be able to continue the previous session. Failure to modify the message command file, INMSG, before resetting the session may result in some data being lost or duplicated.

---

24020      Restart and original recovery levels are not equal.

**Explanation:** The restart level differs from the original recovery level. The session has not started. Either your session file, SESSION, is damaged or another user is using this user ID.

**User Response:** Reset the session using the RESET command line parameter on the IEBASE command. Also, make sure there is not another user using this user ID. If the problem persists, contact the Help Desk. Before starting the next session, review the message response file, OUTMSG, to see which commands were processed successfully. Remove these commands from the message command file, INMSG, so they are not processed again.

**Warning:** If you reset the session using the RESET command line parameter you will no longer be able to continue the previous session. Failure to modify the message command file, INMSG, before resetting the session may result in some data being lost or duplicated.

---

24100      Session and Information Exchange checkpoints do not match.

**Explanation:** In a session using checkpoint-level recovery, the checkpoint numbers for the send or receive side of the session do not match the values Information Exchange recorded. Your session file, SESSION, may be damaged.

**User Response:** Reset the session using the RESET command line parameter on the IEBASE command. Also, make sure there is not another user using this user ID. Before starting the next session, review the message response file, OUTMSG, to see which commands were processed successfully. Remove these commands from the message command file, INMSG, so they are not processed again. If the problem persists, contact the Help Desk.

**Warning:** If you reset the session using the RESET command line parameter you will no longer be able to continue the previous session. Failure to modify the message command file, INMSG, before resetting the session may result in some data being lost or duplicated.

---

24200      Invalid time zone.

**Explanation:** You specified an invalid time zone.

**User Response:** Correct the TIMEZONE parameter in the IDENTIFY command in the profile command file, INPRO, and retry the program. If the problem persists, contact the Help Desk.

---

24210 Invalid maximum segments.

**Explanation:** Expedite Base/400 used an invalid value for the MAXMSGs parameter in the TRANSMIT command in the profile command file, INPRO. The valid range of values is 1 to 10.

**User Response:** Correct the MAXMSGs parameter in the TRANSMIT command in the profile command file, INPRO, and retry the program.

---

24220 Invalid reset field.

**Explanation:** Expedite used an invalid value for a field when trying to start the session with Information Exchange.

**User Response:** Contact the Help Desk.

---

24230 Invalid field on session start.

**Explanation:** Expedite Base/400 used an invalid value for a field when trying to start the session with Information Exchange.

**User Response:** Ensure that your Information Exchange User Profile indicates your Information Exchange user ID has the ability to reset sessions. If it does, contact the Help Desk for further assistance.

---

24270 Incorrect Information Exchange password.

**Explanation:** Your Information Exchange password is incorrect.

**User Response:** Correct the IEPASSWORD in the IDENTIFY command in the profile command file, INPRO, or in the START command in the message command file, INMSG, and retry the program. If you just changed your password, make sure you update the IDENTIFY or START command to reflect the new password.

---

24280 Invalid Information Exchange user ID.

**Explanation:** You specified an invalid user ID. Information Exchange does not recognize the account or user ID specified in the START command or the profile.

**User Response:** If a START command was used with ACCOUNT and USERID parameters, make sure they are correct. If the IEACCOUNT and IEUSERID are taken from the profile, make sure you specified them correctly in the IDENTIFY command in the profile command file, INPRO. If the problem continues, contact the Help Desk.

---

24290 Invalid new Information Exchange password.

**Explanation:** You are an Extended Security Option user, and you specified an invalid new Information Exchange password.

**User Response:** Correct the NIEPASSWORD parameter in the IDENTIFY command in the profile command file, INPRO, or in the START command in the message command file, INMSG, and retry the program. ESO passwords have special requirements. Refer to the product documentation for information about these requirements.

---

24300 Invalid password. Userid was revoked.

**Explanation:** You are an ESO user and sent three successive session starts to Information Exchange with incorrect passwords. The Information Exchange user ID has been revoked.

**User Response:** Contact your service administrator to request that your password be reset using Information Exchange Administration Services. Resetting the password restores the user ID.

---

24310 New password is required.

**Explanation:** You are an ESO user and did not specify a new password. If the Information Exchange password for an ESO user is the same as the Information Exchange user ID, the ESO user must specify a new password.

**User Response:** Use the NIEPASSWORD parameter of the IDENTIFY command in the profile command file, INPRO, or of the START command in the message command file, INMSG, to change the password.

---

24320 Error starting session with Information Exchange.

**Explanation:** There was an error trying to start the session with Information Exchange.

**User Response:** Contact the Help Desk.

---

24600 Information Exchange did not return a valid session end response.

**Explanation:** The Information Exchange session may not have ended.

**User Response:** Try to restart the session. If the problem persists, contact the Help Desk.

---

24610 Information Exchange did not end the session due to an error.

**Explanation:** Information Exchange indicated that there was an error, and the session could not end properly.

**User Response:** Try to restart the session. If the problem persists, contact the Help Desk.

---

24620 Dequeue error.

**Explanation:** There was an error trying to dequeue an Information Exchange account or user ID.

**User Response:** If this occurred at the end of the command file, no action is needed. If you have multiple START and END commands in the command file, and it occurred on an earlier END command, try the program again. If the problem continues, contact the Help Desk.



## Session errors

This section describes the return codes for session errors.

---

28000      Warnings generated for the command.

**Explanation:** This return code indicates that warning messages were generated during the command, but the command was able to complete.

**User Response:** Check the WARNING records in the message response file, OUTMSG, for details.

---

28010      The IE session completed normally but not all requests were processed.

**Explanation:** The Information Exchange session completed normally, but not all of the requests in the message command file, INMSG, were processed, or some requests generated warnings.

**User Response:** Check the message response file, OUTMSG, to see which requests did not process normally. Correct those requests in a new INMSG and retry the program.

---

28100      QUERY response indicates warning.

**Explanation:** Information Exchange found one or more errors in the QUERY command but was still able to process the command. Expedite Base/400 may not write AVAILABLE records for some or all of the messages in your mailbox.

Check the error messages in your mailbox to see what caused the error. If needed, correct the error and retry the command.

---

28120      GETMEMBER response indicates a warning.

**Explanation:** The response from Information Exchange to the GETMEMBER command shows that there was a warning while processing the command.

**User Response:** Make sure the library and member that you are trying to retrieve exist, and that you have access to them. If there is a system error message in your mailbox, retrieve it or view it via Information Exchange Administration Services to determine what caused the error. Correct the problem and try the command again.

---

28140      AUDIT response indicates warning.

**Explanation:** Information Exchange found one or more errors in the AUDIT command but was still able to process the command. However, the audit records in your mailbox may be different than requested.

**User Response:** Check the error messages in your mailbox to see what caused the error. If the audit file in the mailbox does not meet your requirements, correct the AUDIT command in the message command file, INMSG, and retry the program.

---

28141      AUDIT response indicates error.

**Explanation:** Information Exchange found one or more errors in the AUDIT command and was unable to process the command.

**User Response:** Check the error messages in your mailbox to see what caused the error. Correct the AUDIT command and retry the command. No audit file has been placed in your mailbox.

---

28160 IE could not verify the destination.

**Explanation:** Information Exchange encountered an error when verifying the destination for the SEND, SENDSTREAM, or SENDEDI command. The data was not sent.

**User Response:** Try the command again. If the problem persists, contact the Help Desk.

---

28170 LISTLIBRARIES response indicates a warning.

**Explanation:** Information Exchange found one or more errors in the LISTLIBRARIES command but was still able to process the command. Expedite Base may not write LIBRARYLIST records for some or all of the libraries you have access to.

**User Response:** Check the error messages in your mailbox to see what caused the error. If needed, correct the error and retry the command.

---

28171 There are more files to be received.

**Explanation:** The session ended successfully, but there are more files in the mailbox to be received.

**User Response:** Process the data already received and run Expedite Base again to receive the additional data. If you switch to checkpoint-, user-, or file-level recovery, you will be able to receive all of the files in your mailbox in a single session without encountering the 28171 return code. Refer to Chapter 5, "Using session-level recovery," for more information.

---

28175 LISTMEMBERS response indicates a warning.

**Explanation:** Information Exchange found one or more errors in the LISTMEMBERS command but was still able to process the command. Expedite Base may not write MEMBERLIST records for some or all of the members in the library you have access to.

**User Response:** Check the error messages in your mailbox to see what caused the error. If needed, correct the error or retry the command.

---

28180 PURGE response indicates an error.

**Explanation:** An unexpected error occurred while trying to process the PURGE command.

**User Response:** Retry the program. If the problem persists, contact the Help Desk.

---

28190 Invalid common data header received.

**Explanation:** Expedite Base/400 received an invalid Common Data Header. The data was received and processed as if no CDH was received.

**User Response:** Verify that the file was received as expected. You may wish to inform the sender that the interface sent an invalid CDH. If the sending interface was an Expedite Base product, contact the Help Desk.

28200 COMMIT command only valid after a SEND, SENDEDI, or PUTMEMBER command.

**Explanation:** COMMIT command only valid after a SEND, SENDEDI, or PUTMEMBER has been specified.

**User Response:** A COMMIT command was specified but there was no SEND, SENDEDI or PUTMEMBER command preceding it. The COMMIT command did not initiate a commit.

## APPC communications support errors

---

29814 Time out waiting for a session with the IE LU 6.2 Relay.

**Explanation:** Expedite Base/400 timed out waiting for an LU 6.2 session with the Information Exchange Common Front End. You need a session on Information Exchange Common Front End for every concurrent user of Expedite Base/400. For example, if five people will be using Expedite Base/400 at the same time, you need five sessions on the Information Exchange Common Front End. This error occurs when there are more concurrent users of Expedite Base/400 on your system than there are available sessions on the Information Exchange Common Front End.

**User Response:** Retry the program when fewer people are running Expedite Base/400 on your system. If you normally have this many people using Expedite Base/400 at the same time, contact the Help Desk to increase the number of sessions available to you on the Information Exchange Common Front End.

---

29850 LU62 DEALLOCATE attempt failed.

**Explanation:** During termination of APPC communications, the DEALLOCATE attempt issued by Expedite did not complete normally. Termination of IEBASE continued.

**User Response:** If the problem persists, activate an AS/400 comm trace (System Service Tools).

---

29960 The open connection failed for SNA LU 6.2 communications.

**Explanation:** Expedite attempted to open a connection, but could not do so successfully, using Intersystem Communications Facility (ICF). The attempt to open the file QSYS/QICDMF failed.

**User Response:** Examine the operating system messages to determine the reason for this error. These will be displayed in the user's job log or the system's error message log.

---

29970 Send data failed for SNA LU 6.2 communications.

**Explanation:** Expedite Base/400 attempted to send data but failed.

**User Response:** Examine the operating system messages to determine the reason for this error. These will be displayed in the user's job log or the system's error message log.

---

29975 Receive data failed for SNA LU 6.2 communications.

**Explanation:** Expedite Base/400 attempted to receive data but failed.

**User Response:** Examine the operating system messages to determine the reason for this error. These will be displayed in the user's job log or the system's error message log.

---

29980 The allocate failed for SNA LU 6.2 communications.

**Explanation:** Expedite Base/400 attempted to allocate a conversation with the Remote LU, but was unable to do so.

**User Response:** This is a general message indicating a failure in acquiring the remote application (IELU). You will need to display your AS/400 job log details to get more information as to the cause of the failure. For example, you may need to check the Remote Location Name, in the configuration lists, to verify that it specifies the correct Remote LU Name, Network ID, etc.

---

29999      Session end response failure.

**Explanation:** Expedite did not receive a session end response from Information Exchange, or a communication failure occurred upon receiving the session end response.

**User Response:** Check the message response file, OUTMSG, or response work file, OUTWORK, to determine if all your commands completed successfully with RETURN records of zeros. No further recovery processing is needed. However, if you were receiving files from your Information Exchange mailbox, check to make sure all data was received by verifying that it is no longer in Information Exchange. This can be done by viewing your mailbox with Information Exchange Administration Services or by running a QUERY command to get a list of AVAILABLE response records for each file in your mailbox. If the data is still in your mailbox, then process the data already received, create an input file to receive only the data still in your mailbox, and run the program again. Otherwise, if there are RETURN records of non-zeros, then perform the appropriate recovery processing for that data.

## GSKIT errors

This section describes the return codes for SSL handshaking errors.

---

30006 Invalid common name on returned certificate.

**Explanation:** The certificate returned during the SSL handshake contains an invalid common name.

**User Response:** The common name that is returned has been written to the BASETRC file if you have requested tracing, and the file exists. If the problem persists, turn on tracing, rerun the job, and send the resulting BASETRC file to GXS for help in determining the cause of this error.

**User Response:** The certificate returned during the SSL handshake matches a certificate listed in the CERTFIL dataset.

---

30007 Un-trusted certificate received.

**Explanation:** The certificate returned during the SSL handshake matches a certificate listed in the CERTFIL dataset.

**User Response:** The certificate serial that has been returned has been written to the BASETRC file if you have requested tracing, and the file exists. The partner certificate serial number matches a serial number listed in the CERTFIL file. If the problem persists, turn on tracing, rerun the job, and send the resulting BASETRC file to GXS for help in determining the cause of this error.

---

30008 User not allowed through the Secure front End (SFE) Gateway.

**Explanation:** The user has no access privileges to the Secure Front End Gateway.

**User Response:** Check that you have specified the correct IP address for the Secure Front End Gateway that you are allowed to communicate with. If you believe that you have the correct IP address, contact the GXS Community Support for how to proceed.

---

30009 Account/Userid does not match certificate's account/userid.

**Explanation:** The user information found in the X.509 certificate does not match the IE system.account.userid that was specified on the IDENTIFY or START command.

**User Response:** Make sure that the information you gave when you requested the certificate matches the IE system, account, and userid that you have requested on the IDENTIFY or START command. Correct the error and retry the program.

---

30010 Invalid Certificate.

**Explanation:** The certificate is not valid for use on this SFE Gateway.

**User Response:** Make sure that the certificate specified was generated by the IGS PKI Profile Server. Correct the error and retry the program. If the problem persists, turn tracing on, retry the program, and contact the Help Desk.

---

30011 Unable to negotiate security specifications.

**Explanation:** The SSL negotiation did not complete with a satisfactory protection level for the SFE Gateway.

If the problem persists, turn tracing on, retry the program, and contact the Help Desk.

---

30012 Connection Temporarily refused.

**Explanation:** The connection to the SFE Gateway has been temporarily refused.

**User Response:** If the problem persists, turn tracing on, retry the program, and contact the Help Desk.

---

30013 SSL unknown error.

**Explanation:** An unexpected SSL error has occurred.

**User Response:** If the problem persists, turn tracing on, retry the program, and contact the Help Desk.

---

30100 SSL API Error: gsk\_environment\_open error.

**Explanation:** The SSL API encountered an error on the gsk\_environment\_open function call.

**User Response:** If the problem persists, turn tracing on, retry the program, and contact the Help Desk.

---

30101 SSL API Error: KEYRINGFILE or APPLICID error.

**Explanation:** The SSL API reported a problem with the KEYRINGFILE or APPLICID parameters.

**User Response:** Check that the KEYRINGFILE or APPLICID parameters are present and correct. If the problem persists, turn tracing on, retry the program, and contact the Help Desk.

---

30102 SSL API Error: KEYRINGFILE error.

**Explanation:** The SSL API encountered an error with the KEYRINGFILE parameter.

**User Response:** Check that the KEYRINGFILE parameter is present and correct. If the problem persists, turn tracing on, retry the program, and contact the Help Desk.

---

30103 SSL API Error: KEYRINGPASSWORD or KEYRINGSTASHFILE error.

**Explanation:** The SSL API reported a problem with the KEYRINGPASSWORD or KEYRINGSTASHFILE parameters.

**User Response:** Check that the KEYRINGPASSWORD or KEYRINGSTASHFILE parameter is present and correct. If the problem persists, turn tracing on, retry the program, and contact the Help Desk.

---

30104 SSL API Error: certificate error.

**Explanation:** The SSL API reports a problem with the certificate.

**User Response:** Check the certificate. If the problem persists, turn tracing on, retry the program, and contact the Help Desk.

---

30105      SSL API Error: CIPHERSUITES error.

**Explanation:** The SSL API encountered a parameter error with the CIPHERSUITES parameter.

**User Response:** Check the CIPHERSUITES parameter to make sure it is correct. If the problem persists, turn tracing on, retry the program, and contact the Help Desk.

---

30106      SSL API Error: SSLVERSION error.

**Explanation:** The SSL API encountered a parameter error with the SSLVERSION parameter.

**User Response:** Check the SSLVERSION parameter to make sure it is correct. If the problem persists, turn tracing on, retry the program, and contact the Help Desk.

---

30107      SSL API Error: AUTHORIZATION error.

**Explanation:** The SSL API encountered a certificate authorization error.

**User Response:** Check the certificate to make sure it is correct. If the problem persists, turn tracing on, retry the program, and contact the Help Desk.

---

30108      SSL API Error: session type error.

**Explanation:** The SSL API encountered a session type error.

**User Response:** If the problem persists, turn tracing on, retry the program, and contact the Help Desk.

---

30109      SSL API Error: gsk\_environment\_init error.

**Explanation:** The SSL API encountered an error on the gsk\_environment\_init function call.

**User Response:** If the problem persists, turn tracing on, retry the program, and contact the Help Desk.

---

30110      SSL API Error: gsk\_secure\_soc\_init error.

**Explanation:** The SSL API encountered an error on the gsk\_secure\_soc\_init function call.

**User Response:** If the problem persists, turn tracing on, retry the program, and contact the Help Desk.

---

30111      SSL API Error: gsk\_secure\_soc\_read error.

**Explanation:** The SSL API encountered an error on the gsk\_secure\_soc\_read function call.

**User Response:** If the problem persists, turn tracing on, retry the program, and contact the Help Desk.

---

30112      SSL API Error: gsk\_secure\_soc\_write error.

**Explanation:** The SSL API encountered an error on the gsk\_secure\_soc\_write function call.

**User Response:** If the problem persists, turn tracing on, retry the program, and contact the Help Desk.

---



---

30113      SSL API Error: APPLICID error.

**Explanation:** The SSL API reported a problem with the APPLICID parameter.

**User Response:** If the problem persists, turn tracing on, retry the program, and contact the Help Desk.

---

30114      SSL API Error: gsk\_secure\_soc\_open error.

**Explanation:** The SSL API encountered an error on the gsk\_secure\_soc\_open function call.

**User Response:** If the problem persists, turn tracing on, retry the program, and contact the Help Desk.

---

30115      SSL API Error: socket descriptor error.

**Explanation:** The SSL API encountered a socket descriptor error.

**User Response:** If the problem persists, turn tracing on, retry the program, and contact the Help Desk.

---

30116      SSL API Error: keyring file label error.

**Explanation:** The SSL API encountered a keyring label error.

**User Response:** Check the keyring file label and make sure it is correct. If the problem persists, turn tracing on, retry the program, and contact the Help Desk.

---

30117      SSL API Error: Cache error.

**Explanation:** The SSL API encountered a cache error.

**User Response:** Check the cache size and make sure it is correct. If the problem persists, turn tracing on, retry the program, and contact the Help Desk.

---

30118      SSL API Error: certificate authority not trusted.

**Explanation:** The certificate authority is not trusted.

**User Response:** Check the certificate authority you have and make sure it is correct. If the problem persists, turn tracing on, retry the program, and contact the Help Desk.

## Unexpected errors

This section describes the return codes for unexpected errors.

---

31000      Unexpected condition found.

**Explanation:** Expedite has encountered an unexpected condition.

**User Response:** Contact the Help Desk. You will be asked to FAX or send your BASETRC file to the Help Desk for problem determination.

---

31344      Invalid message inquiry response.

**Explanation:** Expedite received an unexpected error from Information Exchange while processing a MSGINFO command.

**User Response:** Try the program again later. If the problem persists, contact the Help Desk.

---

31346      Invalid session inquiry response.

**Explanation:** Expedite received an unexpected error from Information Exchange while processing a SESSIONINFO command.

**User Response:** Try the program again later. If the problem persists, contact the Help Desk.

---

31360      Error indication received from Information Exchange.

**Explanation:** Expedite received an unexpected error from Information Exchange.

**User Response:** Wait and retry the program later. Also, make sure there is not another user using this user ID. If the problem persists, contact the Help Desk.

---

31810      File operation *zzzzzzzz* failed on file *xxxxxxx*.

**Explanation:** A checkpoint recovery file was damaged.

**User Response:** Check your file and make sure there are no I/O problems. Check the file permissions to make sure Expedite has access to the file. Reset the session and retry the program. If the problem persists, contact the Help Desk. You will be asked to FAX or send the trace BASETRC to the Help Desk for problem determination.

**Warning:** If you reset the session using the RESET parameter on the IEBASE call command, you will no longer be able to continue the previous session. Failure to modify the message command file, INMSG, before resetting the session may result in some data being lost or duplicated.

---

32000      Expedite Base/400 encountered an unexpected condition.

**Explanation:** Expedite Base/400 encountered an unexpected condition and is unable to continue.

**User Response:** Reset the session and retry the program. If the problem persists, contact the Help Desk. You will be asked to FAX or send the trace file BASETRC to the Help Desk for problem determination.

**Warning:** If you reset the session using the IEBASE parameter on the IEBASE call command, you will no longer be able to continue the previous session. Failure to modify the message command file, INMSG, before resetting the session may result in some data being lost or duplicated.

## Using data compression

---

Expedite Base/400 provides integrated data compression and decompression through components of the TDAccess product provided by bTrade, Inc. This product was formerly known as Comm-Press, and may not be available in all countries. In this programming guide and in error messages, references to either TDAccess or Comm-Press will mean the compression and decompression components.

Compression reduces the size of the files that are transmitted through Information Exchange. Significant savings in network charges and transmission time are possible when using data compression supplied by bTrade, Inc.

Information concerning bTrade, Inc. can be found on their Web site at URL: [www.btrade.com](http://www.btrade.com).

When you use data compression, some Expedite Base/400 parameters are impacted, and session restarts may have to be handled differently. These considerations are described in this appendix.



**NOTE:** When using the bTrade compression or decompression routines, both the sender and receiver of compressed data must have the licensed routines in order to compress and decompress the data.

## Understanding the TDAccess files used with Expedite Base/400

The following modules are provided when you order the TDAccess product:

- INMSGP

This compression program reads the INMSG file, compresses the appropriate files based on the COMPRESS parameter, and builds an INMSGC file reflecting the location of the compressed files for transmission.

- OUTMSGP

This decompression program reads the OUTMSG file after Expedite Base/400 receives the files, then decompresses the received compressed files.

- **ERRORCMP**

This file contains the error messages issued when the TDAccess modules encounter an error.

## Preparing work files for data compression

The following work files are created or required when using Expedite Base/400 with the TDAccess product:

- **COMPWRK** (created in the current library during decompression)
- **CPLOOKUP** (user created and supplied, required only when using **COMPRESS(T)**)
- **INMSGC** (created in the same library that contains **INMSG**)
- **INMSGR** (created in the same library that contains **INMSG**)
- **OUTMSGC** (created in the same library that contains **OUTMSG**)
- **OUTMSGR** (created in the same library that contains **OUTMSG**)
- **COMPTRC** (created in the current library only when using **BASE(Y)** on the **TRACE** command)

### **COMPWRK**

**OUTMSGP** uses this work file during decompression. When compressed data is received, it is decompressed into the **COMPWRK** file. **OUTMSGP** then copies the data back into the received file specified on the **RECEIVE** or **RECEIVEDI** command.

### **CPLOOKUP**

This is the compression lookup table. It specifies whether compression should be performed for a particular sender/receiver pair.

Create **CPLOOKUP** with fixed-length, 80-byte records. See “Compressing files with **COMPRESS(T)**” on page 299 for more information.

### **INMSGC**

**INMSGP** uses this file. **INMSGC** contains all the entries found in **INMSG**, plus the name and location of the corresponding compressed files. **IEBASER** reads **INMSGC** to determine what commands to process. **INMSGC** is deleted upon successful completion of Expedite Base processing. However, if you have specified a data recovery method other than session-level recovery and your Information Exchange session does not complete successfully, **INMSGC** is not deleted and must remain unchanged for successful data recovery processing.

### **INMSGR**

This file is used in recovery situations to synchronize **INMSG** and **INMSGC**. **IEBASEPR** uses this file in restart situations where the **COMPRESS** parameter has been specified on **SEND** and **SENDEDI** commands.

### **OUTMSGC**

**INMSGP** and **OUTMSGP** write compression and decompression errors to this file.

### **OUTMSGR**

**OUTMSGP** uses this file as a work file during decompression. All the messages in **OUTMSG** are copied to **OUTMSGR** as the first step in the decompression process. The messages are updated and copied back to **OUTMSG** as the decompression takes place.

### **COMPTRC**

Trace messages are written to this file when **BASE(Y)** is specified on the **TRACE** command.

### **\$IEnnnnn**

These temporary files have names that begin with “\$IE” and are created in the current library.



**NOTE:** Normally, any temporary files are deleted as part of a successful Expedite Base processing. However, when your Information Exchange session does not complete successfully, the temporary files remain. You should not delete, remove, or rename these files. You should, instead, complete the Information Exchange session or allow RESET processing to remove the temporary files.

## Compressing files with COMPRESS(Y)

When COMPRESS(Y) is included on the SEND or SENDEDI command and INMSGP is available, the specified file is compressed prior to transmission. If the file to be sent contains multiple EDI envelopes, all envelopes are sent compressed. Additional parameters appear in the OUTMSG file along with the echoed SEND command:

```
COMSW(COMM-PRESS) COMVER(230) COMFILE(CURLIB/$IEnnnnn)
```

where COMSW refers to the software providing the compression, COMVER refers to the version and release of that software, and COMFILE refers to the file name for the compressed file to be sent. This information is also contained in the CDH, to allow proper decompression at the receiving locations.

## Compressing files with COMPRESS(T)

When COMPRESS(T) is included on the SEND or SENDEDI command, the specified file is compressed prior to transmission only if the SENDER, RECEIVER, and COMPRESS parameters are listed in the CPLOOKUP file. This file defines a series of paired receivers and senders, and for each pair indicates whether compression should be performed.

Each entry in the compression lookup table must follow this format:

```
sender(sender) receiver(receiver) compress(y|n);
```

### **sender**

Indicates the account and user ID or EDI source of a sender.

### **receiver**

Indicates the account and user ID, alias and aliasname, listname, or EDI destination of a receiver.

### **compress**

Indicates whether compression should be performed for this sender/receiver pair.

- |   |   |
|---|---|
| y | Compress the data for this sender/receiver pair.        |
| n | Do not compress the data for this sender/receiver pair. |

The following is an example of a compression lookup table:

```

SENDER(acct1 user01) RECEIVER(acct1 user02) COMPRESS(y);
SENDER(acct1 user01) RECEIVER(alias1 alias2) COMPRESS(y);
SENDER(acct1 user01) RECEIVER(acct1 user03) COMPRESS(y);
SENDER(acct1 user01) RECEIVER(listname02) COMPRESS(y);

```

For each SEND command, INMSGP identifies the sender from a START command in INMSG or from an IDENTIFY command if AUTOSTART(Y) is specified in INPRO. INMSGP identifies the receiver from the ACCT and USERID, ALIAS and ALIASNAME, or LISTNAME parameters of the SEND command.

When entering values for the sender and receiver parameters, the ACCT or ALIAS must be 7 characters long (padded with trailing blanks, if necessary). The USERID or ALIASNAME must begin in the next character position.

For the SENDEDI command, INMSGP identifies the sender and receiver from the EDI header. INMSGP looks for a corresponding entry in the CPLOOKUP file. TDAccess supports X12, UCS, EDIFACT, and UN/TDI EDI formats. The sender and receiver entries for EDI data must match *exactly* what appears in the appropriate field of the EDI header. The *inmsgp* program examines only the EDI header to resolve the sender/receiver pairs. The Expedite EDI tables are not used. See Chapter 4, “Sending and receiving EDI data,” for a description of which EDI header fields are used to identify the EDI source and destination.

The COMPRESS(T) parameter and the CPLOOKUP file allow you to control what gets compressed, based on the receiver. The CPLOOKUP file, like the INPRO and INMSG files, can be edited and modified when IEBASE is not running.

## Decompressing received files

When compressed files are received, the OUTMSG file contains new parameters used by the OUTMSG program to process the received compressed files. The OUTMSGP program reads the OUTMSG file and decompresses the data. The following new parameters are found in the OUTMSG file for each compressed file received:

```

compress(y) comsw(comm-press) comver(230)
comfile(curlib/$iennnnn) dcmprc(00000)

```

All of the parameters except DCMPRC are obtained from the CDH. When the received file is not compressed, none of the data compression parameters shown above appear in the OUTMSG entry.

The OUTMSGP program copies the OUTMSG file to the OUTMSGR file, then processes and copies each response record back into OUTMSG. When a RECEIVED response record indicates compressed data, OUTMSGP decompresses the data in the received file into the COMPWRK file. Any uncompressed messages contained in the received file are also copied to the COMPWRK file. After successful decompression, all data is copied from COMPWRK back into the received data set.

The DCMPRC parameter in the RECEIVED response record provides a TDAccess return code. You can use this return code to verify successful decompression processing or determine that an error condition exists. The DCMPRC parameter value of ‘00000’ indicates successful decompression processing. If DCMPRC is not ‘00000’, see the related error messages at the end of this appendix to determine the appropriate action. Error messages are also written to the OUTMSGC file.

## Decompressing RECEIVESTREAM data

In some situations, you may want to decompress files directly, instead of through Expedite Base/400. For example, you may be decompressing data from the OUTMSG file that was received as the result of a RECEIVESTREAM command. The DECOMP program is provided for these situations. The DECOMP program is included on the TDAccess installation tape. Refer to the TDAccess *User's Guide* for more information.

## Expedite Base/400 considerations when using COMPRESS(Y) or COMPRESS(T)

Most of the command parameters, when used with the COMPRESS parameter, are supported as documented in this publication. However, the following command parameters function differently when they are used with the COMPRESS(Y) and COMPRESS(T) parameters.

### DATATYPE(EIB) on SEND commands

INMSGP uses the DATATYPE parameter during the compression process to determine whether EBCDIC to ASCII translation should be performed. After compression, the data is sent with DATATYPE(B).

### DELIMIT(CINIL) on SEND commands

INMSGP uses the DELIMIT parameter during the compression process to determine whether the data is delimited with carriage-return or line-feed characters. After compression, the data is sent with DELIMIT(N).

### ENDSTR(endstring) on SENDSTREAM and RECEIVESTREAM commands

This parameter is permitted only when the endstring is on a record boundary.

### RESRECL(SIE) on RECEIVE commands

Decompressed data is always treated as if RESRECL(S) was specified (that is, records are split if they are too long).

### FORMAT(Y)

The FORMAT(Y) parameter is not supported. If FORMAT(Y) is specified on a SEND, SENDEDI, or SENDSTREAM command, an error is returned. If FORMAT(Y) is specified on a RECEIVE, RECEIVEEDI, or RECEIVESTREAM command, it is ignored.

### DLMOVERRIDE(Y/N)

The DLMOVERRIDE parameter is not supported.

### EDIOPT(F)

The EDIOPT(F) parameter is not supported.



**NOTE:** For EDI data sent using the SENDEDI command, the character used as the segment terminator must not be in the range X'EO' to X'FF'.

## Restart and recovery considerations with TDAccess

The logic used in restart and recovery situations, described earlier in this publication, applies to restart and recovery situations where some or all of the files sent and received are compressed.

Because INMSGP processes the INMSG file before Expedite Base/400 processing takes place, INMSGP must determine whether a restart situation exists before it does any processing. If the INMSGC file is not empty, INMSGP assumes that a restart is required.

INMSGP restarts by comparing the INMSGC and OUTMSG files. All commands that were echoed to the OUTMSG file are left unchanged in the INMSGC file and are not reprocessed. Remaining commands in the INMSG file are processed (that is, any requested compression is performed) and copied to the INMSGC file.

Because of the way INMSGP uses the INMSGC file to determine restart status, you should not use an old INMSGC file that might contain data that is still needed. Unpredictable results would occur. You may want to use the RESET runtime parameter each time you run IEBASE, except for known restart situations. The RESTART parameter causes all INMSG commands to be processed.

A special recovery situation can occur during decompression. After received data is decompressed, it is copied from the COMPWRK file back to the received file. If the received file fills up, an error message is written to the OUTMSGC file and processing stops. To recover from this situation, take the following steps:

1. Allocate a larger file.
2. Copy the data out of COMPWRK into the new file.
3. Rerun IEBASE using the DECOMP runtime parameter to complete OUTMSGP processing.

Failure to properly dispose of the data in the COMPWRK file prior to rerunning IEBASE could result in the loss of the received data.

**Restarting the session when using COMPRESS parameter:** In restart situations, you cannot change certain Expedite Base/400 files, such as the INMSG and OUTMSG files. You also cannot change the INMSGC file. The program uses the INMSGC file when it processes compressed files.

**Restart the session after modifying INMSG:** As noted earlier in this publication, you can modify entries in INMSG that are in error and restart the session. However, you cannot modify INMSG entries that have already been echoed to the OUTMSG file.

**Using IEBASE RESET:** When you use IEBASE RESET, you must remove the completed SEND and RECEIVE requests from the INMSG file.

**Restarting the session for OUTMSGP processing:** When the DCMPC parameter indicates an error condition, you may be able to correct the error condition and restart the OUTMSGP processing to decompress received files. To restart just the OUTMSGP decompression processing, specify the DECOMP parameter with IEBASE as follows:

```
call iebase parm('decomp')
```

**Decompressing files independent of Expedite Base/400:** In some error recovery situations, you may want to decompress files directly, instead of through Expedite Base/400. Use the DECOMP program to do this, as explained in “Decompressing RECEIVESTREAM data” on page 301.



## Error messages and return codes for data compression

The INMSGP and OUTMSGP modules write error messages to OUTMSGC for error conditions that require user intervention. The modules set the AS/400 completion code to 8. OUSTMGP also updates the DCMPRC field in the RECEIVED response record to indicate the results of decompressing that received message.

Some decompression errors do not prevent further processing, and only result in a non-zero value being placed in the DCMPRC field of the appropriate RECEIVED response record. For example, assume that a compressed message is corrupted during transmission. The cyclic-redundancy check (CRC) would fail, resulting in a DCMPRC return code of 26015. Processing would continue with the next RECEIVED response record and, if no other session errors occurred, the SESSIONEND return code would be changed to 28010, indicating that not all commands were processed successfully.

---

26001      Error allocating memory.

**Explanation:** A request for storage failed.

**User Response:** Examine the operating system error message to determine the cause of the error. Correct the problem and restart the session. If the error occurred during decompression, use the DECOMP parameter when restarting the session.

---

26003      Invalid use of COMPRESS parameter.

**Explanation:** COMPRESS(Y) was specified on a SEND or SENDEDI command but the Comm-Press data compression software was not found.

**User Response:** Contact your marketing representative to acquire the Comm-Press data compression software.

---

26004      Unable to decompress received files.

**Explanation:** Compressed files were received, but the Comm-Press data compression software was not found.

**User Response:** Contact your marketing representative to acquire the Comm-Press data compression software.

---

26005      Invalid value for COMPRESS parameter.

**Explanation:** Valid values for the COMPRESS parameter are *n*, *y*, and *t*.

**User Response:** See “SEND command” on page 129, “SENDEDI command” on page 135, and “SENDSTREAM command” on page 140 for instructions regarding the use of the COMPRESS parameter.

---

26006      Format parameter not valid with COMPRESS(Y).

**Explanation:** The FORMAT parameter is not supported when using compression.

**User Response:** Remove the FORMAT parameter from the SEND command.

---

26007      Cannot compress for reserved message class.

**Explanation:** COMPRESS(Y) cannot be used with the specified message class, which is reserved for network interconnects.

**User Response:** Change the message class and retry the session.

---

26008      TRANSLATE parameter not valid with COMPRESS(Y).

**Explanation:** The TRANSLATE parameter is not supported when using compression.

**User Response:** Remove the TRANSLATE parameter from the SEND command.

---

26009      Compressed segment not found.

**Explanation:** A RECEIVED response record indicates compression data, but the compressed segment is not in the received file. This may be the result of an error in transmission.

**User Response:** Request that the sender retransmit the file, then receive it again.

---

26010      Error creating temporary file name.

**Explanation:** A file error occurred.

**User Response:** Examine the operating system error message to determine the cause of the error. Correct the problem and restart the session. If the error occurred during decompression, use the DECOMP parameter when restarting the session.

---

26015      Compressed segment in error.

**Explanation:** A RECEIVED response record indicates compressed data, but the compressed segment failed CRC validation. The segment is removed from the input file, and processing continues.

**User Response:** You must receive the file again.

---

26018      Invalid EDI envelope.

**Explanation:** The EDI header or trailer does not conform to the EDI standard, or a premature end-of-file condition was encountered on input. This error can be caused by a missing or incorrect segment terminator in the EDI header.

**User Response:** Correct the mapping of the EDI data and retry the command.

---

26020      Error opening input file: filename.

**Explanation:** A file error occurred.

**User Response:** Examine the operating system error message to determine the cause of the error. Correct the problem and restart the session. If the error occurred during decompression, use the DECOMP parameter when restarting the session.

---

26021      Error reading input file: filename.

**Explanation:** A file error occurred.

**User Response:** Examine the operating system to determine the cause of the error. Correct the problem and restart the session. If the error occurred during decompression, use the DECOMP parameter when restarting the session.

---

26024      Error positioning input file: filename.

**Explanation:** A file error occurred.

**User Response:** Examine the operating system to determine the cause of the error. Correct the problem and restart the session with the DECOMP parameter.

---

26030      Error opening output file: filename.

**Explanation:** A file error occurred.

**User Response:** Examine the operating system error message to determine the cause of the error. Correct the problem and restart the session. If the error occurred during decompression, use the DECOMP parameter when restarting the session.

---

26032      Error writing output file: filename.

**Explanation:** A file error occurred.

**User Response:** Examine the operating system error message to determine the cause of the error. Correct the problem and restart the session. If the error occurred during decompression, use the DECOMP parameter when restarting the session.

---

26033      Error closing output file: filename.

**Explanation:** A file error occurred.

**User Response:** Examine the operating system error message to determine the cause of the error. Correct the problem and restart the session. If the error occurred during decompression, use the DECOMP parameter when restarting the session.

---

26034      Error copying output file: filename.

**Explanation:** A file error occurred.

**User Response:** Examine the operating system error message to determine the cause of the error. Correct the problem and restart the session. If the error occurred during decompression, use the DECOMP parameter when restarting the session.

---

26040      Error opening work file.

**Explanation:** A file error occurred.

**User Response:** Examine the operating system error message to determine the cause of the error. Correct the problem and restart the session.

---

26041      Error reading work file.

**Explanation:** A file error occurred.

**User Response:** Examine the operating system error message to determine the cause of the error. Correct the problem and restart the session.

---

26042      Error writing work file.

**Explanation:** A file error occurred.

**User Response:** Examine the operating system error message to determine the cause of the error. Correct the problem and restart the session.

---

26060      Error opening file: filename.

**Explanation:** An error occurred opening the indicated file.

**User Response:** Examine the operating system error message to determine the cause of the error. Correct the problem and restart the session. If the error occurred during decompression, use the DECOMP parameter when restarting the session.

---

26061      Error reading file: filename.

**Explanation:** An error occurred reading the indicated file.

**User Response:** Examine the operating system error message to determine the cause of the error. Correct the problem and restart the session. If the error occurred during decompression, use the DECOMP parameter when restarting the session.

---

26062      Error writing file: filename.

**Explanation:** An error occurred writing the indicated file.

**User Response:** Examine the operating system error message to determine the cause of the error. Correct the problem and restart the session. If the error occurred during decompression, use the DECOMP parameter when restarting the session.

---

26063      Error closing file: filename.

**Explanation:** An error occurred closing the indicated file.

**User Response:** Examine the operating system error message to determine the cause of the error. Correct the problem and restart the session. If the error occurred during decompression, use the DECOMP parameter when restarting the session.

---

26064      Error positioning file: filename.

**Explanation:** An error occurred positioning the indicated file.

**User Response:** Examine the operating system error message to determine the cause of the error. Correct the problem and restart the session. If the error occurred during decompression, use the DECOMP parameter when restarting the session.

---

26065      Error copying: filename.

**Explanation:** An error occurred copying the indicated file.

**User Response:** Examine the operating system error message to determine the cause of the error. Correct the problem and restart the session. If the error occurred during decompression, use the DECOMP parameter when restarting the session.

---

26066      EOF reached before end of command.

**Explanation:** The SEND or SENDEDI command was not completed before End-of-File. The command is probably missing the terminating semicolon.

**User Response:** Correct the command and restart the session.

---

26067      Error removing: filename.

**Explanation:** An error occurred removing the indicated file.

**User Response:** Examine the operating system error message to determine the cause of the error. Correct the problem and restart the session. If the error occurred during decompression, use the DECOMP parameter when restarting the session.

---

26068      Error renaming: filename.

**Explanation:** An error occurred renaming the indicated file.

**User Response:** Examine the operating system error message to determine the cause of the error. Correct the problem and restart the session. If the error occurred during decompression, use the DECOMP parameter when restarting the session.

---

26098      Invalid Comm-Press version.

**Explanation:** The received file was compressed with a later version of Comm-Press than was used to decompress the file.

**User Response:** Contact your marketing representative to acquire the latest version of the Comm-Press data compression software.

---

26099      Compressed file not decompressed.

**Explanation:** The received file is compressed and DECOMP processing has not been performed.

**User Response:** Run IEBASE with the DECOMP command line parameter.

---

26nnn      Unknown error occurred.

**Explanation:** An unexpected error occurred during Comm-Press processing.

**User Response:** Contact Comm-Press, Inc. technical support for help in resolving this problem.





## Glossary

---

This glossary defines words as they are used in this book. It includes terms and definitions from the *IBM Dictionary of Computing* (New York: McGraw-Hill, 1994). If you are looking for a term and cannot find it here, see the *Dictionary of Computing* for additional definitions.

This glossary includes terms and definitions from:

- *The American National Dictionary for Information Systems*, ANSI x3.172-1990, copyright 1990 by the American National Standards Institute (ANSI). Copies may be purchased from the American Standards Institute, 11 West 42 Street, New York, New York 10036. Definitions are identified by the symbol (A) after the definition.
- *The Information Technology Vocabulary*, developed by Subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC JTC1/SC1). Definitions of published parts of this vocabulary are identified by the symbol (I) after the definition.

### A

abend. Abnormal end of task, as in “the session abends.”

account name. A name that identifies an account to a program, device, or system.

acknowledgment. A response from Information Exchange that lets you know whether your messages were delivered, received, purged, or various combinations of these three.

address. (1) A unique code assigned to a user connected to a network. (2) The location in the storage of a computer where data is stored.

advanced interactive executive (AIX). An operating system that serves as an interface between users and Information Exchange.

AIX. Advanced Interactive Executive.

alias. An alternate name used in place of an account and user ID.

alias table. An alternate name file. Expedite Base/400 uses alias tables to resolve EDI destinations.

alphanumeric. Generally, any keyboard character, but for practical purposes should be restricted to alphabetic, numeric, space, and common punctuation characters.

American National Standard Code for Information Interchange (ASCII). The standard code, using a coded character set consisting of 7-bit coded characters (8 bits include parity check), used for information interchange among data processing systems, data communication systems, and associated equipment. The ASCII set consists of control characters and graphic characters. (A)

application program interface (API). The formally defined programming language interface used by an IBM system control program or a licensed program to communicate with the user of the program.

archive. A place to store messages on a database for future reference.

ASCII. American National Standard Code for Information Interchange.

asynchronous. A protocol that permits a communication device to operate in an unsynchronized and unpredictable manner, much like a human conversation; used for modems and low-speed ASCII terminals (PCs).

attribute. A property or characteristic of one or more entities, such as length, value, color, or intensity.

audit trail. A way of tracking and verifying basic information about the status of messages.

## B

base trace file (BASETRC). The file that provides Expedite Base/400 trace information (other than link trace). You need to request this trace only when working with IBM on a problem.

BASETRC. Base trace file.

Batch Data Interface (BDI). A product that allows users to communicate with Information Exchange if they have a remote job entry processor or terminal.

BDI. Batch Data Interface.

binary file. A file, such as an executable computer program, that contains machine instructions that a person cannot read and that cannot be entered from a computer keyboard.

binary synchronous (bisynchronous). A protocol that allows a communication device to operate in a synchronized and predictable manner; used for high-speed computer-to-computer communication and for mainframe computer-to-nonprogrammable terminal communication.

## C

carriage-return and line-feed characters (CRLF). A formatting control that moves the printing or display point to the first position of the next line.

CDH. Common data header.

CDRSC. Cross-domain resource.

centralized alias table. Permanent tables that reside within Information Exchange and contain a centralized list of addresses. You can put a listing of your trading partners' addresses in this table instead of maintaining destination tables in multiple locations. A centralized alias table enables Expedite Base/400 to resolve destinations, because the table contains a list of EDI destinations paired with Information Exchange destinations. Expedite Base/400 searches this table for an EDI destination and then uses the corresponding Information Exchange destination as the actual address.

certificate. A digital document that binds a public key to the identity of the certificate owner; thereby, enabling the certificate owner to be authenticated. A certificate is issued by a certificate authority (CA).

certificate authority (CA). A digital document that binds a public key to the identity of the certificate owner; thereby, enabling the certificate owner to be authenticated. A certificate is issued by a certificate authority (CA).

character. Generally, any letter, number, punctuation mark, or other symbol that can be entered at a computer keyboard.

checkpoint-level recovery. A method of restart and recovery within Expedite Base/400; a point where information about the status of a job can be recovered so that the job step can be restarted later.

CICS. Customer Information Control System.

command. A request to run a particular program or function.

command file. A file containing Expedite Base/400 commands. There are two Expedite Base/400 command files, profile command (INPRO) and message command (INMSG). You place commands pertaining to your profile in INPRO and those pertaining to the transfer of files or data in INMSG.

committal. The point at which a message is either delivered, canceled, or purged. When a session fails, all uncommitted messages are lost.

common data header (CDH). The first message in a group of messages, the common data header provides information about the messages.

compression. The process of eliminating gaps, empty fields, and redundant data to shorten the length of files.



CRLF. Carriage-return and line-feed characters.

Customer Information Control System (CICS). A program that enables transactions entered at remote terminals to be processed at the same time.

## D

default. An alternative value an application program uses automatically when none has been specified.

delivery acknowledgment. An acknowledgment generated by Information Exchange when a destination user receives a message from the Information Exchange mailbox.

destination. In a network, any point or location, such as a node, station, or a particular terminal, to which information is sent.

distribution list. A list of the addresses of users with whom a certain user communicates. It is used to send messages to several people without having to type their addresses.

## E

EBCDIC. Extended binary-coded decimal interchange code.

EDI. Electronic data interchange.

EDI destination table. A list of EDI destinations paired with Information Exchange destinations used by Expedite Base/400.

EDIFACT. An electronic data interchange standard developed by the United Nations Economic Commission for Europe.

EDI qualifier table. A list of types of EDI data (X12, UCS, EDIFACT, or UN/TDI) paired with the ID qualifier for a particular type of data (for example, 01 for an X12 DUNS number).

electronic data interchange (EDI). The process of sending specially formatted business documents directly from one computer to another electronically.

electronic data interchange control files. Files used to send EDI data: EDI qualifier tables (QUALTBL) and EDI destination tables (TTABLExx).

envelope. Information Exchange's electronic file that contains the text of a message, together with a message header.

error message file (ERRORMSG). The file that provides Expedite Base/400 with error descriptions in the RETURN, WARNING, SESSIONEND, and PROFILERC output records.

ERROR.MSG. Error message file.

ERRORTXT. Extended error text file.

ESO. Extended Security Option.

extended binary-coded decimal interchange code (EBCDIC). A coded character set consisting of 256 8-bit characters.

extended error text file (ERRORTXT). The file that provides Expedite Base/400 with extended error descriptions in the RETURN, WARNING, SESSIONEND, and PROFILERC response records.

Extended Security Option. An option that provides additional password and Information Exchange mailbox security.

## F

file-level recovery. A method of restart and recovery within Expedite Base/400; checkpoints are taken for each file sent and received.

## G

global alias. An alternative name that can be used on a particular system.

global alias table. (1) A system-wide alias table. (2) An alternative name table set up within a system.

## H

host computer. The primary or controlling computer in a multiple computer installation.

## I

ibm0rely. The LU name of the Information Exchange system in the United States.

IEBASE. Information Exchange base program.

Information Exchange. (1) A communication service that permits users to send and receive information electronically. (2) A continuously running CICS application on the network that stores and forward information to trading partners.

Information Exchange Administration Services. An online, panel-driven product that the Information Exchange Service Administrator uses to perform administrative tasks for Information Exchange.

Information Exchange base (IEBASE) program. The program that starts Expedite Base/400.

Information Exchange mailbox. See *mailbox*.

Information Exchange service administrator. The person who coordinates the use of Information Exchange within a company.

INMSG. Expedite Base/400's message command file.

INPRO. Expedite Base/400's profile command file.

## L

leased lines. A connection between systems or devices that does not have to be made by dialing.

library. A place to store information for an extended period of time. A library consists of a collection of files called library members.

link trace file (LINKTRC). The file that provides Expedite Base/400 with link trace information. You need to request a link trace only when working with IBM on a problem.

LINKTRC. Link trace file.

## M

mailbox. A temporary storage area for electronic mail from which data is retrieved by the intended recipient.

member. A file in a library.

message. (1) Any piece of data that users send or receive. (2) The smallest subdivision of information that can be sent from one user to another. (3) An

instruction or explanation on the screen that tells you what the system is doing or warns you that the system has detected an error.

message class. A category agreed upon among trading partners that is used to group mail.

message command. Commands that pertain to the transferring of files or data. You place message commands in the message command file (INMSG).

message command file (INMSG). The file that sends and receives files and messages, including EDI data. The name of this file is INMSG. You enter file transfer requests with message commands into INMSG.

message group. A collection of messages that is treated as a single entity. A file of records to be printed as a single report is an example of a message group.

message response file (OUTMSG). A file containing Expedite Base/400's and Information Exchange's reply to certain message commands. The name of this file is OUTMSG. Expedite Base/400 generates this file after it processes the message command file (INMSG). OUTMSG contains return codes, error messages, and completion codes.

## N

NCP. See *network control program*.

Network control program (NCP). An IBM licensed program that provides communication controller support for single-domain, multiple-domain, and interconnected network capability.

## O

organizational alias. (1) An alias that can be used by any user in an account. (2) A company-wide alias table.

organizational alias table. An alias table set up within an account.

OUTMSG. The message response file for Expedite Base/400.

OUTPRO. The profile response file for Expedite Base/400.

## P

parameter. (1) A variable that is given a constant value for a specified application and that may denote the application. (I) (A) (2) An item in a menu for which you specify a value or for which the system provides a value when the menu is interpreted. (3) Data passed between programs or procedures.

private alias table. An alias table set up for an individual user.

profile command file (INPRO). A file in which you place profile commands pertaining to profile-specific information, such as your passwords, user ID, and account ID. The name for this file is INPRO.

profile response file (OUTPRO). A file containing Expedite Base/400's and Information Exchange's replies to certain profile commands. The name for this file is OUTPRO. Expedite Base/400 generates this file after processing the profile command file (INPRO). The profile response file contains return codes, error messages, and completion codes.

purge acknowledgment. The acknowledgment generated by Information Exchange when a message reaches the storage time limit and is purged from the receiver's mailbox.

## Q

qualifier table. A list of types of EDI data (X12, UCS, EDIFACT, or UN/TDI) paired with the ID qualifier for a particular type of data (for example, 01 for an X12 DUNS number).

## R

receipt acknowledgment. The acknowledgment generated by Information Exchange when a message reaches the receiver's mailbox after a successful Expedite Base/400 session. It appears in the message response file (OUTMSG).

recovery level. The point from which a session needs to be restarted following a failure. Expedite Base/400 has four recovery levels: session-, file-, user-, and checkpoint-level recovery.

response file. An output file into which Expedite Base/400 echoes commands from INPRO and INMSG along with their associated return codes.

## S

Secure Socket Layer . A secure method of communicating by way of the Internet.

service administrator. A primary contact person in an organization who controls use of the service by the users within the organization and who assists the service support groups (for example, Help Desk).

session. The period of time during which you can communicate with a computer system or one of its programs; usually, the elapsed time between logon and logoff.

session-level. A method of restart and recovery within Expedite Base/400; no messages are committed until the session ends normally.

SSL. See *Secure Socket Layer*.

syntax. The rules for constructing a command.

## T

TCP/IP. Transmission Control Protocol/Internet Protocol.

Transmission Control Protocol/Internet Protocol (TCP/IP). A set of communications protocols that support peer-to-peer connectivity functions for both local and wide area networks.

trace file. A file that provides you with a history of transactions. In Expedite Base/400 there are two types of trace files: base trace (BASETRC) and link trace (LINKTRC).

trace message text file (TRACEMSG). A file containing trace messages in languages other than English.

trading partners. The business associates with whom users of EDI exchange information electronically.

## U

UCS. Uniform Communication Standard.

Uniform Communication Standard (UCS). A standard EDI format used in the grocery industry.

United Nations/Trade Data Interchange (UN/TDI). An EDI standard for administration, commerce, and transportation fields developed by the United Nations Economic Commission for Europe.

user-initiated recovery. A method of restart and recovery within Expedite Base/400; checkpoints are taken after each COMMIT command, unless there is nothing to commit.

user class. A short description that users can assign to their documents to identify these documents to their trading partners.

user group. A set of users within an account who can communicate with one another.

USERLUNAME. The LU name that Expedite Base/400 uses when communicating with Information Exchange. Your system programmer must define it to your AS/400 system.

user message class. A category agreed upon among trading partners that is used to group mail.

user profile. A list of the characteristics that describe how a user works with the Information Exchange.

## W

wildcard character. A synonym for pattern-matching character.

## X

X12. An electronic data interchange standard developed by the American National Standards Institute (ANSI).



# Index

---

## A

- account ID 309
- acknowledgments 5
- addresses, validating 65
- alias tables
  - SENDEDI command 29
  - sending EDI data 36
  - using 36
- American National Standard Code for Information Interchange (ASCII) 309
- API commands
  - appearance in text ix
  - definition 309
  - purpose 4
  - syntax 4
- ARCHIVEMOVE command 86
- ASCII operating systems
  - communicating with 24
  - general discussion 24
- attributes
  - BASETRC 18
  - INMSG 14
  - INPRO 13
  - LINKTRC 18
  - OUTMSG 15
  - OUTPRO 14
- AUDIT command 87
- authorizations, validating 67
- AUTOEND record 155
  - format 155
- AUTOSTART record 156
  - format 156
- AVAILABLE record 157
  - format 157

## B

- base trace file (BASETRC)

- attributes 18
  - description 17
- BASETRC
  - attributes 18
  - description 17
- binary data
  - considerations 23
  - general discussion 24
  - in a CDH 23
  - receiving from a host operating system 26
  - receiving from an ASCII operating system 24
  - sending to a host operating system 25
  - sending to an ASCII operating system 24
  - sending to host operating system 26
- binary file 310

## C

- calling IEBASE parameters 20
- CANCEL command 90
- carriage-return and line-feed characters (CRLF) 310
- CDH
  - See* common data header (CDH)
- centralized alias tables
  - SENDEDI command 29
  - sending EDI data 36
  - using 36
- CERTFIL 19
- certificates
  - validation file 19
- character 310
- checkpoint-level recovery
  - changing files on restart 63
  - files to create 12, 16
  - general discussion 55
  - resetting a session 64
  - response record format
    - checking return codes 59

- examples 60
  - processing the message response file 58
  - restarting after an error 62
  - setting up work files
    - considerations 57
    - creating the output work file (OUTWORK) 58
    - creating the receive work file (RCVWORK) 57
    - creating the session file (SESSION) 57
  - command and response files 11
  - commands
    - ARCHIVEMOVE 86
    - AUDIT 87
    - CANCEL 90
    - COMMIT 93
    - DEFINEALIAS 94
    - END 97
    - GETMEMBER 98
    - IDENTIFY 74
    - LIST 101
      - limitations 101
    - LISTLIBRARIES 104
    - LISTMEMBERS 105
    - LISTVERIFY 106
    - MSGINFO 109
    - PURGE 110
    - PUTMEMBER 111
    - QUERY 115
    - RECEIVE 116
    - RECEIVEEDI 122
      - EDI destination table entry format 44
      - EDI qualifier table entry format 43
      - receiving EDI data 42
    - RECEIVESTREAM 126
    - SEND 129
    - SENDEDI 135
      - blanks following EDI segments 31
      - EDI destination table entry format 44
      - EDI envelopes 30
      - EDI qualifier table entry format 43
      - locating EDI destinations 31
      - overview 29
      - resolving the EDI destination 32
      - sending EDI data 31
      - specifying Information Exchange control fields 40
    - SENDSTREAM 140
    - SESSIONINFO 144
    - SNACOMM 77
    - SSL 78
    - START 145
    - syntax example 71
    - syntax limitations 72
    - TCPCOMM 79
    - TESTMSG 147
    - TRACE 80
    - TRANSMIT 82
      - autostart option 12
  - commands, message
    - syntax examples 72
  - commands, profile
    - syntax examples 72
  - comments in tables 43, 45
  - COMMIT command 93
  - common data header (CDH)
    - characteristics 23
    - construction of 217
    - format 217
    - use by Expedite Base/400 23
    - user data 218
  - communicating with other systems
    - introduction 23
    - to older Information Exchange interfaces
      - common data header unsupported 27
      - considerations 27
  - completion codes 224
  - compression of data 297
- ## D
- data compression 297
  - data recovery 83
  - decompression of data 297
  - DEFINEALIAS command 94
  - destination verification errors 267
  - distribution lists, EDI 30
- ## E
- EBCDIC
    - translating to ASCII 24, 25
    - use in AS/400 2
  - EDI data
    - receiving
      - EDI envelopes 42
      - general discussion 42
      - using the AUTOEDI parameter 42
      - using the RECEIVEEDI command 42
    - sending
      - blanks following EDI segments 31
      - EDI envelopes 30
      - general discussion 31
      - resolving the EDI destination 33

- SENDEDI command overview 29
- EDI destination table (TTABLExx)
  - destination table 31
  - entry format 44
  - optional file 19
  - parameter descriptions 44
  - SENDEDI command 29
  - using 33, 35
- EDI distribution lists 30, 31, 39
- EDI envelope
  - data type definition 29, 30
  - general discussion 30
  - SENDEDI command 29
- EDI qualifier table (QUALTBL)
  - entry format 43
  - optional file 19
  - parameter descriptions 43
  - qualifier table 31
  - SENDEDI command 29
  - using 33
- END command 97
- error message file (ERRORMSG)
  - description 17
- error messages
  - description 17
  - identifying Expedite Base/400 error messages 5
  - identifying Information Exchange error messages 5
  - reason codes 225
  - return codes
    - file validation errors 278
- ERRORMSG
  - See* error message file (ERRORMSG)
- ERRORTXT
  - See* extended error text (ERRORTXT)
- Expedite Base/400
  - additional features
    - addresses, validating 67
    - authorizations, validating 67
    - free-format messages 67
    - payment levels, validating 67
  - checkpoint-level recovery
    - changing files on restart 63
    - general discussion 55
    - message response file considerations 58
    - resetting a session 64
    - response record format 58
    - restarting after an error 62
    - selecting 56
    - setting up work files 57
  - common data header (CDH)
    - construction of 217
    - general discussion 217
    - user data 218
  - data recovery files 16
  - EDI data, receiving
    - general discussion 42
    - RECEIVEEDI command overview 42
  - EDI data, sending
    - blanks following EDI segments 31
    - EDI envelopes 30
    - general discussion 31
    - resolving the EDI destination 32
    - SENDEDI command overview 29
  - file-level recovery
    - changing files on restart 63
    - general discussion 56
    - message response file considerations 58
    - resetting a session 64
    - response record format 58
    - restarting after an error 62
    - selecting 56
    - setting up work files 57
  - installing 187
  - introducing
    - acknowledgments 5
    - API commands 4
    - command and response files 11
    - Information Exchange 1
    - system security 8
  - objects 210
  - optional files 17
  - received files 16
  - return codes
    - file validation errors 278
  - session-level recovery
    - checking command RECEIVED records 50
    - checking command SENT records 50
    - checking return codes 49
    - OUTMSG examples 51
  - user-initiated recovery
    - changing files on restart 63
    - general discussion 55
    - message response file considerations 58
    - resetting a session 64
    - response record format 48, 58
    - restarting after an error 62
    - selecting 56
    - setting up work files 57
    - using the common data header 23
- extended error text file (ERRORTXT)
  - description 17
- extended security option 8

**F**

- file validation error return codes 278
- file-level recovery 56
- files
  - creating
    - general discussion 11
  - optional
    - base trace file (BASETRC) 17
    - error message file (ERRORMSG) 17
    - extended error text file (ERRORTXT) 17
    - status display file (MESSAGE) 17
    - trace message text (TRACEMSG) 18
  - received 16
  - required
    - message command file (INMSG) 14
    - message response file (OUTMSG) 15
    - profile command file (INPRO) 12
    - profile response file (OUTPRO) 14
- free-format messages 67

**G**

- general environment errors 279
- GETMEMBER command 98

**I**

- IDENTIFY command 74
- IEBASE parameters
  - command form 20
  - parameters 20
  - statement example 21
- Information Exchange
  - centralized alias tables 36
  - common data header
    - construction of 217
    - general discussion 217
    - user data 217
  - general discussion 1
  - identifying error messages 5
  - specifying control fields
    - message name (MSGNAME) 40
    - MSGSEQNO 40
    - MSGUCLS 41
  - using a ZZ qualifier 37
- Information Exchange messages 223
- INMSG file
  - attributes 14
  - general discussion 14
  - sample 15
- INPRO file

- attributes 13
- general discussion 12
- required IDENTIFY parameters 13
- required SNACOMM parameter 13
- sample 13

- input to Expedite Base/400 4
- installing the Expedite Base/400 program 187

**L**

- libraries
  - adding and retrieving 111
  - copying and retrieving 98
- LIBRARYLIST record 162
  - format 162
- link trace file (LINKTRC)
  - attributes 18
- LINKTRC
  - attributes 18
- LIST command 101
  - limitations 101
- LISTLIBRARIES command 104
- LISTMEMBERS command 105
- LISTVERIFY command 106
- logical unit (LU) name
  - defining 190, 196
  - ordering 190

**M**

- MEMBERLIST record 164
  - format 164
- MEMBERPUT record 166
  - format 166
- message command file (INMSG)
  - attributes 14
  - general discussion 14
  - sample 15
- message commands
  - description 84
- message response file
  - attributes 15
  - general discussion 15
  - processing for session-level recovery
    - checking command RECEIVED records 50
    - checking command SENT records 50
    - checking return codes 49
    - OUTMSG examples 51
    - response record format 48
  - sample 15
  - using checkpoint-level recovery



- considerations 58
  - processing 58
  - selecting 56
  - setting up work files 57
- message response file (OUTMSG) 153
- MOVED record 167
  - format 167
- MSGINFO command 109
- MSGNAME, assigning 40
- MSGRESP record 168
  - format 168
- MSGSEQNO, assigning 40
- MSGUCLS
  - assigning for EDIFACT and UN/TDI data 41
  - assigning for X12 and UCS data 41

## N

- NCP dial connection 196
- NCP leased connection 190
- NOTSENT record 170
  - format 170
- number-sign 71

## O

- older interface considerations 27
- OUTMSG example 60
- OUTMSG file 153
  - attributes 15
  - general discussion 15
  - processing for session-level recovery
    - checking command RECEIVED records 50
    - checking command SENT records 50
    - checking return codes 49
    - OUTMSG examples 51
    - response record format 48
  - sample 15
  - using checkpoint-level recovery
    - considerations 58
    - processing 58
    - selecting 56
    - setting up work files 57
- OUTPRO file
  - attributes 14
  - general discussion 14
  - records 149
  - response records 149
  - sample 14
- OUTPRO response records
  - introduction 149
- output from Expedite Base/400 4

- output work file (OUTWORK)
  - contents 149
  - creating 58
- OUTWORK
  - See* output work file (OUTWORK)
- OUTWORK example 60

## P

- parameter 313
- parameters
  - command syntax examples 72
- parser errors 265
- payment levels, validating 67
- profile command file (INPRO)
  - attributes 13
  - general discussion 12
  - required IDENTIFY parameters 13
  - required SNACOMM parameter 13
  - sample 13
- profile commands
  - general discussion 73
- profile response file
  - attributes 14
  - general discussion 14
  - record
    - introduction 149
    - PROFILERC 150
    - sample 14
- PROFILERC record 150
  - format 150
- PURGE command 110
- PUTMEMBER command 111

## Q

- QUERY command 115

## R

- RCVWORK
  - See* receive work file (RCVWORK)
- reason codes 225
- RECEIVE command 116
- receive work file (RCVWORK)
  - creating 57
- received files 16
- RECEIVED record 172
  - format 172
- RECEIVED records 50
- RECEIVEEDI command 122
- RECEIVESTREAM command 126

- receiving EDI data 42
    - general discussion 42
  - records
    - AUTOEND 155
    - AUTOSTART 156
    - AVAILABLE 157
    - format
      - AUTOEND 155
      - AUTOSTART 156
      - AVAILABLE 157
      - LIBRARYLIST 162
      - MEMBERLIST 164
      - MEMBERPUT 166
      - MOVED 167
      - MSGRESP 168
      - NOTSENT 170
      - PROFILERC 150
      - RECEIVED 172
      - RETURN 151, 178
      - SENT 179
      - SESSIONEND 181
      - SESSIONRESP 182
      - WARNING 152, 186
    - LIBRARYLIST 162
    - MEMBERLIST 164
    - MEMBERPUT 166
    - MOVED 167
    - MSGRESP 168
    - NOTSENT 170
    - PROFILERC 150
    - RECEIVED 172
    - RETURN 151, 178
    - SENT 179
    - SESSIONEND 181
    - SESSIONRESP 182
    - WARNING 152
  - recovery of data 83
  - recovery, user-initiated 93
  - resetting Expedite Base/400 session
    - determining data already transmitted 64
    - removing completed requests 64
    - RESET command 64
  - response files
    - checkpoint-level recovery
      - examples 60
  - response record
    - format, session-level recovery 48
    - OUTPRO
      - PROFILERC 150
    - SENDEDI 41
    - using checkpoint-level recovery
      - checking return codes 59
      - considerations 58
      - format 58
      - processing the message response file 58
  - response records 149
  - restarting
    - after an error 62
    - changing files 63
  - return codes
    - file validation errors 278
    - for non-session-level recovery 59
    - for session-level recovery--checking 49
  - RETURN record 151, 178
    - format 151, 178
- ## S
- sample audit report program 219
  - samples
    - checkpoint-level recovery response file
      - OUTMSG example 60
      - OUTWORK example 60
      - SESSIONEND (not 00000 or 28010) 60
    - command syntax 71, 72
    - INMSG 15
    - INPRO 13
    - OUTMSG 15
      - SESSIONEND(00000) 51
      - SESSIONEND(28010) 52
    - OUTPRO 14
  - SEND command 129
  - SENDEDI command 135
    - blanks following EDI segments 31
    - EDI envelopes 30
    - locating EDI destinations 31
    - overview 29
    - resolving the EDI destination 32
    - response records 41
  - sending EDI data
    - blanks following EDI segments 31
    - centralized Information Exchange alias
      - tables 36
    - EDI destination table entry format 44
    - EDI destination tables 35
    - EDI distribution lists 39
    - EDI envelopes 30
    - EDI qualifier table entry format 43
    - EDI qualifier tables 33
    - general discussion 31
    - resolving the EDI destination 32
    - SENDEDI command overview 29
    - specifying Information Exchange control fields
      - MSGNAME 40

- MSGSEQNO 40
  - MSGUCLS 41
  - SENDEDI response records 41
    - using a ZZ qualifier 37
  - SENDSTREAM command 140
  - SENT record 179
    - format 179
  - SENT records 50
  - session 313
    - See also* session work file (SESSION)
  - session work file (SESSION)
    - creating 57
  - SESSIONEND (not 00000 or 28010) example 60
  - SESSIONEND record 181
    - format 181
  - SESSIONEND(00000) example 51
  - SESSIONEND(28010) example 51
  - SESSIONINFO command 144
  - session-level recovery
    - processing the message response file
      - checking command RECEIVED records 50
      - checking command SENT records 50
      - checking return codes 49
      - INMSG examples 51
      - OUTMSG examples 51
      - response record format 48
  - SESSIONRESP record 182
    - format 182
  - SNACOMM command 77
  - SSL command 78
  - SSL implementation 8, 206
  - SSL parameters 74, 145
  - START command 145
  - successful completion 226
  - syntax
    - API commands 4
    - command 71
    - limitations 72
- T**
- tables
    - centralized alias 31
    - centralized Information Exchange alias 36
    - EDI destination 35
    - EDI qualifier 33
      - using a ZZ qualifier 37
  - TCPCOMM command 79
  - terminology conventions x
  - TESTMSG command 147
  - text data
    - receiving from a host operating system 25
    - sending to a host operating system 25
  - TRACE command 80
  - trace message text (TRACEMSG)
    - description 18
  - TRACEMSG
    - See* trace message text (TRACEMSG)
  - transferring files
    - introduction 23
    - to ASCII systems 24
    - to older Information Exchange interfaces
      - common data header unsupported 27
      - considerations 27
  - TRANSMIT command 82
    - autostart option 12
  - type conventions ix
- U**
- UCS
    - destination tables, bypassing 38
  - unexpected errors 296
  - user-initiated recovery 56, 93
  - using a ZZ qualifier 37
- V**
- verify parameter 68
- W**
- WARNING record 152
    - format 152, 186
  - WARNING record (OUTMSG response) 186
  - work files
    - considerations 56
    - creating
      - output work file (OUTWORK) 58
      - receive work file (RCVWORK) 57
      - session file (SESSION) 57
- Z**
- ZZ qualifier 30, 37

