



GXS

Information Exchange Administration Mailbox Command Reference

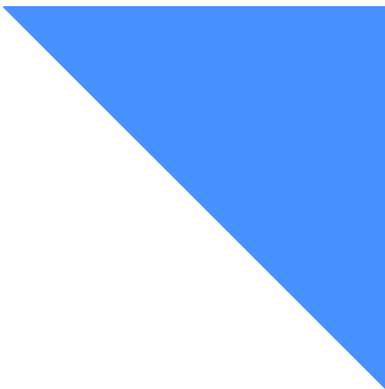
Version 4 Release 6

Fourth Edition (November 2005)

This edition replaces document number GC34-2260-02.

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

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



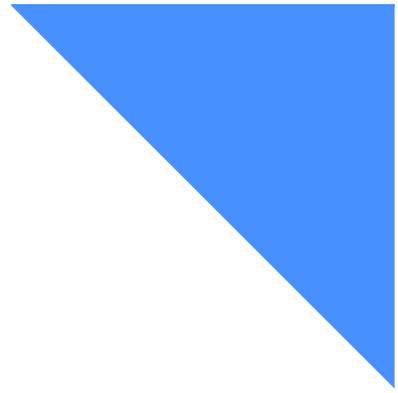
Contents

- To the reader v
 - Who should read this book v
 - How to use this book v
 - Related books v

- Chapter 1. Understanding the Information Exchange administration mailbox 1
 - Administration mailbox commands 1
 - Using the mailbox commands 3
 - Creating messages 3
 - System messages 3
 - Command syntax 5
 - Command names and keywords 6
 - Command examples 6
 - Message responses 7
 - Understanding the message response format 8
 - Response header 9
 - Individual response message entries 10
 - Command image 10
 - Example of submitted commands 11
 - Example of command response 12

- Chapter 2. Using the ALIAS FUNCTION(GET) command 13
 - Command syntax 13
 - Parameters and subparameters 14
 - Response to ALIAS FUNCTION(GET) command 18
 - Expedite Base response format 19
 - Stream response format 20
 - ALIAS FUNCTION(GET) stream data format 20

Chapter 3. Using the LIST FUNCTION(GET) command	21
Command syntax	21
Parameters and subparameters	22
Response to LIST FUNCTION(GET) command	26
Expedite Base response format	27
Stream response format	28
LIST FUNCTION(GET) stream data format	28
Chapter 4. Using the TPARTNER command	29
Command syntax when adding	29
Command syntax when changing	30
Command syntax when deleting	30
Command syntax when erasing	30
Parameters and subparameters	31
Chapter 5. Using the USER command	37
Command syntax	37
Parameters and subparameters	39
Chapter 6. Understanding the administration mailbox messages	47
Severity codes	47
Messages	48
Appendix A. TIMEZONE values	63
Appendix B. Panel values versus command syntax	65
Alternate administrator authority	67



To the reader

This book describes the Information Exchange administration mailbox facility. It describes how to use the administration mailbox commands and keywords with the SEND command. It also explains the command syntax and provides command examples.

Who should read this book

This book is written for users with an interface such as one of the Expedite products or any other interface that allows them to send messages or non-EDI files to Information Exchange.

How to use this book

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

Commands and parameters are displayed in this book in uppercase letters; for example, ACCOUNT. In the examples and command syntax, these appear in lowercase. However, you can type commands and parameters in uppercase or lowercase letters. The following conventions are used in command syntax examples:

- Required parameters and values are in boldface type.
- Parameter variables and values are italicized.
- Default values are underlined.
- Alternative constructions are listed within braces.
- The word *blank* (when listed as a variable) refers to a blank space.

Blank lines have been added to some examples to help you use them. Therefore, some examples in the book may not look exactly like what you see when you use your interface.

Related books

The following books contain information related to the topics covered in this book. These documents can be viewed on the GXS Website: <http://www.gxs.com>.

Expedite/Async

- Using Expedite/Async, GC34-2317
- Expedite Base
- Expedite Base/AIX for RISC System/6000 Programming Guide, GC34-3280
- Expedite Base/MVS Programming Guide, GC34-2204
- Expedite Base for Windows Programming Guide, GC34-2253
- Expedite Base/400 Programming Guide, GC34-2254

Expedite/CICS

- Customizing and Developing Applications with Expedite/CICS, GC34-3304
- Using the Expedite/CICS Display Application, GC34-3303

Expedite/Direct

- Using Expedite/Direct, GC34-2214
- Expedite for Windows
- Expedite for Windows Software Development Kit Programming Guide, GC34-3285
- Expedite for Windows Software Development Kit Programming Reference, GC34-3284
- Expedite for Windows User's Guide, GC34-2341

Information Exchange

- Information Exchange Administration Mailbox Command Reference, GC34-2260
- Information Exchange Administration Services Messages and Codes, GC34-2323
- Information Exchange Administration Services User's Guide, GC34-2221
- Information Exchange Administration Supplement, GN10-4073
- Information Exchange Charges Reference, GX66-0653
- Information Exchange Messages and Formats, GC34-2324
- Information Exchange via TCP/IP FTP Gateway Quick Reference, GX66-0664
- Information Exchange via TCP/IP FTP Gateway User's Guide, GC34-2345

Understanding the Information Exchange administration mailbox

With the administration mailbox function, Information Exchange users can perform Information Exchange Administration Services functions without using the Information Exchange Administration Services panels. It provides an asynchronous interface to services currently provided by Information Exchange Administration Services. For more information on specific fields and values, see the *Information Exchange Administration Services User's Guide*.

The administration mailbox is account *SYSTEM* user ID *ADMIN***. You can send various commands to this mailbox by placing the commands in a text file and sending the file in the same way you send a text message.



NOTE: Administration mailbox commands are not supported by Expedite/PC products prior to Version 3.0.

Administration mailbox commands

The administration mailbox commands include:

- **ALIAS FUNCTION(GET)**

Use this command to retrieve an alias table into a user's mailbox. The ALIAS FUNCTION(GET) creates messages that are sent from account *SYSTEM* and user ID ADMINGET.

For detailed information on the command syntax, see "Using the ALIAS FUNCTION(GET) command" on page 13.

■ **LIST FUNCTION(GET)**

Use this command to retrieve a distribution list into a user's mailbox. The LIST FUNCTION(GET) creates messages that are sent from account *SYSTEM* and user ID ADMINGET.

For detailed information on the command syntax, see "Using the LIST FUNCTION(GET) command" on page 21.

■ **TPARTNER**

Use this command to work with trading partner list data.

For detailed information on the command syntax, see "Using the TPARTNER command" on page 29.

■ **USER**

Use this command to work with user profile data.

For detailed information on the command syntax, see "Using the USER command" on page 37.

Using the mailbox commands

When you send a message using the Information Exchange administration mailbox commands, it is sent to Information Exchange like any other message, except that it is addressed to account *SYSTEM* and user ID *ADMIN***. The following sections provide more detailed information on how to use the Information Exchange administration mailbox commands.

Creating messages

The text of the messages consists of one or more commands. You can format the commands with a text editor of your choice and send a file of commands to Information Exchange by using an interface such as one of the Expedite products, or any other interface that enables you to send messages or non-EDI files to Information Exchange.

The message charge mode field of the SEND MESSAGE command must be one of the valid values (1 - 6). There is no charge for messages sent to or from account *SYSTEM* and user ID *ADMIN***. However, you will incur receive-side charges once you receive a message from account *SYSTEM* and user ID ADMINGET.

For additional information on the format of messages received from account *SYSTEM* and user ID *ADMIN***, see “Understanding the message response format” on page 8.

The use of fields in the message header to identify message class, name, and input fields, as well as common data header (CDH) fields, is unnecessary because all pertinent information is included in the text of the message.

System messages

In addition to *SYSTEM* *ADMIN** messages, Information Exchange also generates other messages and sends them to the user’s mailbox. These include information messages, such as acknowledgments, and error messages. The format of these messages is similar to the format of those sent from other users.

The message in the mailbox displays the sender as account `*SYSTEM*`, and the user ID as follows:

<code>ADMINGET</code>	Identifies this message as output from GET requests to the administration mailbox.
<code>*AUDITS*</code>	Identifies this message as containing audit information.
<code>*ERRMSG*</code>	Identifies this message as containing either error messages or message acknowledgments.
<code>*EVENTS*</code>	Identifies this message as coming from Information Exchange's session trace function.
<code>*LIBRARY</code>	Identifies this message as coming from Information Exchange's library function.
<code>*LSTRSP*</code>	Identifies this message as containing the output from the LIST VERIFY command.
<code>*PRBRSP*</code>	Identifies this message as containing the output from the ASYNCHRONOUS PROBE command.
<code>*TSTMSG*</code>	Identifies this message as containing test messages the user requested from Information Exchange.

The following account `*SYSTEM*` user IDs are found *only* on a user's invoice. They are generated when the user browses the respective areas using Information Exchange Administration Services or performs the designated function.

<code>*ADTBRS*</code>	Generated when browsing audits.
<code>LIBBRWS*</code>	Generated when browsing library members through Information Exchange Administration Services.
<code>*LIBBRWS</code>	Generated when browsing library members through the Browse Library Members command.
<code>*LIBSTOR</code>	Generated when storing members in a library.
<code>*MSGBRS*</code>	Generated when viewing selected messages in the mailbox.

For additional information on account `*SYSTEM*` messages, see the *Information Exchange Interface Programming Guide*.

Command syntax

The Information Exchange administration mailbox command syntax is as follows:

```
command keyword(value) keyword(value) ...keyword(value);
```

where:

command	Specifies the name of the command.
keyword	Identifies the name of a keyword and the value associated with it (the value is enclosed in parentheses).
value	Defines the value associated with the keyword. A keyword value can either be a simple field value or a sublist of parameters, each separated by a blank.
...	Indicates you can specify as many parameters as necessary. This is not part of the syntax.
;	Ends the command.

A parameter identifies a keyword and its associated value.

When creating commands, it is important to remember the following about the command syntax:

- A command can span multiple lines.
- A command name, keyword name, or value cannot span multiple lines.
- One or more blanks must occur after a parameter.
- One or more blanks can occur after any right parenthesis.
- No blanks can occur within a value enclosed by parentheses.

For example, `userid(mimi)` is valid, but `userid(mi mi)` is not. If you need to include blanks in the string, you must use single quotes around the value string, such as `userid('mi mi')`.

- You can specify comments within the command.
Any values inside `/* */` delimiters are ignored when the command is processed.
- A semicolon (;) at the end of a command ends the command and acts as a separator between multiple commands.

Command names and keywords

The command names are intended to be associated with the objects that are being acted upon. The keyword `FUNCTION` is used to specify actions that can be performed on the objects. These actions include add, change, delete, erase, and get.

The keyword names are intended to be descriptive and easy to remember. Underscores are used to separate long keywords, especially if they consist of two separate words. Because of the length of some keywords, synonyms are available that consist of eight characters or less.

The keyword `REQUESTID` is used to cross-reference a request sent to the administration mailbox with the corresponding response received from the administration mailbox. This is an optional keyword that defaults to blanks.



NOTE: When a command error is so severe that it cannot be parsed, the `REQUESTID` cannot be retrieved for the error message and will be blank. These types of errors typically include invalid value lengths, invalid keywords, mutually exclusive parameters, and so on.

The keyword `POSITIVE_ACK` is used to specify if a positive acknowledgment response is requested. This is an optional keyword and the default is `Y`. For more information, see “Message responses” on page 7.

Command examples

This section gives some examples of how you might use the Information Exchange administration mailbox commands.

The following is an example of how to change one field in a user’s profile. In this example, the time zone field in the user’s profile is being changed to `est`. The submitter of the command has specified `positive_ack(y)` to request an acknowledgment message if the command processes with no warnings or terminating errors. The submitter also has specified a `requestid` (in this case, `12345678`) that can be correlated with the acknowledgment message received.

```
/* Set the default time zone to EST */  
  
user function(change) requestid(12345678) positive_ack(y)  
    account(aacct) userid(auser)  
    timezone(est);
```

The following is an example of how to change multiple fields in a user's profile with one command by using subparameters within parentheses. In this example, the reset session and force archive fields in the user's profile are being changed. The submitter of the command has specified *reset_session(y)* to allow the user to reset the user's own session. The submitter also has specified *force_archive(n)* to indicate that the user's mail will not automatically be stored in the archive. Both RESET_SESSION and FORCE_ARCHIVE are subparameters of the PROFILEINFO parameter, so they are specified within parentheses following PROFILEINFO.

```
user function(change)
  account(aacct) userid(auser)
  timezone(est)
  profileinfo(reset_session(y) force_archive(n));
```

For further information on using the administration mailbox commands, see the following:

- “Using the ALIAS FUNCTION(GET) command” on page 13.
- “Using the LIST FUNCTION(GET) command” on page 21.
- “Using the TPARTNER command” on page 29.
- “Using the USER command” on page 37.

Message responses

If you send multiple commands within the same message, when Information Exchange processes the message, it separates the commands so that each one is processed independently.

Information Exchange sends a response to the submitter's Information Exchange mailbox for each individual command it receives. These responses include error, warning, or informational messages. If there are no errors or warnings, Information Exchange sends a positive acknowledgment (this is the default) to the submitter's Information Exchange mailbox upon successful completion of the command.

The message user classes for these responses are as follows:

ERROR	The response contains terminating errors.
WARNING	The response contains warnings, plus a successful completion message.
POS_ACK	The response contains no warnings or terminating errors.



NOTE: If you specify N for POS_ACK, you will not receive any positive acknowledgments.

If you include two commands in the same message and do not use a semicolon to end the first command, Information Exchange processes the two commands as one. Even if there are no other errors, because the semicolon is missing, Information Exchange generates an error response stating that the command has invalid keywords. If you send this same message with a semicolon separator, Information Exchange processes each command separately and sends a positive acknowledgment to your mailbox upon successful completion of each command.



NOTE: Because the commands are processed asynchronously, the order in which Information Exchange processes the commands cannot be guaranteed. To ensure successful completion of a command before you submit a second command, you must wait for a response for the previous command. This is especially important if the commands you submit refer to the same object.

Understanding the message response format

The message response format is designed in a readable format, but the data is column-dependent so it can be processed by a program interface, if desired. The response is in a 77-character, carriage-return line-feed (CRLF) format with a CDH.

Each message response contains:

- A header line
- One or more message entry lines
- The image of the command that generated the message

A response is created to inform the user of terminating errors and warnings, and to provide the user with positive acknowledgment for commands submitted to Information Exchange using the administration mailbox facility.

The sender account and user ID is ***SYSTEM**ADMIN****. The message user class is **ERROR** for those messages containing terminating errors; **WARNING** for those messages containing warnings, but not terminating errors; and **POS_ACK** for those messages containing no warnings or errors.

Response header

The first line of each message placed in the user's mailbox is a header in the format shown in the following table. For examples, see "Example of submitted commands" on page 11, and "Example of command response" on page 12.

NAME	COLUMN	SIZE	DESCRIPTION
RECTYPE	1	1	* identifies the header line.
	2	7	The actual value is System:
SYSID	9	3	The system ID that issued the error message.
	12	9	The actual value is Errors:
ECOUNT	21	3	The number of errors detected.
	24	11	The actual value is Warnings:
WCOUNT	35	3	The number of warnings detected.
	38	5	The actual value is At:
TIMESTAMP	43	19	The time the command message was received by Information Exchange. The format is YYYY/MM/DD-HH:MM:SS in the user's default date and time.
	62	8	The actual value is ReqID:
REQUESTID	70	8	The request ID specified by the submitter. This is a free-format field provided for users to coordinate their requests with the responses they receive. The response has the REQUESTID as the message name. The default is blank.

Individual response message entries

There is a message entry for each error, warning, or informational condition detected. Each entry begins on a new line.

NAME	COLUMN	SIZE	DESCRIPTION
RECTYPE	1	1	A minus sign identifies a message entry line. This column contains a plus sign for continuation lines.
ENUMBER	2	5	The unique number identifying the message. This column is blank for continuation lines.
SEVERITY	7	1	The severity of the error. <i>I - Informational, W - Warning, or E - Error</i> . This column is blank for continuation lines.
	8	1	A colon identifies the first line of each message entry. This column is blank for continuation lines.
	9	1	Blank filler
TEXT	10	68	The text explaining the warning or error condition. If the text is longer than can be placed on a single line, the message is broken at the last word and continued on the next line starting with column 10. The RECTYPE of the continued message line is a plus sign.

Command image

The image of the command that generated the messages is the third part of the response.

The first column (RECTYPE) of the first line of the command image contains a period. The first column (RECTYPE) of every line that follows contains a plus sign to indicate continuation.

Example of submitted commands

The following is a hypothetical example of commands submitted to ***SYSTEM**ADMIN**** that illustrates the messaging format.

The user has submitted two commands in a single message.

```
/* ***** */
/* Change altadmin1 authority and profileinfo */
/* fields for USER1 in account ACCT1 */
/* ***** */
    user function(change)
    account(acct1) userid(user1)
    requestid(userm23)
    altadmin1(account(userm24) authority(wrongone))
    profileinfo(receiving(blocked) archive_days(512));
/* ***** */
/* Change altadmin1 authority and profileinfo */
/* fields for USER2 in account ACCT2 */
/* ***** */
    user function(change)
    account(acct2) userid(user2)
    requestid(userm24)
    altadmin1(account(userm23) authority(mailonly))
    profileinfo(receiving(blocked) archive_days(045));
```

Example of command response

The user receives two responses from ***SYSTEM**ADMIN****, one for each command in the administrative services request message. One is an error, the other a positive acknowledgment.

■ **User class: 'ERROR ', MsgName: 'USERM23'**

```
*System:USA Errors:001 Warnings:001 At:1999/09/30-2:27:03 ReqID:USERM23
-32203W: ARCHIVE_DAYS(512) is an invalid value, the default of 001
+      was taken.
-32150E: AUTHORITY(WRONGONE) is an invalid value.
-32301I: Request was not processed because of errors.
. USER FUNCTION(CHANGE) ACCOUNT(ACCT1) USERID(USER1)
+      REQUESTID(USERM23) ALTADMIN1(ACCOUNT(USERM24)
+      AUTHORITY(WRONGONE)) PROFILEINFO(RECEIVING(BLOCKED)
+      ARCHIVE_DAYS(512));
```

■ **User class: 'POSACK ', Msgname: 'USERM24'**

```
*System:USA Errors:000 Warnings:000 At:1999/09/30-12:27:04 ReqID:USERM24
-32300I: Request was processed successfully.
. USER FUNCTION(CHANGE) ACCOUNT(ACCT2) USERID(USER2)
+      REQUESTID(USERM24) ALTADMIN1(ACCOUNT(USERM23)
+      AUTHORITY(MAILONLY)) PROFILEINFO(RECEIVING(BLOCKED)
+      ARCHIVE_DAYS(045));
```

Using the ALIAS FUNCTION(GET) command

The ALIAS FUNCTION(GET) command is issued to the administration mailbox to obtain a copy of an alias table. The originator of the response that you receive from the administration mailbox is account *SYSTEM* and user ID ADMINGET. The FUNCTION performed is a required parameter, as is ALIASTABLE.

Command syntax

The ALIAS FUNCTION(GET) command is as follows:

```
alias function(get)
  aliastable(tbl)
  requestid(blank | requestid)
  positive_ack(y | n)
  owner(account(acct) userid(user))
  output_format(h | s)
  retain(days)
  msg_name(name)
  msg_sequence(seqn)
  delivery_class(blank | p)
  destination({account(account) userid(userid)} |
    {aliastable(tbl) aliasname(alias)});
```

Parameters and subparameters

The following table describes the ALIAS FUNCTION(GET) parameters and subparameters. Subparameters follow their related parameter and are italicized.

NAME	REQ'D	SIZE	DESCRIPTION
FUNCTION	Yes	N/A	The function that is to be performed on the alias table. GET is the only valid function at this time.
ALIASTABLE	Yes	4	The alias table identifier. The first character specifies the alias table type. The valid values are: G, O, and P. <ul style="list-style-type: none"> • G indicates a global table. • O indicates an organizational table. • P indicates a private table. The next 3 characters are the alias table name. (Synonym: ALIASTBL).
REQUESTID	No	8	The submitter identifier for the command. The REQUESTID can be correlated with the acknowledgment message received. The default is blank . (Synonym: REQID).
POSITIVE_ACK	No	1	This indicates whether the user wants a positive acknowledgment - Y or N. If this parameter is not specified, the default is Y. The user always receives negative acknowledgments. (Synonym: POS_ACK or POSACK).

NAME	REQ'D	SIZE	DESCRIPTION
OWNER	No	N/A	The owner of the alias table. If the OWNER parameter is specified, an owner ACCOUNT must be defined within. The default is the submitter ACCOUNT and USERID.
<i>ACCOUNT</i>	No	8	The account ID associated with the alias table. Not required for G-type tables, or if the owner ACCOUNT is the same as the submitter account for O-type tables, or if the owner ACCOUNT and USERID are the same as the submitter ACCOUNT and USERID for P-type tables.
<i>USERID</i>	No	8	The user ID associated with the alias table. Not required for G-type tables, or for O-type tables unless the owner ACCOUNT is different from the submitter ACCOUNT. Required for P-type tables if the OWNER ACCOUNT and USERID are different from the submitter ACCOUNT and USERID. A cluster child cannot be the owner; if you specify a cluster child ID, the system defaults to the cluster parent ID.
OUTPUT_FORMAT	No	1	If OUTPUT_FORMAT is an S, the alias table will be retrieved in stream format (the input format for the Information Exchange Define Alias command, SDIDALS, starting with the number of alias entries (items) field). If OUTPUT_FORMAT is a B, the alias table will be retrieved in Expedite Base format (the input format for the DEFINEALIAS command). The default is B. (Synonym: OUTPFMAT).
RETAIN	No	3	The retention period for the retrieved alias table message. If not specified, the system default is used.
MSG_NAME	No	8	The name of the message in the destination mailbox. (Synonym: MSGNAME).

NAME	REQ'D	SIZE	DESCRIPTION
MSG_SEQUENCE	No	5	The sequence number associated with the message. This may be nonnumeric. (Synonym: MSGSEQNO).
DELIVERY_CLASS	No	1	The class of delivery service for this file. May be blank, or P for high priority. The default is blank. (Synonym: DELCLASS).
DESTINATION	No	N/A	The destination where the retrieved data is to be sent. If the DESTINATION parameter is specified, either: <ul style="list-style-type: none"> • A destination ACCOUNT and USERID, OR • A destination ALIASTABLE and ALIASNAME must be specified within. You cannot specify a cluster parent ID or an alias for a cluster parent ID as a destination. The default is the submitter ACCOUNT and USERID . (Synonym: DEST).
<i>ACCOUNT</i>	No	8	The account ID of the destination. When the destination USERID is specified, the ACCOUNT must also be specified. No other destination parameters are appropriate.
<i>USERID</i>	No	8	The user ID of the destination. When the destination ACCOUNT is specified, the USERID must also be specified. No other destination parameters are appropriate.

NAME	REQ'D	SIZE	DESCRIPTION
<i>ALIASTABLE</i>	No	4	<p>The alias table identifier of the destination. The first character specifies the Alias Table Type. The valid values are: G, O, or P.</p> <ul style="list-style-type: none">• G indicates a global table.• O indicates an organizational table.• P indicates a private table. <p>The next three characters are the alias table name.</p> <p>When a destination ALIASTABLE is specified, ALIASNAME must also be specified. No other destination parameters are appropriate.</p> <p>(Synonym: ALIASTBL).</p>
<i>ALIASNAME</i>	No	16	<p>The alias name of the destination. The alias name is defined within the alias table. When ALIASNAME is specified, a destination ALIASTABLE must also be specified. No other destination parameters are appropriate.</p> <p>(Synonym: ALIASNM).</p>

Response to *ALIAS FUNCTION(GET)* command

The successful execution of the *ALIAS FUNCTION(GET)* command results in a message containing the entries of an alias table being placed in the destination mailbox.

The following explains some of the fields returned in the message:

- User message class

The user message class (assigned by Information Exchange) contains the alias table name the user chooses to retrieve, followed by a sequence number. The sequence number begins with 001. If there are more than 9,999 entries in the alias table, a second message will be placed in the mailbox with the sequence number incremented by 1 (002 and so on).

- Message name

The message name is assigned as an identifier by the user when the user chooses to retrieve an alias table.

- Message sequence

The message sequence is assigned by the user as a control number for a message.



NOTE: You can only retrieve aliases into your own mailbox or into the mailboxes of users for whom you are an administrator. When receiving the message from a mailbox, the receiver will be billed for receive-side charges.

The format of the response will be in either Expedite Base format or “stream” format, depending on the *OUTPUT_FORMAT* parameter.

Expedite Base response format

The Expedite Base format will be the same as the input format for the Define Alias command. For more information and an explanation of the fields, refer to an Expedite Base product programming guide.

The following is an example of the Expedite Base response format.

When specifying the following command:

```
alias function(get) aliastable(glll) output_format(b);
```

the following is returned in the mailbox, from *SYSTEM* ADMINGET, with the user message class being G-LLL001:

```
#####
DEFINEALIAS  FUNCTION(E)                                #-ERASE
OLD VERSION OF ALIAS TABLE
      ALIASTABLE(GLLL);
#####
DEFINEALIAS  FUNCTION(N)                                #-
CREATE NEW VERSION OF ALIAS TABLE
      ALIASTABLE(GLLL) AUTHORITY(P)
#####
DEFINENAME(USER1      ) ALIAS(GMMM) ALIASNAME(NAME1      )
DEFINENAME(USER2      ) SYSID(SYS)  ACCOUNT(ACCT        ) USERID(AABBCC )
DEFINENAME(USER3      )              ACCOUNT(ACCT        ) USERID(AABBCC )
#####
;
```

The CDH will define the lines as 77 characters each, to be followed by a CRLF.

Stream response format

The stream format will be the format needed for the Information Exchange DEFINEALIAS command, SDIDALS, starting with the number of alias entries (items) field. For more information, see the *Information Exchange Interface Programming Guide*.

The following is an example of the stream response format.

When specifying the following command:

```
alias function(get) aliastable(glll) output_format(s);
```

the following is returned in the mailbox, from *SYSTEM* ADMINGET, with the user message class being (G-LLL001):

```
0003USER1   GMMMNAME1   USER2   ISYSACCT   AABBCC   USE
R3         ACCT     AABBCC
```

ALIAS FUNCTION(GET) stream data format

#	COL	SIZE	NAME	DESCRIPTION
1	1	4	ENTRIES	Number of the following entries.
2	5	36n*		Beginning of the first alias entry.
		16	ALIAS	Alias name.
		1	TBLTYPE	Alias table type (G, O, P), or intersystem indicator (I), or blank (for a local account and user ID).
		3	TBLNAME	Alias table name (when TBLTYPE is G, O, or P), intersystem system ID (when TBLTYPE is I), or blank (when TBLTYPE is blank).
		16	USER	Alias name reference or account and user ID.
* n represents the number of entries.				

A CDH is not generated for the stream format.

Using the LIST FUNCTION(GET) command

The LIST FUNCTION(GET) command is issued to the administration mailbox to obtain a copy of a distribution list. The originator of the response you receive from the administration mailbox is account *SYSTEM* and user ID ADMINGET. The function performed is a required parameter, as are LISTNAME and LISTTYPE.

Command syntax

The LIST FUNCTION(GET) command is as follows:

```
list function(get)
listname(list)
listtype(a | g | p)
requestid(blank)
positive_ack(y | n)
owner(account(acct) userid(user))
output_format(b | s)
retain(days)
msg_name(name)
msg_sequence(seqn)
delivery_class(blank | p)
destination( {account(account) userid(userid)} |
{aliastable(tbl) aliasname(alias)});
```

Parameters and subparameters

The following table describes the LIST FUNCTION(GET) parameters and subparameters. Subparameters follow their related parameter and are italicized.

NAME	REQ'D	SIZE	DESCRIPTION
FUNCTION	Yes	N/A	The function that is to be performed upon the distribution list. GET is the only valid function at this time.
LISTNAME	Yes	8	The name of the distribution list.
LISTTYPE	Yes	1	The type of distribution list. The valid values are: A, P, or G. <ul style="list-style-type: none"> • A indicates an account-level distribution list, which is available to every user in the account. • P indicates a private distribution list, which is available only to the user for whom the list was created. • G indicates an account-level group list.
REQUESTID	No	8	Specifies a submitter identifier for the command. The REQUESTID can be correlated with the acknowledgment message received. The default is blank . (Synonym: REQID).
POSITIVE_ACK	No	1	Indicates whether the user wants a positive acknowledgment - Y or N. If this parameter is not specified, the default is Y . The user will always receive negative acknowledgments. (Synonym: POS_ACK or POSACK).

NAME	REQ'D	SIZE	DESCRIPTION
OWNER	No	8	The owner of the distribution list. If the OWNER parameter is specified, an owner ACCOUNT must be defined within. The default is the submitter ACCOUNT and USERID .
<i>ACCOUNT</i>	No	8	The account ID associated with the distribution list.
<i>USERID</i>	No	8	The user ID associated with the distribution list. A cluster child cannot be the owner; if you specify a cluster child ID, the system defaults to the cluster parent ID.
OUTPUT_FORMAT	No	1	If OUTPUT_FORMAT is an S, the distribution list will be retrieved in stream format (the input format for the Information Exchange List Define command, SDILSTD, starting with the 'Destination ID entry' fields). If the OUTPUT_FORMAT is a B, the distribution list will be retrieved in Expedite Base format (the input format for the LIST command). The default is B . (Synonym: OUTPFMAT).
RETAIN	No	3	The retention period associated with the message containing the retrieved list. If not specified, the system default is used.
MSG_NAME	No	8	The name of the message in the destination mailbox. (Synonym: MSGNAME).
MSG_SEQUENCE	No	5	The sequence number associated with the message. This may be nonnumeric. (Synonym: MSGSEQNO).

NAME	REQ'D	SIZE	DESCRIPTION
DELIVERY_CLASS	No	1	The class of delivery service for this file. May be blank or P for high priority. The default is blank . (Synonym: DELCLASS).
DESTINATION	No	N/A	The destination where the retrieved data is to be sent. If the DESTINATION parameter is specified, either: <ul style="list-style-type: none"> • A destination ACCOUNT and USERID, OR • A destination ALIASTABLE and ALIASNAME must be specified within. The default is the submitter ACCOUNT and USERID. (Synonym: DEST).
ACCOUNT	No	8	The account ID of the destination. When the destination USERID is specified, the ACCOUNT must also be specified. No other destination parameters are appropriate.
USERID	No	8	The user ID of the destination. When the destination ACCOUNT is specified, the USERID must also be specified. No other destination parameters are appropriate.

NAME	REQ'D	SIZE	DESCRIPTION
<i>ALIASTABLE</i>	No	4	<p>The alias table identifier of the destination. The first character specifies the alias table type. The valid values are: G, O, or P.</p> <ul style="list-style-type: none"> • G indicates a global table. • O indicates an organizational table. • P indicates a private table. <p>The next three characters are the alias table name.</p> <p>When a destination <i>ALIASTABLE</i> is specified, an <i>ALIASNAME</i> must also be specified. No other destination parameters are appropriate.</p> <p>(Synonym: <i>ALIASTBL</i>).</p>
<i>ALIASNAME</i>	No	16	<p>The alias name of the destination. The alias name is defined within the alias table. When <i>ALIASNAME</i> is specified, a destination <i>ALIASTABLE</i> must also be specified. No other destination parameters are appropriate.</p> <p>(Synonym: <i>ALIASNM</i>).</p>

Response to LIST FUNCTION(GET) command

The successful execution of the LIST FUNCTION(GET) command results in a message containing the entries of a distribution list being placed in the destination mailbox.

The following is an explanation of some of the fields returned in the message:

- User message class

The user message class (assigned by Information Exchange) contains the name of the distribution list the user chooses to retrieve.

- Message name

The message name is assigned as an identifier by the user when the user chooses to retrieve a distribution list.

- Message sequence

The message sequence is assigned by the user as a control number for a message.



NOTE: You can only retrieve distribution lists into your own mailbox or into the mailboxes of users for whom you are an administrator. When receiving the message from a mailbox, the receiver is billed for receive-side charges.

The format of the response will be in either Expedite Base format or “stream” format, depending on the OUTPUT_FORMAT parameter.

Expedite Base response format

The Expedite base format will be in the form of an Expedite Base LIST command to create a new version of the distribution list. For more information and an explanation of the fields, refer to an Expedite Base product programming guide.

The following is an example of the Expedite Base response format.

When specifying the following command:

```
list function(get) listname(listlll) listtype(a) output_format(b);
```

the following is returned in the mailbox, from *SYSTEM* ADMINGET, with the user message class being the list name (LISTLLL):

```
#####
List Function(N)
#-Replace with New Copy
      ListName(LISTLLL )      ListType(A)
#      Owner(Account(ACCT ))
#####
      ACCOUNT(ACCT )      USERID(USER1 )
      ACCOUNT(ACCT )      USERID(USER2 )
      ACCOUNT(ACCT )      USERID(USER3 )
ALIAS(GLLL) ALIASNAME(NAME1 )
      ACCOUNT(ACCT )      USERID(USER4 )
      ACCOUNT(ACCT )      USERID(USER5 )
SYSID(SYS)  ACCOUNT(ACCT )      USERID(USER6 )
#####
;
```

The CDH will define the lines as 77 characters each, to be followed by a CRLF.

Stream response format

The stream format will be the same format as that used for the ‘Destination ID entry’ fields for the Information Exchange List Define command, SDILSTD. For more information, see the *Information Exchange Interface Programming Guide*.

The following is an example of the stream response format.

When specifying the following command:

```
list function(get) listname(listlll) listtype(a) output_format(s);
```

the following is returned in the mailbox, from *SYSTEM* ADMINGET, with the user message class being the listname:

```
ACCT USER1  ACCT USER2  ACCT  USER3  GLLNAME1
ACCT USER4  ACCT  USER5  ISYSACCT  USER6
```

LIST FUNCTION(GET) stream data format

#	COL	SIZE	NAME	DESCRIPTION
1	1	1	TBLTYPE	Alias table type (G, O, P), or intersystem indicator (I), or blank (for a local account and user ID).
2	2	3	TBLNAME	Alias table name (when TBLTYPE is G, O, or P), intersystem system ID (when TBLTYPE is I), or blank (when TBLTYPE is blank).
3	5	16	USER	This field is either the 16-character alias name of the desired destination, or the Information Exchange account (characters 5 through 12) and user ID (characters 13 through 20) of the desired destination.

A CDH is not generated for the stream format.

Using the TPARTNER command

The TPARTNER command is used to perform a function on the user's trading partner list. The functions available are ADD, CHANGE, DELETE (delete a single partner), and ERASE (delete the entire trading partner list). The FUNCTION performed is a required parameter, as are ACCOUNT and USERID. Other required parameters are FUNCTION dependent.

Any synonyms for the keywords of the command will be indicated in the chart of parameters following the keyword name.

Command syntax when adding

The command when *adding* is as follows:

```
tpartner function(add)
account(account) userid(userid)
positive_ack(y | n)
requestid(blank | requestid)
partner({aliastable(ttbl) aliasname(alias)}) |
  {sysid(your sysid | sys) account(account) userid(userid)})
sending(nopay | blocked | paysend | payboth)
receiving(nopay | blocked | payreceive | payboth)
ccpartner(ccnopay | ccprocess | ccreceive | ccboth)
ccrecipient(ccnopay | ccprocess | ccreceive | ccboth)
eso(z | y);
```

Command syntax when changing

The command when *changing* is as follows:

```
tpartner function(change)  
  
account(account) userid(userid)  
positive_ack(y | n)  
requestid(blank | requestid)  
partner({aliastable(tbl) aliasname(alias)} |  
        {sysid(your_sysid | sys) account(account) userid(userid)})  
sending(nopay | blocked | paysend | payboth)  
receiving(nopay | blocked | payreceive | payboth)  
ccpartner(ccnopay | ccprocess | ccreceive | ccboth)  
ccrecipient(ccnopay | ccprocess | ccreceive | ccboth)  
eso(n | y);
```

Command syntax when deleting

The command when *deleting* is as follows:

```
tpartner function(delete)  
  
account(account) userid(userid)  
positive_ack(y | n)  
requestid(blank | requestid)  
partner({aliastable(tbl) aliasname(alias)} |  
        {sysid(your_sysid | sys) account(account) userid(userid)});
```

Command syntax when erasing

The command when *erasing* is as follows:

```
tpartner function(erase)  
  
account(account) userid(userid)  
positive_ack(y | n)  
requestid(blank | requestid);
```

Parameters and subparameters

The following table describes the TPARTNER parameters and subparameters. Subparameters follow their related parameter and are italicized.

NAME	REQ'D	SIZE	DESCRIPTION
FUNCTION	Yes	6	The function to be performed on a user's trading partner list - ADD, CHANGE, DELETE, or ERASE.
ACCOUNT	Yes	8	The account ID of the user whose trading partner list is updated.
USERID	Yes	8	The user ID of the user whose trading partner list is updated. A cluster child ID cannot be the owner of a trading partner list. If you specify a cluster child ID, the system defaults to the cluster parent ID.
POSITIVE_ACK	No	1	This indicates whether the user wants a positive acknowledgment - Y or N. If this parameter is not specified, the default is Y. The user will always receive negative acknowledgments. (Synonym: POS_ACK or POSACK).
REQUESTID	No	8	The requester identifier for the command. The REQUESTID can be correlated with the acknowledgment message received. The default is blank. (Synonym: REQID).

NAME	REQ'D	SIZE	DESCRIPTION
PARTNER	Yes(1)	N/A	PARTNER is a field with subparameters. PARTNER may not be specified as PARTNER().
<i>ALIASTABLE</i>	Yes(2)	4	The alias table identifier of the trading partner's alias. The first character specifies the alias table type. The valid values are: G, O, or P. <ul style="list-style-type: none"> • G indicates a global table. • O indicates an organizational table. • P indicates a personal table. The next three characters are the alias table name. (Synonym: ALIASTBL).
<i>ALIASNAME</i>	Yes(2)	16	The alias name of the trading partner. (Synonym: ALIASNM).
<i>SYSID</i>	No	3	The system ID of the trading partner specified by partner ACCOUNT and partner USERID if the trading partner is on another system.
<i>ACCOUNT</i>	Yes(3)	8	The account ID of the trading partner.
<i>USERID</i>	Yes(3)	8	The user ID of the trading partner.
<p>(1) This parameter is not allowed for FUNCTION(ERASE).</p> <p>(2) This parameter is required when the partner ACCOUNT and partner USERID parameters are omitted. Otherwise, it is not permitted.</p> <p>(3) This parameter is required when the partner ALIASTABLE and partner ALIASNAME parameters are omitted. Otherwise, it is not permitted.</p>			

NAME	REQ'D	SIZE	DESCRIPTION
SENDING	No(4)	7	<p>This indicates the user's payment agreement when sending to this trading partner. The valid values are: NOPAY, BLOCKED, PAYSSEND, and PAYBOTH.</p> <ul style="list-style-type: none"> • NOPAY indicates that no payments are authorized. • BLOCKED indicates that the user is not able to send messages to this partner. • PAYSSEND indicates that only the send-side charges are authorized for payment. • PAYBOTH indicates that both send-side and receive-side charges are authorized for payment.
RECEIVING	No(4)	10	<p>This indicates the user's payment agreement when receiving from this trading partner. The valid values are: NOPAY, BLOCKED, PAYRECEIVE, and PAYBOTH.</p> <ul style="list-style-type: none"> • NOPAY indicates that no payments are authorized. • BLOCKED indicates that the user is not able to receive messages from this partner. • PAYRECEIVE indicates that only the receive-side charges are authorized for payment. • PAYBOTH indicates that both send-side and receive-side charges are authorized for payment. <p>(Synonym: RECVING).</p>
(4) This parameter is required for FUNCTION(ADD).			

NAME	REQ'D	SIZE	DESCRIPTION
CCPARTNER	No(4)	9	<p>This indicates the owner of the trading partner list's payment agreement for carbon copy charges when the owner of the list is the partner in a carbon copy relationship. This is specified for each trading partner in the list and will be used when that trading partner is the requester of a carbon copy and specifies "P" (partner) in either or both the "Who pays for processing" field or the "Who pays for receiving" field of the carbon copy relationship. The valid values are: CCNOPAY, CCPROCESS, CCRECEIVE, and CCBOTH.</p> <ul style="list-style-type: none"> • CCNOPAY indicates that no carbon copy payments are authorized. • CCPROCESS indicates that the user agrees to pay only the carbon copy processing charge. • CCRECEIVE indicates that the user agrees to pay only the carbon copy receiving charge. • CCBOTH indicates that the user agrees to pay both the carbon copy processing charge and the carbon copy receiving charge. <p>(Synonym: CCPARTN).</p>
(4) This parameter is required for FUNCTION(ADD).			

NAME	REQ'D	SIZE	DESCRIPTION
CCRECIPIENT	No(4)	9	<p>This indicates the owner of the trading partner list's payment agreement for carbon copy charges when the owner of the list is the recipient in a carbon copy relationship. This is specified for each trading partner in the list and will be used when that trading partner is the requester of a carbon copy and specifies "R" (recipient) in either or both the "Who pays for processing" field or "Who pays for receiving" field of the carbon copy relationship. The valid values are: CCNOPAY, CCPROCESS, CCRECEIVE, and CCBOTH.</p> <ul style="list-style-type: none"> • CCNOPAY indicates that no carbon copy payments are authorized. • CCPROCESS indicates that the user agrees to pay only the carbon copy processing charge. • CCRECEIVE indicates that the user agrees to pay only the carbon copy receiving charge. • CCBOTH indicates that the user agrees to pay both the carbon copy processing charge and the carbon copy receiving charge. <p>(Synonym: CCRECIP).</p>
ESO	No	1	<p>This indicates whether mail will be exchanged only if both users are Extended Security Option (ESO) users - Y or N.</p> <p>When adding, the default is N.</p>
(4) This parameter is required for FUNCTION(ADD).			

Using the USER command

The USER command is used to perform a function on a user's profile record. The FUNCTION performed is a required parameter. Currently, the only available function is CHANGE. Other required parameters are ACCOUNT and USERID. For the CHANGE function, the other parameters are optional; if not specified, the affected profile fields are not altered.



NOTE: A registration order must be placed to set up or modify the cluster fields in a cluster user ID.

Command syntax

The command is as follows:

```

user function(change)

account(account) userid(userid)
positive_ack(y | n)
requestid(blank | requestid)
altadmin1(account(account) userid(userid)
  authority(all | mailonly | nonmail | sysmail) )
altadmin2(account(account) userid(userid)
  authority(all | mailonly | nonmail | sysmail) )
altadmin3(account(account) userid(userid)
  authority(all | mailonly | nonmail | sysmail) )
altadmin4(account(account) userid(userid)
  authority(all | mailonly | nonmail | sysmail) )
user_level(general | sa)
support_access(y | n)
timezone(zone)
date_format(yymmdd | mmddy | ddmmyy | yyddmm | mmyydd | ddyymm)
language(lang)
{profile_type(0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11
  | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20)} |

```

```
{profileinfo(
  reset_session(y | n)
  change_alias(y | n)
  immediate(y | n)
  eso(y | n)
  force_archive(y | n)
  archive_days(mm)
  change_groups(y | n)
  change_profiles(y | n)
  change_own(y | n)
  change_libs(y | n)
  purge_msg(y | n)
  deliver_msg_in_seq(y | n)
  audit_days(nn)
  change_x400(y | n)
  sending(nopay | blocked | paysend | payboth)
  receiving(nopay | blocked | payreceive | payboth)
  ccpartner(ccnopay | ccprocess | ccreceive | ccboth)
  ccrecipient(ccnopay | ccprocess | ccreceive | ccboth));
```

Parameters and subparameters

The following table describes the USER parameters and subparameters. Subparameters follow their related parameter and are italicized.

NAME	REQ'D	SIZE	DESCRIPTION
FUNCTION	Yes	6	The function to be performed on a user's profile - CHANGE.
ACCOUNT	Yes	8	The account ID of the user profile to be changed.
USERID	Yes	8	The user ID of the user profile to be changed. A cluster child cannot be specified. If you specify a cluster child ID, the system defaults to the cluster parent ID.
POSITIVE_ACK	No	1	This indicates whether the user wants a positive acknowledgment - Y or N. The default is Y if this parameter is not specified. The user will always receive negative acknowledgments. (Synonym: POS_ACK or POSACK).
REQUESTID	No	8	The submitter identifier for the command. The REQUESTID can be correlated with the acknowledgment message received. The default is blank. (Synonym: REQID).

NAME	REQ'D	SIZE	DESCRIPTION
<i>ALTADMINn</i>	No	N/A	<i>ALTADMINn</i> (where $n = 1, 2, 3,$ or 4) is a field with subparameters. It can be specified as <i>ALTADMINn</i> (), which will remove any values contained in the subparameters. It can also be specified with the following three parameters in this table as its subparameters. You cannot specify a cluster user ID as an alternate administrator.
<i>ACCOUNT</i>	No	8	The account ID of the user's alternate administrator. If <i>altadminn</i> <i>ACCOUNT</i> is specified without the <i>USERID</i> , all service administrators in the account specified will serve as alternate administrator for the user.
<i>USERID</i>	No	8	The user ID of the user's alternate administrator.
<i>AUTHORITY</i>	No	8	The alternate administrator authority level. If <i>altadminn</i> <i>AUTHORITY</i> is specified, the <i>ACCOUNT</i> must also be specified. The valid values are: <i>ALL</i> , <i>MAILONLY</i> , <i>NONMAIL</i> , and <i>SYSMAIL</i> . <ul style="list-style-type: none"> • <i>All</i> indicates full user access administrator and is the default. • <i>MAILONLY</i> indicates an administrator that may only view mail. • <i>NONMAIL</i> indicates an administrator that may only manipulate the user's profile data. • <i>SYSMAIL</i> indicates an administrator that may only manipulate the user's profile data and view messages from account ID *SYSTEM*. (Synonym: <i>AUTH</i>).
<i>USER_LEVEL</i>	No	7	The user's authorization level - <i>GENERAL</i> or <i>SA</i> . For a general user, specify <i>GENERAL</i> . For a service administrator, specify <i>SA</i> . (Synonym: <i>USRLEVEL</i>).

NAME	REQ'D	SIZE	DESCRIPTION
SUPPORT_ACCESS	No	1	This indicates whether the support personnel are able to view and delete the user's mail - Y or N. (Synonym: SUPPACC).
TIMEZONE	No	5	The user's preference for how times are displayed. This value can either be a valid time zone or an offset from GMT. Offsets are entered as <i>Ennnn</i> or <i>Wnnnn</i> , where <i>nnnn</i> indicates the amount of time either east or west of GMT; see Appendix A, "TIMEZONE values."
DATE_FORMAT	No	6	The user's preference for how dates are displayed. Specify <i>YYMMDD</i> , <i>MMDDYY</i> , <i>DDMMYY</i> , <i>YYDDMM</i> , <i>MMYYDD</i> , or <i>DDYYMM</i> . (Synonym: DATEFMAT).
LANGUAGE	No	2	The user's preference for the language used by Information Exchange. When an invalid value is entered, the system default is used.
PROFILE_TYPE	No	2	This indicates the user's profile type. It can be between 0 and 20. If you change any of the fields in PROFILEINFO, the PROFILE_TYPE is set to zero. If you set the PROFILE_TYPE to zero, the PROFILEINFO fields are not altered. PROFILE_TYPE and PROFILEINFO are mutually exclusive. (Synonym: PROFTYPE).
PROFILEINFO	No	N/A	PROFILEINFO is a field with subparameters. If it is specified with no parameters, <i>PROFILEINFO()</i> , no action is taken, and a warning is issued. PROFILEINFO can be specified with the following parameters as its subparameters. (Synonym: PROFINFO).
<i>RESET_SESSION</i>	No	1	This indicates whether the user can reset the user's own session - Y or N. (Synonym: RESETSES).
<i>CHANGE_ALIAS</i>	No	1	This indicates whether the user can create or update global or organizational tables - Y or N. (Synonym: CHGALIAS).

NAME	REQ'D	SIZE	DESCRIPTION
<i>IMMEDIATE</i>	No	1	This indicates whether the user can receive immediate priority messages - Y or N. (Synonym: IMMED).
<i>ESO</i>	No	1	This indicates whether the user is an Extended Security Option (ESO) user - Y or N.
<i>FORCE_ARCHIVE</i>	No	1	This indicates whether the user's mail is to be stored in the archive, regardless of what the receive command indicated - Y or N. If you set this field to Y, archiving will be billable based on the number of archive days. (Synonym: FORCARCH).
<i>ARCHIVE_DAYS</i>	No	3	This indicates the number of days the user's mail is to be stored in the archive. The minimum is zero, and the maximum is system defined. If an invalid numeric value is specified, it may default to the system-defined default. (Synonym: ARCHDAYS).
<i>CHANGE_GROUPS</i>	No	1	This indicates whether the user can add or update groups in the account and account - level distribution lists - Y or N. (Synonym: CHGGRPS).
<i>CHANGE_PROFILES</i>	No	1	This indicates whether the user can change another user's profile within the user's account - Y or N. (Synonym: CHGPROFS).
<i>CHANGE_OWN</i>	No	1	This indicates whether the user can change the user's own profile - Y or N. (Synonym: CHGOWN).
<i>CHANGE_LIBS</i>	No	1	This indicates whether the user can define and update libraries or library members within the user's account - Y or N. (Synonym: CHGLIBS).
<i>PURGE_MSG</i>	No	1	This indicates whether the user can use the message purge command from external interfaces - Y or N. (Synonym: PURGMSG).

NAME	REQ'D	SIZE	DESCRIPTION
<i>DELIVER_MSGS_IN_SEQ</i>	No	1	This indicates whether you want Information Exchange to force sequential delivery of messages. This includes only the transfer of messages to trading partners attached to the same Information Exchange as yourself and the construction of archived message sets for messages you have received. Specify Y to force sequential message processing. With a specification of N, Information Exchange can transfer the messages as quickly as possible with a higher level of availability. (Synonym: DELIVERSEQ).
<i>AUDIT_DAYS</i>	No	2	This indicates the number of days the user's audit trail is saved. The minimum and the maximum are system defined. If an invalid numeric value is specified, it may default to the system-defined default. (Synonym: AUDDAYS).
<i>CHANGE_X400</i>	No	1	This indicates whether the user can add or copy X.400 user profiles, or add, change, or delete X.400 trading partner attributes. (Synonym: CHGX400).
<i>SENDING</i>	No	7	This indicates the user's payment agreement when sending. The valid values are: NOPAY, BLOCKED, PAYSEND, and PAYBOTH. <ul style="list-style-type: none"> • <i>NOPAY</i> indicates that the user is not willing to pay. • <i>BLOCKED</i> indicates that the user is not able to send messages. • <i>PAYSEND</i> indicates that only the send-side charges are authorized for payment. • <i>PAYBOTH</i> indicates that both send-side and receive-side charges are authorized.

NAME	REQ'D	SIZE	DESCRIPTION
<i>RECEIVING</i>	No	10	<p>This indicates the user's payment agreement when receiving. The valid values are: <i>NOPAY</i>, <i>BLOCKED</i>, <i>PAYRECEIVE</i>, and <i>PAYBOTH</i>.</p> <ul style="list-style-type: none">• <i>NOPAY</i> indicates that the user is not willing to pay when receiving.• <i>BLOCKED</i> indicates that the user is not able to receive messages.• <i>PAYRECEIVE</i> indicates that only the receive-side charges are authorized for payment.• <i>PAYBOTH</i> indicates that both send-side and receive-side charges are authorized for payment. <p>(Synonym: <i>RECVING</i>).</p>

NAME	REQ'D	SIZE	DESCRIPTION
<i>CCPARTNER</i>	No	9	<p>This indicates this user's payment agreement for carbon copy charges when this user is the partner in a carbon copy relationship. This payment agreement only applies when:</p> <ul style="list-style-type: none"> The requester of a carbon copy specified "P" (partner) in either or both the "Who pays for processing" field or the "Who pays for receiving" field when setting up the carbon copy relationship. There is no entry in this user's Trading Partner List for the carbon copy requester. <p>The valid values are: <i>CCNOPAY</i>, <i>CCPROCESS</i>, <i>CCRECEIVE</i>, and <i>CCBOTH</i>.</p> <ul style="list-style-type: none"> <i>CCNOPAY</i> indicates that no carbon copy payments are authorized. <i>CCPROCESS</i> indicates that the user agrees to pay only the carbon copy processing charge. <i>CCRECEIVE</i> indicates that the user agrees to pay only the carbon copy receiving charge. <i>CCBOTH</i> indicates that the user is willing to pay both the carbon copy processing charge and the carbon copy receiving charge. <p>(Synonym: <i>CCPARTN</i>).</p>

NAME	REQ'D	SIZE	DESCRIPTION
<i>CCRECIPIENT</i>	No	9	<p>This indicates this user's payment agreement for carbon copy charges when this user is the recipient in a carbon copy relationship. This payment agreement only applies when:</p> <ul style="list-style-type: none"> The requester of a carbon copy specified "R" (recipient) in either or both the "Who pays for processing" field or the "Who pays for receiving" field when setting up the carbon copy relationship. There is no entry in this user's Trading Partner List for the carbon copy requester. <p>The valid values are: CCNOPAY, CCPROCESS, CCRECEIVE, and CCBOTH.</p> <ul style="list-style-type: none"> <i>CCNOPAY</i> indicates that no carbon copy payments are authorized. <i>CCPROCESS</i> indicates that the user agrees to pay only the carbon copy processing charge. <i>CCRECEIVE</i> indicates that the user agrees to pay only the carbon copy receiving charge. <i>CCBOTH</i> indicates that the user is willing to pay both the carbon copy processing charge and the carbon copy receiving charge. <p>(Synonym: CCRECIP).</p>

Understanding the administration mailbox messages

This chapter describes administration mailbox messages. The messages include the severity code and an explanation of actions the user needs to take. For an explanation of the severity codes, see “Severity codes” below.



NOTE: In many of the following messages, parameter values are represented by a series of letters; for example, *ACCOUNT(aaaaaaaaa)*.

When the actual message is displayed, this series of letters will be replaced with an actual value that will make the message more meaningful.

Severity codes

The following is an explanation of the message severity code values:

This value:	Indicates:
I	No error exists. Information Exchange has processed the command completely. Example: Message delivery receipt.
W	Warning. Information Exchange has processed the command, but some field has been ignored or a default value has been taken. Example: The value for <i>Audit Days</i> was invalid in the <i>USER</i> command and the default was taken.
E	Error. Information Exchange has not processed the command. Example: A message was sent to an invalid destination.

Messages

The following section describes the administration mailbox messages.

32001 ACCOUNT(aaaaaaaaa) does not exist.

Explanation: The account you specified does not exist.

Severity: E

System Action: Information Exchange has not processed the command.

User Response: Ensure the account is correct and retry the command. If it was entered correctly, you must first define the account before entering this command.

32002 USERID(uuuuuuuu) in ACCOUNT(aaaaaaaaa) does not exist.

Explanation: The account/user ID combination you specified does not exist.

Severity: E

System Action: Information Exchange has not processed the command.

User Response: Ensure the account/user ID combination is correct and retry the command. If they were entered correctly, you must first define the user ID in the specified account before entering this command.

32003 PARTNER(pppppppp) does not exist in this user's trading partner list.

Explanation: You tried to change/delete a trading partner that does not exist in the user's trading partner list.

Severity: E

System Action: Information Exchange has not processed the command.

User Response: Ensure the trading partner is correct and retry the command.

32004 ALIASTABLE(ttbl) does not exist.

Explanation: There is no alias table with the name and type you entered.

Severity: E

System Action: Information Exchange has not processed the command.

User Response: Ensure the table name and type are correct and retry the command.

32005 ACCOUNT(aaaaaaaaa) USERID(uuuuuuuu) is defined as a cluster child ID. The cluster parent ID was used.

Explanation: You tried to add, change, or get an object (such as a user profile or a trading partner list) that is owned by a cluster parent ID using the cluster child ID. The cluster parent ID will be used.

Severity: W

System Action: Information Exchange processed the command using the cluster parent ID as the owner.

User Response: If you did not want the cluster parent ID to be used, delete or update the object to reverse the system action for the cluster parent ID.

32006 LISTNAME(IIIIIII) for LISTTYPE(t) does not exist.

Explanation: The list name specified for the list type specified does not exist.

Severity: W

System Action: Information Exchange has not processed the command.

User Response: Ensure the list name is correct and retry the command.

32007 SYSID(sys) does not exist.

Explanation: The Information Exchange system ID you specified does not exist.

Severity: E

System Action: Information Exchange has not processed the command.

User Response: Ensure the system ID is correct and retry the command.

32008 ALIASNAME(aaaaaaaaa) cannot be resolved in table ALIASTABLE(tbl).

Explanation: The alias name specified was not defined.

Severity: E

System Action: Information Exchange has not processed the command.

User Response: Ensure the alias name exists and retry the command.

32009 PARTNER(pppppppp) already exists in this user's trading partner list.

Explanation: You tried to add a trading partner that already exists in the user's trading partner list.

Severity: E

System Action: Information Exchange has not processed the command.

User Response: Ensure the trading partner is correct and retry the command.

32010 **ACCOUNT(aaaaaaaaa) USERID(uuuuuuuu) does not have a trading partner list.**

Explanation: You tried to change, delete, or erase a trading partner list that does not exist.

Severity: E

System Action: Information Exchange has not processed the command.

User Response: You must first create a trading partner list before updating.

32011 **PARTNER(pppppppp) already existed in the user's trading partner list with the parameter values you requested. No update was performed.**

Explanation: You tried to add or change a partner in the user's trading partner list, but the partner already existed with the same parameter values that you requested.

Severity: W

System Action: Information Exchange processed the command, but it resulted in no changes.

User Response: No action is required.

32012 **ACCOUNT(aaaaaaaaa) USERID(uuuuuuuu) already existed with the same parameter values you requested. No update was performed.**

Explanation: You tried to update a user profile, but the user existed with the same parameter values that you requested.

Severity: W

System Action: Information Exchange processed the command, but it resulted in no changes.

User Response: No action is required.

32013 **PARTNER(pppppppp) is defined as a cluster child ID. The cluster parent ID was used.**

Explanation: You tried to add a trading partner list entry that is defined as a cluster child ID. The partner entry was added using the cluster parent ID.

Severity: W

System Action: Information Exchange processed the command and substituted the cluster parent ID in this field.

User Response: If you did not want the cluster parent ID to be used, delete the trading partner entry from the list.

32014 DESTINATION(ACCOUNT(aaaaaaa) USERID(uuuuuuu)) is defined as a cluster parent ID.

Explanation: You attempted to retrieve the output of a GET function into a destination mailbox that is defined as a cluster parent ID.

Severity: E

System Action: Information Exchange has not processed the command.

User Response: Enter a specific cluster child ID or an account ID and user ID that is not defined as a cluster ID as the DESTINATION of a GET function.

32015 DESTINATION(ALIASTABLE(tbl) ALIASNAME(nnnnnnn)) is defined as a cluster parent ID.

Explanation: You attempted to retrieve the output of a GET function into a destination mailbox that is defined as a cluster parent ID.

Severity: E

System Action: Information Exchange has not processed the command.

User Response: Enter a specific cluster child ID or an account ID and user ID that is not defined as a cluster ID as the DESTINATION of a GET function.

32021 A single command exceeds the nnn character maximum limit.

Explanation: You entered a command that exceeds the allowable character limit, which is 32767.

Severity: E

System Action: Information Exchange has not processed the command.

User Response: Check the command syntax and retry the command.

32022 The command has unbalanced quotes.

Explanation: You entered a command with a quote in the text, but no close quote was found.

Severity: E

System Action: Information Exchange has not processed the command.

User Response: Complete the quoted string by adding the close quote and re-enter the command.

32023 The command has unbalanced comment delimiters.

Explanation: You entered a command with a begin-comment delimiter in the text, but no end-comment delimiter was found.

Severity: E

System Action: Information Exchange has not processed the command.

User Response: Complete the comment by adding the end-comment symbol and re-enter the command.

32050 ACCOUNT(aaaaaaa) USERID(uuuuuuu) is not authorized to perform this function XXXXXXXX.

Explanation: The user does not have the proper authority to complete this function. XXXXXXXX will be substituted with more information.

Severity: E

System Action: Information Exchange has not processed the command.

User Response: Select another function or contact your service administrator.

32051 You are not authorized to GET for DESTINATION(ACCOUNT(aaaaaaa) USERID(uuuuuuu)).

Explanation: You attempted to retrieve the output of a GET function into a destination mailbox for which you are not authorized.

Severity: E

System Action: Information Exchange has not processed the command.

User Response: Select another function or contact your service administrator.

32052 You are not authorized to access XXXXXXXX.

Explanation: You do not have authorization to access the referenced object.

Severity: E

System Action: Information Exchange has not processed the command.

User Response: Select another function or contact your service administrator.

32100 An unrecognized parameter was used.

Explanation: You entered an invalid parameter.

Severity: E

System Action: Information Exchange has not processed the command.

User Response: Correct the parameter and retry the command.

32101 A required parameter XXXXXXXXX() was omitted.

Explanation: You failed to include a required parameter.

Severity: E

System Action: Information Exchange has not processed the command.

User Response: Add the required parameter and retry the command.

32102 Mutually exclusive parameters have been used. Error detected on XXXXXXXXX().

Explanation: The command entered contains incompatible parameters.

Severity: E

System Action: Information Exchange has not processed the command.

User Response: Remove one of the mutually exclusive parameters and retry the command.

32103 Keywords or values cannot continue on the next line.

Explanation: The command syntax did not contain a semicolon to indicate the end of the command.

Severity: E

System Action: Information Exchange has not processed the command.

User Response: Correct the command syntax and retry the command.

32105 Parameter XXXXXXXXX() was previously detected.

Explanation: The parameter has already been specified.

Severity: E

System Action: Information Exchange has not processed the command.

User Response: Retry the command with the parameter entered once.

32106 A severe syntax error was detected.

Explanation: There is a severe syntax error in the command issued.

Severity: E

System Action: Information Exchange has not processed the command.

User Response: Correct the syntax of the command and retry.

32107 XXXXXXXXX() is a required parameter when XXXXXXXXX() is used.

Explanation: You failed to include a required parameter.

Severity: E

System Action: Information Exchange has not processed the command.

User Response: Add the required parameter and retry the command.

32108 An unrecognized command was used.

Explanation: You entered an invalid command.

Severity: E

System Action: Information Exchange has not processed the command.

User Response: Ensure the command is correct and retry.

32110 XXXXXXXXX() contains a value of invalid length.

Explanation: The value you entered for this field has an invalid length.

Severity: E

System Action: Information Exchange has not processed the command.

User Response: Retry the command with a valid value.

32150 XXXXXXXXX(xxxxxxxx) is an invalid value.

Explanation: The value you entered for this field is not valid.

Severity: E

System Action: Information Exchange has not processed the command.

User Response: Retry the command with a valid value.

32151 No parameters were entered on the XXXXXXXXX() command to cause an update to occur.

Explanation: You tried to update using the listed command, but you did not enter any parameters that would cause an update to occur.

Severity: E

System Action: Information Exchange has not processed the command.

User Response: Correct the command and retry.

32200 The profile type for ACCOUNT(aaaaaaa) USERID(uuuuuuu) has been changed from XX to 00 due to your modifications.

Explanation: You have changed profile information for a user that belonged to a certain profile type. The profile type of that user has been changed to 00.

Severity: W

System Action: Information Exchange processed the command.

User Response: If you want the user to have a certain profile type, you must change the user's profile again. Otherwise, the user's profile will be unique.

32201 ALTADMINn(ACCOUNT(aaaaaaa) USERID(uuuuuuu)) is a duplicate administrator. The field was ignored.

Explanation: You have defined an administrator that already exists. The value entered was ignored.

Severity: W

System Action: Information Exchange processed the command, but it resulted in no changes to this field.

User Response: Correct the command and retry, if appropriate.

32202 XXXXXXXX(xx) is an invalid value, the default of yy was taken.

Explanation: You have entered an invalid value for the parameter.

Severity: W

System Action: Information Exchange has used the default value for this field.

User Response: Ensure the default is sufficient or retry the command.

32203 ARCHIVE_DAYS is set to nnn and FORCE_ARCHIVE is set to N, so archiving will only occur if requested in the receive command. It will be billable based on the number of archive days.

Explanation: You have entered a non-zero value (nnn) in the *Archive_Days* field but have left the *Force_Archive* field set to N. The system will not automatically archive every message in this case, but will archive only those messages for which the *Archive* field is set to Y in the receive message command. When archiving occurs, you will be billed for archive charges based on the number of archive days.

Severity: W

System Action: Information Exchange processed the command, but found an unusual condition.

User Response: Indicate the way you want received messages to be handled. To have received messages always stored in the archive, type Y in the *Force_Archive* field. To have received messages archived only as requested by the RECEIVE command, type N in the *Force_Archive* field. If you do not want the messages archived, specify zero in the *Archive days* field.

32204 Mail cannot be exchanged between ACCOUNT(aaaaaaa) USERID(uuuuuuu) and PARTNER(pppppppp) because the receiver is unknown.

Explanation: You updated a trading partner list with an unknown partner.

Severity: W

System Action: Information Exchange processed the command.

User Response: Verify the partner you entered is a valid user and retry the command.

32205 Mail cannot be exchanged between ACCOUNT(aaaaaaa) USERID(uuuuuuu) and PARTNER(pppppppp) because the receive charges do not agree.

Explanation: You updated a trading partner with receive-side payment levels that do not agree.

Severity: W

System Action: Information Exchange processed the command.

User Response: Ensure that this is your intended result.

32206 Mail cannot be exchanged between ACCOUNT(aaaaaaa) USERID(uuuuuuu) and PARTNER(pppppppp) because the send charges do not agree.

Explanation: You updated a trading partner with send-side payment levels that do not agree.

Severity: W

System Action: Information Exchange processed the command.

User Response: Ensure that this is your intended result.

32207 Mail cannot be exchanged between ACCOUNT(aaaaaaa) USERID(uuuuuuu) and PARTNER(ppppppp) because the send and receive charges do not agree.

Explanation: You updated a trading partner with send-side and receive-side payment levels that do not agree.

Severity: W

System Action: Information Exchange processed the command.

User Response: Ensure that this is your intended result.

32208 Mail exchanges between ACCOUNT(aaaaaaa) USERID(uuuuuuu) and PARTNER(ppppppp) may be limited because of ESO restrictions.

Explanation: You updated a trading partner list with a partner requesting ESO restrictions.

Severity: W

System Action: Information Exchange processed the command.

User Response: Ensure that this is your intended result.

32209 You entered XXXXXXXX(). No update was made for that field.

Explanation: You entered a field with subparameters, but no subparameters were updated.

Severity: W

System Action: Information Exchange processed the command, but it resulted in no changes to this field.

User Response: Ensure that this is your intended result.

32210 You may only update the user's preference fields. Any attempts to update other fields were ignored.

Explanation: You may only update the user's preference fields: TIMEZONE, DATE_FORMAT, and LANGUAGE. Any attempts to update other fields were ignored.

Severity: W

System Action: Information Exchange processed the command, but ignored some fields.

User Response: Ensure that this is your intended result.

32211 You may only update certain fields. Any attempts to update fields for which you are not authorized were ignored.

Explanation: You may only update certain fields. Any attempts to update protected fields were ignored. The protected fields are as follows: USER_LEVEL, PROFILE_TYPE, CHANGE_ALIAS, CHANGE_GROUPS, CHANGE_PROFILES, CHANGE_OWN, CHANGE_LIBS, PURGE_MSG, and CHANGE_X400.

Severity: W

System Action: Information Exchange processed the command, but ignored some fields.

User Response: Ensure that this is your intended result.

32212 You are not authorized to update the SUPPORT_ACCESS field.

Explanation: You may only update certain fields. Customer support users are not authorized to update the SUPPORT_ACCESS field.

Severity: W

System Action: Information Exchange processed the command, but ignored this field.

User Response: Ensure that this is your intended result.

32213 XXXXXXXX() is not supported on this installation of Information Exchange. The value you entered was ignored.

Explanation: You requested to change a field that is not supported on this installation of Information Exchange. The request to change that field was ignored.

Severity: W

System Action: Information Exchange processed the command, but ignored this field.

User Response: Ensure that this is your intended result.

32214 ALTADMINn(AACCOUNT(aaaaaaa) USERID(uuuuuuu)) is a cluster user, which is not allowed.

Explanation: The alternate administrator specified is a cluster user. Cluster users may not be specified as alternate administrators.

Severity: E

System Action: Information Exchange has not processed the command.

User Response: Do not specify a cluster user as an alternate administrator.

32215 Archiving is a billable function. The charges are based on the number of archive days.

Explanation: You have entered a non-zero value in the *Archive_days* field and the value of the *Force_Archive* field is set to **y**. The system will automatically archive every message. When archiving occurs, you will be billed for archive charges based on the number of archive days.

Severity: W

System Action: Information Exchange processed the command.

User Response: Indicate the way you want received messages to be handled. To have received mail always stored in the archive, type **y** in the *Force_Archive* field. To have received mail archived only as requested by the receive command, type **n** in the *Force_Archive* field.

32216 Archiving and ARCHIVE_DAYS (000) is invalid. The default of nnn days was taken.

Explanation: You have entered a value of **Y** for the *Force_archive* field and have entered a zero value in the *Archive_days* field.

Severity: W

System Action: Information Exchange has used the default value for the specified field.

User Response: Ensure the default is sufficient or retry the command.

32217 To pay charges as a carbon copy recipient, you cannot block communications as a receiver.

Explanation: You added or changed a user profile or trading partner list entry in such a way that the specified payment levels conflict.

Severity: E

System Action: Information Exchange has not processed the command.

User Response: If you wish to pay charges as a carbon copy recipient, you cannot specify “blocked” for the receiving field. If you wish to block communications when receiving, you cannot specify to pay charges as a carbon copy recipient.

32218 To pay charges as a carbon copy partner, you cannot block communications as both a sender and as a receiver.

Explanation: You added or changed a user profile or trading partner list entry in such a way that the specified payment levels conflict.

Severity: E

System Action: Information Exchange has not processed the command.

User Response: If you wish to pay charges as a carbon copy partner, you cannot specify “blocked” for both the sending and receiving fields. If you wish to block communications both when sending and receiving, you cannot specify to pay charges as a carbon copy partner.

32251 A system error occurred.

Explanation: An unexpected system error occurred.

Severity: E

System Action: Information Exchange has not processed the command.

User Response: Contact GXS Community Support.

32300 Request was processed successfully.

Explanation: The command was processed successfully.

Severity: I

System Action: No error exists. Information Exchange has processed the command completely.

User Response: No action is required.

32301 Request was not processed because of errors.

Explanation: The command was not processed because an error was detected. This message will be accompanied by one or more specific messages.

Severity: I

System Action: Information Exchange has not processed the command.

User Response: Correct the command and retry.

32302 The alias you specified resolved to SYSID(sys) ACCOUNT(aaaaaaaa)
USERID(uuuuuuuu).

Explanation: This message contains the actual user pointed to by the alias you specified.

Severity: I

System Action: No error exists. Information Exchange has processed the command completely.

User Response: No action is required.

32303 DESTINATION defaulted to ACCOUNT(aaaaaaaa) USERID(uuuuuuuu).

Explanation: This message indicates the default DESTINATION taken by the command you entered.

Severity: I

System Action: No error exists. Information Exchange has processed the command completely.

User Response: No action is required.

TIMEZONE values

This appendix lists the time zones, their abbreviations, and their offsets from Greenwich mean time (GMT).

The valid values for the TIMEZONE parameter are as follows:

TIMEZONE	GMT OFFSET	DESCRIPTION
EAD	E1000	Eastern Australia daylight
JST	E0900	Japan standard time
WED	E0200	Western Europe daylight
EMT	E0200	Eastern Mediterranean time
BST	E0100	British summer time
WES	E0100	Western Europe standard
GMT	E0000	Greenwich mean time
UTC	E0000	Coordinated Universal time
AST	W0400	Atlantic standard time
EDT	W0400	Eastern daylight time
EST	W0500	Eastern standard time
CDT	W0500	Central daylight time
CST	W0600	Central standard time
MDT	W0600	Mountain daylight time
MST	W0700	Mountain standard time
PDT	W0700	Pacific daylight time

TIMEZONE	GMT OFFSET	DESCRIPTION
PST	W0800	Pacific standard time
YDT	W0800	Alaska daylight time
YST	W0900	Alaska standard time
AHD	W1000	Hawaii standard time
AHS	W1000	Hawaii standard time

Panel values versus command syntax

The following tables show the relationship between values used in the Information Exchange Administration Services panels and values used in the command interface.

DESCRIPTION	COMMAND VALUE	PANEL VALUE	
		AS SENDER	Y/N
	SENDING		
Indicates that the user is not willing to pay when sending.	NOPAY	Will pay for sending?	N
		Will pay for receiving?	N
		Block communication?	N
Indicates that the user is not able to send messages.	BLOCKED	Will pay for sending?	N
		Will pay for receiving?	N
		Block communication?	Y
Indicates that only the send-side charges are authorized for payment when the user is sending.	PAYSEND	Will pay for sending?	Y
		Will pay for receiving?	N
		Block communication?	N
Indicates that both send-side and receive-side charges are authorized for payment when the user is sending.	PAYBOTH	Will pay for sending?	Y
		Will pay for receiving?	Y
		Block communication?	N
	RECEIVING		
Indicates that the user is not willing to pay when receiving.	NOPAY	Will pay for sending?	N
		Will pay for receiving?	N
		Block communication?	N
Indicates that the user is not able to receive messages.	BLOCKED	Will pay for sending?	N
		Will pay for receiving?	N
		Block communication?	Y

DESCRIPTION	COMMAND VALUE	PANEL VALUE	
Indicates that only the receive-side charges are authorized for payment when the user is receiving.	PAYRECEIVE	Will pay for sending?	N
		Will pay for receiving?	Y
		Block communication?	N
Indicates that both send-side and receive-side charges are authorized for payment when the user is receiving.	PAYBOTH	Will pay for sending?	Y
		Will pay for receiving?	Y
		Block communication?	N
	CCPARTNER	AS CARBON COPY PARTNER	Y/N
Indicates that the user is not willing to pay when the user is a carbon copy partner.	CCNOPAY	Will pay processing charge?	N
		Will pay for receiving?	N
Indicates that only the carbon copy processing charges are authorized for payment when the user is a carbon copy partner.	CCPROCESS	Will pay processing charge?	Y
		Will pay for receiving?	N
Indicates that only the carbon copy receive charges are authorized for payment when the user is a carbon copy partner.	CCRECEIVE	Will pay processing charge?	N
		Will pay for receiving?	Y
Indicates that both carbon copy processing and receive charges are authorized for payment when the user is a carbon copy partner.	CCBOTH	Will pay processing charge?	Y
		Will pay for receiving?	Y
	CCRECIPIENT	AS CARBON COPY RECIPIENT	Y/N
Indicates that the user is not willing to pay when the user is a carbon copy recipient.	CCNOPAY	Will pay processing charge?	N
		Will pay for receiving?	N
Indicates that only the carbon copy processing charges are authorized for payment when the user is a carbon copy recipient.	CCPROCESS	Will pay processing charge?	Y
		Will pay for receiving?	N
Indicates that only the carbon copy receive charges are authorized for payment when the user is a carbon copy recipient.	CCRECEIVE	Will pay processing charge?	N
		Will pay for receiving?	Y
Indicates that both carbon copy processing and receive charges are authorized for payment when the user is a carbon copy recipient.	CCBOTH	Will pay processing charge?	Y
		Will pay for receiving?	Y

Alternate administrator authority

When you designate a user as an alternate administrator, you also specify the type of authority that you want the administrator to have.

DESCRIPTION	COMMAND VALUE	PANEL VALUE
Alternate administrator that can perform all current administrator functions.	ALL	A
Alternate administrator that can only view mail.	MAILONLY	M
Alternate administrator that can do all current administrator functions except view and delete mail.	NONMAIL	N
Alternate administrator that can do all current administrator functions except view and delete mail, but can view messages from account ID *SYSTEM*.	SYSMAIL	S

Update Book Title Variable with your Book Title Here

Alternate administrator authority