Connect:Enterprise UNIX®

User's Guide

Version 2.4

Sterling Commerce
An IBM Company

Connect:Enterprise UNIX User's Guide
Version 2.4

First Edition

Sterling Commerce, Inc.

4600 Lakehurst Court Dublin, OH 43016-2000 * 614/793-7000

Contents

Chapter 1

Local User Commands

The commands described in this chapter enable users at the server location to manipulate data in the repository. Use the command line utilities to do the following tasks:

- Add outbound files to the repository as batches that can then be transmitted to designated remote sites (**cmuadd**)
- Mark batches logically deleted in the repository (**cmudelete**)
- Erase batches from the repository (**cmuerase**)
- Extract batches from the repository collected from remote sites and host site users (**cmuextract**)
- List batches currently in the repository (**cmulist**)
- Update the status of batches and user batch IDs by adding or deleting flags associated with each batch (**cmustatus**)
- Generate a key pair and certificate request for Secure FTP (**cmusslgencsr**)
- Verify the format of the key and certificate and display the certificate in a human-readable format (**cmusslverify**)
- Display the version information for your installation of Connect:Enterprise (**cmuvers**)
- Change your password (**ceupasswd**)

Connect:Enterprise Concepts

To understand Connect:Enterprise, you should be familiar with two basic concepts—batches and flags.

Batches

Data files stored by the repository are called *batches*. Batches are identified by three attributes: batch number, batch ID, and mailbox ID (also known simply as ID).

The batch number is automatically assigned by Connect:Enterprise when the batch is added to the repository. This number is unique for each batch currently on file, and is a number from 1 to 99,999,999.

The batch ID (BID) is the name of the batch (also referred to as user batch ID). When you send a batch to the repository, the batch ID can be used to describe the contents of the batch, such as *Sales Report for January* or *Memo to all personnel*.

The ID specifies the mailbox associated with the batch. For example, when you send a batch, you specify the mailbox ID of the site to receive the batch. Or, when you request a batch, you specify the mailbox ID of the batches being requested. Though the mailbox ID is not required, it should be supplied with all batches sent to the repository unless you are directed otherwise by host site personnel.

In the special case when a batch must be available for transmission to all remote sites, the host site personnel can assign a general mailbox ID, that can be accessed by all remote sites. If your application uses such general mailbox IDs, be sure that the correct mailbox IDs are used when accessing the Connect:Enterprise system.

Flags

Flags are labels assigned to each batch. The commands use flags to identify processes, protocols, and data formats that apply to the data. The following tables list the possible flag values.

Each process flag is explained in the following table:

Process Flags	Description
A	Batch was Added locally by the host site
C	Batch was Collected from a remote site
D	Batch is logically Deleted
E	Batch was Extracted successfully at least once
I	Collected batch is Incomplete
M	Batch can be transmitted Multiple times
N	Batch is permanently Non-transmittable
O	Batch is encrypted
P	Online transmission is in Progress
R	Batch is Requestable for remote connects or auto connects
S	LOG batch
T	Batch was Transmitted successfully. If the batch is flagged as multi-transmittable (M), The T flag is not turned on after the batch is successfully transmitted
U	Batch is permanently Unextractable
X	Batch was transmitted using Bisync with **CODE=A** or **CODE=B** (transparent mode Bisync) and **INDEX=YES**.

Each protocol flag is explained in the following table:

Protocol Flags	Description
A (TCP)	Batch originated at the host site
B (BSC)	Batch originated at a remote Bisync site
F (FTP)	Batch originated at a remote FTP site
G (FTP)	Batch was added with SSL (Secure FTP or HTTPS)
H (HTTP)	Batch was added by the HTTP daemon.
J (BP)	Batch was added by a GIS business process.
L (SSHFTP)	Batch was added with SSHFTP.
Q (ASY)	Batch originated at a remote Async site
V (BP)	Batch was acknowledged by the GIS business process.
W (AS2)	Batch was added by the EDIINT daemon.

Each data format flag is explained in the following table:

Data Format Flags	Description
K (EBC)	EBCDIC (nontransparent text)
Y (BIN)	Binary
Z (ASC)	ASCII

Adding Batch Files to the Repository (cmuadd)

Use the **cmuadd** utility to add batch files to the repository. These files can then be transmitted to remote sites. A newly created batch, added using default **cmuadd** parameters, is flagged as locally added (A) and requestable (R). The requestable flag makes it eligible for transmission to a remote site. One of the three data format flags is always posted to indicate the format of each batch. Refer to *Flags* on page 6 for a complete list of all flag values.

With Connect:Enterprise UNIX (Secure FTP), the mailbox ID associated with a batch to be added is checked against the mailbox IDs listed in encrypt.cfg. If the mailbox ID matches one specified in encrypt.cfg as having either strong, weak, or Triple DES encryption, a batch instance key is created. The batch instance key is used to encrypt the batch. The batch instance key is encrypted using the global key or Triple DES global key and stored in a header line added to the batch, which also indicates the encryption level. The batch is added with the appropriate encryption level and the O flag is set.

Note: It is possible to add batches with batch ID fields containing characters that may act as meta characters in the UNIX shell environment and the operating system environments where the batches are to be sent. Using non-alphanumeric characters in batch IDs may produce unpredictable results; therefore, take appropriate precautions in any context where the batch ID will be exposed to a UNIX shell or other scripting type of environments. Similarly, nonalphanumeric characters in the batch ID may be interpreted as HTML tags in the Site Administration User Interface or WebDAV mailbox listing. Avoid using the following: <, >, %, !, *.

To add batches to the repository, complete the following steps:

1. Select parameters from the following table to define the data you want to add. All parameters are optional.

   Parameters can be input using either the abbreviated or long format. The abbreviated format begins with a single hyphen and no space is required between the parameter and the values. The long format begins with two hyphens. You must separate the parameter and values with a space or an equal sign.

Parameter	Description
-b *batchid* --bid *batchid*	assigns name to batch being added
-c ***a|b|e*** --code *a|b|e*	identifies data format of batch
-d --delete	prompts to delete input files after a successful add.
-e *key* --encr *key*	invokes proprietary encryption. You can use an encryption key between 3 and 8 characters.
-H *hostname* --host *hostname*	host computer where cmuctld started
-i *mboxid* --id *mboxid*	assigns mailbox ID to batch
inputfiles	input files to add
-k --keepadd	retains a $$ADD card in data file
-L *filename* --link *filename*	establishes links to input file
-M --more	posts output messages one screen at a time
-m --multxmit	batch can be transmitted multiple times

Parameter	Description
-o --xmitonce	batch is only transmitted once
-P *portno* --port *portno*	port number where cmuctld started
-p *password* --passwd *password*	Connect:Enterprise password
-Q --quiet	suppresses pagination
-s *nnnnnnnn* --split *nnnnnnnn*	splits large input files
-t --trigger	immediately routes files to remotes
-u *username* --user *username*	user name other than UNIX login
-U *filename* --usrrcd *filename*	file to write to beginning of batch file
-X *nnnn* --cols *nnnn*	columns available for screen display
-Y *nnnn* --rows *nnnn*	rows available for screen display
-Z *level*	turns on tracing at the specified level and writes output to the current directory as a file named in the format: cmuadd.out.*pid*, where *pid* is the process ID assigned by the operating system. Valid values are 1–99.
-? --help	displays usage message

Note: When you run cmuadd without supplying a file name, the system prompts you for the data from your terminal. You must either type in the file from the terminal and signal end-of-file with Ctrl + D, or cancel the operation with Ctrl + C. If you cancel the operation, a batch is created with an I flag indicating an incomplete batch.

2. Enter the **cmuadd** command similar to the following example, using the parameters and values determined in step 1. See *cmuadd Examples* on page 29 for additional examples using the **cmuadd** command.

```
cmuadd -iSFO -b"sample add" -m -c a cmuconnect.out
```

or

```
cmuadd --id SFO --bid "sample add" --multxmit --code a cmuconnect.out
```

A report similar to the following example is displayed. See *Reports* on page 36 for definitions of the fields on the report.

```
Date: mm/dd/yy           Connect:Enterprise UNIX n.n.nn Page: 0001
Time: 09:53:59             ADD Utility
Command Line Parameters:
    cmuadd
    -i SFO
    -b sample add
    -m cmuconnect.out
Mbx ID   Batch #   Batch ID   Bytes    Date-Time   Status   Org ID
SFO      10      sample add  134    02/03/14-09:53  ARMZO   cheryl
Input Files Processed
File Name      File Size     Pr Status
cmuconnect.out     134       processed
Summary Information
```

```
Date: 03/14/02           Connect:Enterprise UNIX nn.n.nn Page: 0002
Time:09:53:59                  ADD Utility
Number of Batches Added:    1
Number of Input Files:      1
Number of Files Bypassed:   0
Number of Input Bytes:      134
```

Deleting Batch Files from the Repository (cmudelete)

The **cmudelete** utility flags a batch file for deletion from the repository. Once flagged as deleted (D), it cannot be accessed by a remote site. However, the file is not *physically* erased until the **cmuerase** utility is run. You can issue the **cmustatus** command to turn off the delete flag if the batch is deleted prematurely. Batches flagged as deleted can still be extracted by the host site if specific batch numbers are designated when using the **cmuextract** command.

To delete batches from the repository, complete the following steps:

1. Select the parameters from the following table to define the data you want to delete. You must include either the **-b** or the **-i** parameter.

   Parameters can be input using either the abbreviated or long format. The abbreviated format begins with a single hyphen and no space is required between the parameter and the values. The long format begins with two hyphens. You must separate the parameter and values with a space or an equal sign. Required parameters are in bold.

Parameter	Description
-b *batchid* **--bid *batchid***	**specifies a user batch ID**
-i *mboxid* **--id *mboxid***	**mailbox ID of batch**
-E *[CC]yymmdd|nnn[[/\:]hhmm]* --end *[CC]yymmdd|nnn[[/\:]hhmm]*	selects batches created on or before date
-F *[!]{A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z]* --flags *[!]{A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z]*	selects batches with specified flags
-f *[CC]yymmdd|nnn[[/\:]hhmm]* --from *[CC]yymmdd|nnn[[/\:]hhmm]*	selects batches created on or after date
-H *hostname* --host *hostname*	host computer where cmuctld started
-M --more	posts output messages one screen at a time
-O *origid* --orig *origid*	originator of the batch
-P *portno* --port *portno*	port number where cmuctld started
-p *password* --passwd *password*	originator of the batch
-Q --quiet	suppresses pagination
-S *[CC]yymmdd|nnn[[/\:]hhmm]* --start *[CC]yymmdd|nnn[[/\:]hhmm]*	selects batches created on or after date
-t *[CC]yymmdd|nnn[[/\:]hhmm]* --to *[CC]yymmdd|nnn[[/\:]hhmm]*	selects batches created on or before date
-u *username* --user *username*	username other than UNIX login
-X *nnnn* --cols *nnnn*	columns available for screen display
-Y *nnnn* --rows *nnnn*	rows available for screen display

Parameter	Description
-Z *level*	turns on tracing at the specified level and writes output to the current directory as a file named in the format: cmuadd.out.*pid*, where *pid* is the process ID assigned by the operating system.
-? --help	displays usage message

2. Enter the **cmudelete** command similar to the following example, using the parameters and values determined in step 1. See *cmudelete Examples* on page 32 for additional examples using the **cmudelete** command.

```
cmudelete -itest -b"sample add"
```

or

```
cmudelete --id test --bid "sample add"
```

A report similar to the following example is displayed. See *Reports* on page 36 for definitions of the fields and more detailed instructions.

```
Date: mm/dd/yy           Connect:Enterprise UNIX n.n.nn     Page: 0001
Time: 09:57:18                 DELETE Utility
Command Line Parameters:
    cmudelete
    -i test
    -b sample add
Batch Status Code Values:
A - Added Offline                                   D - Flagged for Delete
I - Incomplete Collection                           C - Collected online
T - Online Transmit Done                            R - Online Request Allowed
E - Extracted batch                                 M - Multiple transmission
P - Transmission in Progress                        U - Batch Unextractable
N - Batch Nontransmittable                          B - BSC
F - FTP                                             W - AS2
K - EBCDIC                                          H - HTTP
Y - BINARY                                          Q - ASYNC
G - SSL                                             S - LOG batch
X - Transparent Bisync Index=Yes                    O - Batch Encryption
Z - ASCII                                           V - BP Verified
L - SSHFTP                                          J - Business Process
Mbx ID   Batch #   Batch ID   Bytes     Date-Time   Status  Org ID
```

```
Date: mm/dd/yy          Connect:Enterprise UNIX n.n.nn    Page: 0002
Time: 09:57:18                DELETE Utility
test     4    sample add    134     02/03/11-11:15  ADRMZ   cheryl
test     5    sample add    134     02/03/11-11:17  ADRMZ   cheryl
test     6    sample add    134     02/03/11-11:18  ADRMZ   cheryl
test     7    sample add    134     02/03/11-15:16  ADRMZ   cheryl
test     8    sample add    134     02/03/11-15:17  ADRMZ   cheryl
test     9    sample add    134     02/03/14-09:52  ADRMZ   cheryl
test     10   sample add    134     02/03/14-09:53  ADRMZ   cheryl
```

Erasing Batches from the Repository (cmuerase)

The **cmuerase** utility physically erases batches from the repository. You cannot restore batches that you have erased using this command. This differs from the **cmudelete** utility that flags a file for deletion.

Note: Run the **cmuerase** utility at regular intervals to keep the size of the mailbox directories under control.

To erase batches from the repository, complete the following steps:

1. Select the parameters from the following table to define the data you want to erase. You must include either the **-b** or the **-i** parameter.

   The **cmuerase** utility requires at least one limiting parameter. If no parameters are specified, no batches are erased.

   Parameters can be input using either the abbreviated or long format. The abbreviated format begins with a single hyphen and no space is required between the parameter and the values. The long format begins with two hyphens. You must separate the parameter and values with a space or an equal sign. Required parameters are in bold.

Parameter	Description
-b *batchid* **--bid *batchid***	**specifies a user batch ID**
-i *mboxid* **--id *mboxid***	mailbox ID of batch
-a *TIDE* --and *TIDE*	selects batches with all listed flags
-E *[CC]yymmdd|nnn[[/\:]hhmm]]* --end *[CC]yymmdd|nnn[[/\:]hhmm]]*	selects batches created on or before date

Parameter	Description
-F *[!]{A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z]* --flags *[!]{A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z]*	selects batches with specified flags
-f *[CC]yymmdd|nnn[[/|:]hhmm]* --from *[CC]yymmdd|nnn[[/|:]hhmm]*	selects batches created on or after date
-H *hostname* --host *hostname*	host computer where cmuctld started
-M --more	posts output messages one screen at a time
-o *TIDE* --or *TIDE*	selects batches with any of the listed flags
-O *origid* --orig *origid*	originator of the batch
-P *portno* --port *portno*	port number where cmuctld started
-p *password* --passwd *password*	originator of the batch
-Q --quiet	suppresses pagination
-S *[CC]yymmdd|nnn[[/|:]hhmm]* --start *[CC]yymmdd|nnn[[/|:]hhmm]*	selects batches created on or after date
-t *[CC]yymmdd|nnn[[/|:]hhmm]* --to *[CC]yymmdd|nnn[[/|:]hhmm]*	selects batches created on or before date
-u *username* --user *username*	username other than UNIX login
-X *nnnn* --cols *nnnn*	columns available for screen display
-Y *nnnn* --rows *nnnn*	rows available for screen display
-Z *level*	turns on tracing at the specified level and writes output to the current directory as a file named in the format: cmuadd.out.*pid*, where *pid* is the process ID assigned by the operating system.
-? --help	displays usage message

2. Enter the **cmuerase** command similar to the following example, using the parameters and values determined in step 1. See *cmuerase Examples* on page 32 for additional examples using the **cmuerase** command.

```
cmuerase -itest -b"sample add"
```

or

```
cmuerase --id test --bid "sample add"
```

A report similar to the following example is displayed. See *Reports* on page 36 for definitions of the fields and more detailed instructions.

```
Date: mm/dd/yy       Connect:Enterprise UNIX n.n.nn     Page: 0001
Time: 10:19:15           ERASE Utility
Command Line Parameters:
    cmuerase
    -i test
    -b sample add
Mbx ID   Batch #   Batch ID   Bytes   Date-Time   Status    Org ID
test     4     sample add  134     99/03/11-11:15   ADRMZ   cheryl
test     5     sample add  134     99/03/11-11:17   ADRMZ   cheryl
test     6     sample add  134     99/03/11-11:18   ADRMZ   cheryl
test     11    sample add  130     99/03/14-10:12   ARZ     cheryl
test     12    sample add   34     99/03/14-10:14   ARMZ    cheryl
```

Extracting Batches from the Repository (cmuextract)

The **cmuextract** utility extracts batches from the repository collected from remote sites during remote connects and auto connects, or added by host users. Batches are flagged as extracted (E) upon successful extraction. Batches flagged unextractable, deleted, or incomplete are not eligible for extraction. Deleted or incomplete batches can be forced to extract by specifying their batch numbers using **cmuextract -b** (or **--bid**). Unextractable batches are permanently unavailable. Previously extracted batches can be extracted again. Refer to the *Flags* on page 6 for a complete list of all flag values.

With Connect:Enterprise UNIX (Secure FTP), the Mailbox daemon determines if the batch requested for extraction has been stored with encryption. If the O flag is set, the daemon reads the header line to determine the encryption strength. The batch instance key is decrypted using the global key and the batch is extracted with decryption.

To extract batches from the repository, complete the following steps:

1. Select the parameters from the following table to define the data you want to extract. You must include either the **-b** or the **-i** parameter.

   Parameters can be input using either the abbreviated or long format. The abbreviated format begins with a single hyphen and no space is required between the parameter and the values.

The long format begins with two hyphens. You must separate the parameter and values with a space or an equal sign. Required parameters are in bold.

Parameter	Description
-b *batchid* **--bid *batchid***	**specifies a user batch ID**
-i *mboxid* **--id *mboxid***	**mailbox ID of batch**
-c *a|e|n* --conv *a|e|n*	specifies data format for output
-D *key* --decr *key*	invokes proprietary decryption. You can use an encryption key between 3 and 8 characters.
-d --delete	flags batches as logically deleted
-E *[CC]yymmdd|nnn[[/\:]hhmm]* --end *[CC]yymmdd|nnn[[/\:]hhmm]*	selects batches created on or before date
-e *filename* --enddat *filename*	appends the contents of the named file to the end of the output file after the batch is extracted.
-F *[!]{A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|V|W|X|Y|Z]* --flags *[!]{A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|V|W|X|Y|Z]*	selects batches with specified flags
-f *filename* --output *filename*	names the output file
-g --gplus	inserts a GENTRAN header
-H *hostname* --host *hostname*	host computer where cmuctld started
-l *reclen[:rs char]* --reclen *reclen[:rs char]*	record length
-M --more	posts output messages one screen at a time
-O *origid* --orig *origid*	originator of the batch
-o --onebatch	extracts the first complete batch

Parameter	Description
-P *portno* --port *portno*	port number where cmuctld started
-p *password* --passwd *password*	Connect:Enterprise password
-Q --quiet	suppresses pagination
-r *filename* --report *filename*	names output file for report
-S *[CC]yymmdd|nnn[[/|:]hhmm]* --start *[CC]yymmdd|nnn[[/|:]hhmm]*	selects batches created on or after date
-U *filename* usrrcd *filename*	file of keywords to write before output file
-u *username* --user *username*	username other than UNIX login
-w *r|c|t|a* --write *r|c|t|a*	specifies how output file is opened
-X *nnnn* --cols *nnnn*	columns available for screen display
-Y *nnnn* --rows *nnnn*	rows available for screen display
-Z *level*	turns on tracing at the specified level and writes output to the current directory as a file named in the format: cmuadd.out.*pid*, where *pid* is the process ID assigned by the operating system.
-? --help	displays usage message

2. Enter the **cmuextract** command similar to the following example, using the parameters and values determined in step 1. See *cmuextract Examples* on page 33 for additional examples using the **cmuextract** command.

```
cmuextract -itest -b"sample add" -freport.23
```

or

```
cmuextract --id test --bid "sample add" --output report.23
```

A report similar to the following two-page example is displayed. See *Reports* on page 36 for definitions of the fields and more detailed instructions.

```
Date: mm/dd/yy       Connect:Enterprise UNIX n.n.nn    Page: 0001
Time: 10:02:19             EXTRACT Utility
Command Line Parameters:
    cmuextract
    -i test
    -b sample add
Mbox ID  Batch #   Batch ID     Bytes   Date-Time    Status    Pr+
test     4    sample add    134     02/03/11-11:15   ARMZ       99
test     5    sample add    134     02/03/11-11:17   ARMZ       99
test     6    sample add    134     02/03/11-11:18   ADRMZ      14
test     7    sample add    134     02/03/11-15:16   ARMZ       99
test     8    sample add    134     02/03/11-15:17   ARMZ       99
test     9    sample add    134     02/03/14-09:52   ARMZ       99
test     10   sample add    134     02/03/14-09:53   ARMZ       99
Processing Message Values
0  - Extract Failed
11 - Batch bypassed due to transparent data
```

```
Date: mm/dd/yy       Connect:Enterprise UNIX n.n.nn    Page: 0002
Time: 10:02:20           EXTRACT Utility
12 - Batch bypassed due to incomplete data
13 - Requested incomplete batch extracted
14 - Batch bypassed due to delete flag
15 - Requested deleted batch extracted
18 - Batch Re-Extracted
98 - Batch Bypassed
99 - Extract Okay
```

Displaying a List of Batches in the Repository (cmulist)

Use the **cmulist** utility to display a list of all batches or a selected list of batches in the repository. One process flag that is unique to **cmulist** is the Special Batch (S) flag. The Special Batches consist of a receipt of the data, verifying the delivery of data. Refer to the *Flags* on page 6 for a full description of the flags.

To list batches in the repository, complete the following steps:

1. Select the parameters from the following table to define the data you want to list. All parameters are optional.

   Parameters can be input using either the abbreviated or long format. The abbreviated format begins with a single hyphen and no space is required between the parameter and the values. The long format begins with two hyphens. You must separate the parameter and values with a space or an equal sign.

Parameter	Description
-b *batchid* --bid *batchid*	specifies a user batch ID
-E *[CC]yymmdd|nnn[[/|:]hhmm]* --end *[CC]yymmdd|nnn[[/|:]hhmm]*	selects batches created on or before date
-F *[!]{A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z]* --flags *[!]{A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z]*	selects batches with specified flags
-f *[CC]yymmdd|nnn[[/|:]hhmm]* --from *[CC]yymmdd|nnn[[/|:]hhmm]*	selects batches created on or after date
-H *hostname* --host *hostname*	host computer where cmuctld started
-i *mboxid* --id *mboxid*	mailbox ID of batch
-M --more	post output messages one screen at a time
-O *origid* --orig *origid*	originator of batch
-P *portno* --port *portno*	port number where cmuctld started
-p *password* --passwd *password*	Connect:Enterprise password
-Q --quiet	suppresses pagination
-S *[CC]yymmdd|nnn[[/|:]hhmm]* --start *[CC]yymmdd|nnn[[/|:]hhmm]*	selects batches created on or after date
-t *[CC]yymmdd|nnn[[/|:]hhmm]* --to *[CC]yymmdd|nnn[[/|:]hhmm]*	selects batches created on or before date
-u *username* --user *username*	username other than UNIX login
-X *nnnn* --cols *nnnn*	columns available for screen display
-Y *nnnn* --rows *nnnn*	rows available for screen display

Parameter	Description
-Z *level*	turns on tracing at the specified level and writes output to the current directory as a file named in the format: cmuadd.out.*pid*, where *pid* is the process ID assigned by the operating system.
-? --help	displays usage message

2. Enter the **cmulist** command similar to the following example, using the parameters and values determined in step 1. See *cmulist Examples* on page 34 for additional examples using the **cmulist** command.

```
cmulist -itest -b"sample add" >> list
```

or

```
cmulist --id test --bid"sample add" >> list
```

A report similar to the following example is appended to the file called *list*. See *Reports* on page 36 for definitions of the fields and more detailed instructions.

```
Date: mm/dd/yy       Connect:Enterprise UNIX n.n.nn      Page: 0001
Time: 10:06:38            LIST Utility
Command Line Parameters:
    cmulist
    -i test
    -b sample add
Batch Status Code Values:
A - Added Offline                                  D - Flagged for Delete
I - Incomplete Collection                          C - Collected online
T - Online Transmit Done                           R - Online Request Allowed
E - Extracted batch                                M - Multiple transmission
P - Transmission in Progress                       U - Batch Unextractable
N - Batch Nontransmittable                         B - BSC
F - FTP                                            W - AS2
K - EBCDIC                                         H - HTTP
Y - BINARY                                         Q - ASYNC
G - SSL                                            S - LOG batch
X - Transparent Bisync Index=Yes                   O - Batch Encryption
Z - ASCII                                          V - BP Verified
L - SSHFTP                                         J - Business Process
```

```
Date: mm/dd/yy       Connect:Enterprise UNIX n.n.nn    Page: 0002
Time: 10:06:38             LIST Utility
Mbx ID  Batch #   Batch ID   Bytes    Date-Time     Status  Org ID
test      4  sample add   134     99/03/11-11:15   ADRMZ   cheryl
test      5  sample add   134     99/03/11-11:17   ADRMZ   cheryl
test      6  sample add   134     99/03/11-11:18   ADRMZ   cheryl
test      7  sample add   134     99/03/11-15:16   ADRMZ   cheryl
test      8  sample add   134     99/03/11-15:17   ADRMZ   cheryl
test      9  sample add   134     99/03/14-09:52   ADRMZ   cheryl
test     10 sample add   134     99/03/14-09:53   ADRMZ   cheryl
```

AS2 Batches

AS2 batches display differently in the cmulist utility. The batch name includes the batch number to make it easier for you to correlate batches as they are processed.

The following cmulist output example shows batch correlation of the .PL, .RQ, and .MD batches for an outbound batch:

```
sles8ssl 15  test_five.PL       5        06/01/23-17:11     ARTZ  psmil1
sles8ssl 16  test_five.PL.15.RQ 2395     06/01/23-17:11     CRTWY   EDIINT
sles8ssl 17  test_five.PL.15.MD 2967    06/01/23-17:11    CGHY    EDIINT
```

The following cmulist output example shows batch correlation of the .OK, .AS2, and .MD batches for an inbound batch:

```
inbound 4 from-cerhas21-data.4.OK  5   06/01/23-15    CWY  EDIINT
inbound 5 from-cerhas21-data.4.AS2 401 06/01/23-15:   CRWY EDIINT
inbound 6 from-cerhas21-data.4.MD  864 06/01/23-15:56  CWY  EDIINT
```

Updating the Status of Batches and User Batch IDs (cmustatus)

The **cmustatus** utility enables authorized users to:

- Change the process flags that identify the batch status (such as transmit, delete, and extract flags).
- Change the data format flags for the batch contents (for example, ASCII, EBCDIC, or BINARY).
- Modify mailbox IDs and/or user batch IDs.

Refer to *Flags* on page 6 for a complete list of all flag values.

To display the current status of the selected batches without making changes, run **cmustatus** and omit the on and off parameters.

To change the status or flags of batches in the repository, or to change mailbox IDs or batch IDs, complete the following steps:

1. Select the parameters from the following table to define the data you want to change. You must include either the **-b** or the **-i** parameter.

   Parameters can be input using either the abbreviated or long format. The abbreviated format begins with a single hyphen and no space is required between the parameter and the values. The long format begins with two hyphens. You must separate the parameter and values with a space or an equal sign. Required parameters are in bold.

Parameter	Description
-b *batchid* **--bid *batchid***	**specifies a user batch ID**
-i *mboxid* **--id *mboxid***	**mailbox ID of batch**
-B *newbatchid* --nbid *newbatchid*	replaces batch ID with a new one
-E *[CC]yymmdd|nnn[[/\:]hhmm]* --end *[CC]yymmdd|nnn[[/\:]hhmm]*	selects batches created on or before date
-F *[!]{A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z]* --flags *[!]{A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z]*	selects batches with specified flags
-f *[CC]yymmdd|nnn[[/\:]hhmm]* --from *[CC]yymmdd|nnn[[/\:]hhmm]*	selects batches created on or after date
-H *hostname* --host *hostname*	host computer where cmuctld started
-I *newmboxid* --nid *newmboxid*	replaces mailbox ID with a new one
-M --more	posts output messages one screen at a time
-O *origid* --orig *origid*	originator of batch
-o *RDTEM a|e|b|V* --on *RDTEM a|e|b|V*	turns flags on
-P *portno* --port *portno*	port number where cmuctld started
-p *password* --passwd *password*	Connect:Enterprise password

Parameter	Description
-Q --quiet	suppresses pagination
-S *[CC]yymmdd|nnn[[/\:]hhmm]* --start *[CC]yymmdd|nnn[[/\:]hhmm]*	selects batches created on or after date
-t *[CC]yymmdd|nnn[[/\:]hhmm]* --to *[CC]yymmdd|nnn[[/\:]hhmm]*	selects batches created on or before date
-u *username* --user *username*	username other than UNIX login
-X *nnnn* --cols *nnnn*	columns available for screen display
-x *RDTEM a|e|b|V* --off *RDTEM a|e|b|V*	turns flags off
-Y *nnnn* --rows *nnnn*	rows available for screen display
-Z *level*	turns on tracing at the specified level and writes output to the current directory as a file named in the format: cmuadd.out.*pid*, where *pid* is the process ID assigned by the operating system.
-? --help	displays usage message

2. Enter the **cmustatus** command similar to the following example, using the parameters and values determined in step 1.

```
cmustatus -i -test -b"sample add" -xD -oR
```

or

```
cmustatus -id test -bid "sample add" --off D --on R
```

A report similar to the following example is displayed. See *cmustatus Examples* on page 34 for definitions of the fields and more detailed instructions.

```
Date: mm/dd/yy       Connect:Enterprise UNIX n.n.nn    Page: 0001
Time: 10:15:09          STATUS Utility
Command Line Parameters:
    cmustatus
    -i test
    -b sample add
Batch Status Code Values:
A - Added Offline                                   D - Flagged for Delete
I - Incomplete Collection                           C - Collected online
T - Online Transmit Done                            R - Online Request Allowed
E - Extracted batch                                 M - Multiple transmission
P - Transmission in Progress                        U - Batch Unextractable
N - Batch Nontransmittable                          B - BSC
F - FTP                                             W - AS2
K - EBCDIC                                          H - HTTP
Y - BINARY                                          Q - ASYNC
G - SSL                                             S - LOG batch
X - Transparent Bisync Index=Yes                    O - Batch Encryption
Z - ASCII                                           V - BP Verified
L - SSHFTP                                          J - Business Process

Mbx ID   Batch #   Batch ID    Bytes    Date-Time   Status  Org ID
```

```
Date: mm/dd/yy       Connect:Enterprise UNIX n.n.nn        Page: 0002
Time: 10:15:09           STATUS Utility
test      4  sample add       134     02/03/11-11:15   ARMZ  cheryl
test      5  sample add       134     02/03/11-11:17   ARMZ  cheryl
test      6  sample add       134     02/03/11-11:18   ARMZ  cheryl
test      7  sample add       134     02/03/11-15:16   ARMZ  cheryl
test      8  sample add       134     02/03/11-15:17   ARMZ  cheryl
test      9  sample add       134     02/03/14-09:52   ARMZ  cheryl
test     10  sample add       134     02/03/14-09:53   ARMZ  cheryl
test     11  sample add       130     02/03/14-10:12   ARMZ  cheryl
test     12  sample add       134     02/03/14-10:14   ARMZ  cheryl
```

Generating a Key Pair and Certificate Request (cmusslgencsr)

Use the **cmusslgencsr** utility to generate a key pair and certificate request. An RSA-type public/private key pair is generated and written to a file in PKCS#8, BER-encoded, base64-encoded format, encrypted with a password.

Note: The Certificate Wizard may be used instead of **cmusslgencsr** to generate a key pair and certificate request. The Certificate Wizard can also create self-signed certificates. The Certificate Wizard works with chained and self-signed certificates. Refer to the *Certificate Wizard Help* for more information.

If you specify **--password**, the user is prompted for a password; otherwise, it uses the default password, “password”. The user is prompted for the following fields: country, state/province,

city/locality, organization, unit in organization, and common name. Common name is the unique host name of the server being authenticated, and must be in a format that can be resolved by all clients with whom the server can secure a session.

Note: If you specify a password other than the default, you must update the FTP, HTTP, EDIINT, SVID, and AUTH daemons using **cmurefresh -S** *private key password*. This command may be placed in **ceustartup**.

The values supplied by the user and the public key are written to a CSR (Certificate Signing Request) file in PKCS#10, DER-encoded, base64-encoded format. Send this file to a certificate authority (CA) (typically by E-mail or the CA's website) to obtain a certificate. The certificate returned by the CA should be in X.509, BER-encoded format. Concatenate the certificate with the PKCS#8 key file mentioned above and reference the resulting file using the **Key-Cert file** parameter in the SPD file and the AS2 configuration file. If a chain of certificate signers is required, other certificates may be appended to this file as well. The first certificate in the file should be the server's and each successive one should be the signer of the one before it.

For additional information, refer to the *Connect:Enterprise UNIX Installation and Administration Guide*.

To generate a public/private key pair, complete the following steps:

1. Select the parameters from the following table. All parameters are optional.

   Parameters can be input using either the abbreviated or long format. The abbreviated format begins with a single hyphen and no space is required between the parameter and the values. The long format begins with two hyphens. You must separate the parameter and values with a space or an equal sign.

Parameter	Description
-c *filename* --csrfile *filename*	specifies a CSR file
-k *filename* --keyfile *filename*	specifies a private key file
-l *length* --length *length*	specifies length of generated key. Default length is 1024.
-p --password	prompts for password used to encrypt private key
-? --help	displays usage message

2. Enter the **cmusslgencsr** command similar to the following example, using the parameters and values determined in step 1. See *cmusslgencsr Example* on page 35 for additional examples using the **cmusslgencsr** command.

```
cmusslgencsr -ccsrfile -kkeyfile -l698
```

or

```
cmusslgencsr --csrfile csrfile --keyfile keyfile --length 698
```

The system responds with prompts for information, displayed on your screen. The dialog looks similar to the following example, with user input shown in bold:

```
# Connect:Enterprise Private Key and CSR Generation Utility  #
# Setting Up  #
# Generating RSA key pair (698-bit)  #
Using default password 'password'
# File 'privkey.txt' contains your encrypted private key  #
# Generating Certificate Signing Request  #
2 Letter Country Code (max 2 characters):  US
State/Province (max 128 characters):  Texas
City/Locality (max 128 characters):  Irving
Organization Name (max 128 characters):  Sterling Commerce
Organizational Unit (max 128 characters):  CSG
Common Name (server host name) (max 128 characters):  host.csg.stercomm.com
#You may submit file 'csr.txt' to you CA to request a certificate  #
```

Verifying Keys and Certificates (cmusslverify)

Use the **cmusslverify** utility to verify the format of the key and certificates and display the certificate or trusted root certificate fields in a human-readable format. Use this utility to review files referenced by the **Key-Cert file** and **Root certificate file** parameters in the SPD file. This utility only works with chained certificates.

Note: The Certificate Wizard may be used instead of **cmusslverify** to verify a key and self-signed or chained certificates. Refer to the *Certificate Wizard Help* for more information.

To verify a public/private key pair and certificate and display it, complete the following steps:

1. Select the parameters from the following table. You must include either **-k** or **-t**.

   Parameters can be input using either the abbreviated or long format. The abbreviated format begins with a single hyphen and no space is required between the parameter and the values. The long format begins with two hyphens. You must separate the parameter and values with a space or an equal sign.

Parameter	Description
-k *filename* --keycert *filename*	specifies a key file
-t *filename* --trusted *filename*	specifies a trusted root file
-? --help	displays usage message

2. Enter the **cmusslverify** command similar to the following example, using the parameters and values determined in step 1.

```
cmusslverify -kkeycert.txt
```

or

```
cmusslverify --keyfile keycert.txt
```

The system responds with a report displayed on your screen, or you can direct it to a file. The report looks similar to the following example. See *Reports* on page 36 for definitions of the fields and more detailed instructions.

```
# Connect:Enterprise Certificate Verification Utility  #
# Setting Up  #

# Key/Certificate: begin  #
Enter password used to encrypt private key:
# Key/Certificate: file format and password verified  #
-----------
cert00:Common Name == host.csg.stercomm.com
cert00:Country == US
cert00:Organization == Sterling Commerce
cert00:Organizational Unit == CSG
cert00:Start Validity Date == Thu Aug 20 11:44:02 1998
cert00:End Validity Date == Fri Aug 20 11:44:02 1999
cert00:Serial Number == 3597aa86
```

Displaying Version Information (cmuvers)

Use the **cmuvers** command to display the version number and option information for your installation of Connect:Enterprise.

To display the version information, complete the following steps:

1. Select the parameters from the following table.

Parameter	Description
-v or -V	outputs abbreviated version number
-? --help	displays usage message
-b	displays build number

2. Enter the **cmuvers** command similar to the following example, using the parameters and values determined in step 1.

```
cmuvers
```

The system responds with a report displayed on your screen, or you can direct it to a file. The report looks similar to the following example.

```
Connect:Enterprise UNIX n.n.nn
Version Lister
Copyright  1994,1999,2006 Sterling Commerce, Inc.
Requires a U.S. export license to send outside the U.S. or Canada.
Connect:Enterprise is a trademark of Sterling Commerce, Inc.
All rights reserved.
U.S. Patent Number 5,734,820

Secure FTP
```

Changing Your Password

If you are a registered user of Connect:Enterprise, you can change your password. Use the **ceupasswd** command with the following parameters:

Parameter	Description
-H --Host	Host name or IP address where authentication service is running (optional if CMUHOST is set).
-c --confirm	Confirm new password.
-h --help	Lists command line options (help).

Parameter	Description
-P --Port	Port number where authentication service is running (optional if CMUPORT is set).
-u --userid	User ID of person executing **ceupasswd**.
-p --password	Password of person executing **ceupasswd**.
-n --new	New password.
-c --confirm	Confirm new password.
-h --help	Lists command line options (help).

Following is an example of changing your own password.

```
$> ceupasswd -u myuserid -p currentpass -n newpass -c newpass
```

Examples

This section provides examples of report and status commands.

cmuadd Examples

The following examples are for **cmuadd**.

After Connect:Enterprise has been loaded, you want to add the files *report1* from your current directory and *report2* from another directory, */reports/monthly*, to the repository for distribution to your branch office in San Francisco (with the mailbox ID SFO).

The most straightforward way of doing this would be:

```
cmuadd -iSFO report1 /reports/monthly/report2
```

If you want to move the reports from the input directories to the repository and delete the input files after the files are added to the repository:

```
cmuadd -iSFO -d report1 /reports/monthly/report2
```

If you want to allow the remote site to request multiple files using a single name, assign a batch ID for the two files called monthly report. To do this:

```
cmuadd -iSFO -b"monthly report" report1 /reports/monthly/report2
```

Still another way of doing this is to pipe the output of another UNIX command to **cmuadd**. This concatenates the reports, assigns them a batch ID, and adds them to the repository for distribution to SFO.

```
cat report1 /reports/monthly/report2 |cmuadd -iSFO -b"monthly report"
```

You can also add a file offline without using a file name, as shown in the following example:

```
cmuadd -iSFO -b"monthly report"
```

The host site has a binary file it intends to transmit to multiple remote sites. The batch is flagged as added, requestable, and multi-transmittable. There are two possible reasons for adding a batch to the repository with the multi-transmittable flag turned on:

- Most often, it is so that the batch is available for transmission to the same remote site multiple times (on multiple auto connects, for example). In this situation, the batch is not flagged as transmitted upon transmission from the host site mailbox. The multi-transmittable and requestable flags remain in effect after transmissions. The only action the host site can take to make this batch unavailable for transmission is to logically delete it with a **cmudelete** command. Without the delete or transmitted flags, a multi-transmittable batch remains permanently eligible for transmission.
- The other reason for setting the multi-transmittable flag is to transmit a batch to every remote in the list for that auto connect. Add the batch to the repository with the **--multxmit** parameter and start an auto connect with the following command.

  ```
  cmuconnect --list=stores --id=BROADCST
  ```

This is the only circumstance when a transmitted flag is posted after transmission of a multi-transmittable batch. To transmit a single batch to multiple remote sites and flag the batch as transmitted, use the optional **--id** switch. Specifying the mailbox ID BROADCST transmits all eligible multi-transmittable batches with that ID to every remote site defined in the auto connect list. If every remote site in the list successfully receives the BROADCST batch, the batch is flagged transmitted.

The BROADCST can be any mailbox ID value used for the **cmuadd**. Usually, the **--id** parameter is not used on the **cmuconnect** command. Then the remote sites listed in the specified auto connect list receive only batches added with each specific mailbox ID. Every remote receives only its batches and no others.

The **cmuconnect** command is detailed in the *Administrator Commands* chapter.

```
cmuadd --id=SMITH --bid="for your eyes only" -o -ce -k
```

To prevent host extraction yet make an EBCDIC batch available for a single transmission to the remote site SMITH, use the **--xmitonce** parameter to indicate a transmit once/transmit only status for the added batch. The batch displays the added, requestable and unextractable flags when the **cmuadd** completes. Unextractable batches are permanently locked against host extraction. After transmitted to a remote site, this batch is flagged non-transmittable and transmitted.

Note: If transmission of this batch fails midway, it is still flagged as non-transmittable, but not as transmitted. This reveals a failed transmission, but this batch cannot be re-transmitted. Once a non-transmittable flag is set, the batch is permanently locked and the host site has to add a duplicate batch to the repository to retry the whole operation.

The **--keepadd** parameter instructs the **cmuadd** utility to retain a **$$ADD** card in the input file without processing it. This **$$ADD** card is transmitted to the remote site ahead of the data. The remote site is also a Connect:Enterprise host site and requires the **$$ADD** card on the incoming batch.

```
cmuadd -imboxid -bnewid -t file1 file2 file3
```

In this example, the host site decides that this batch should automatically go to a remote site upon receipt from another remote (or upon addition to the repository using the offline batch add utilities). The host site administrator must have previously set up the ACD file with the parameters CONTACT=DATA_IMMEDIATE, sendid=mboxid and batchid=newid.

Multi-adding is allowed for auto connects and the **cmuadd** local command as well as for remote connects. You can supply a list of IDs with the **--id** parameter to the **cmuadd** command. Here, three new batches are added in the respective mailboxes (tom, dick and harry) and all three batches will have the same batch ID (accts recvbl), multi-transmit flag (M) and data format flag as ASCII (a). The **cmuadd** report will show three records, one for each batch.

```
cmuadd -i "tom,dick,harry" -b"accts recvbl" -m -ca datafile
```

In this example, more than one input file (file1, file2 and file3) and multiple mailbox IDs (tom, dick, and harry) are specified. A total of 9 batches are added (because each input data file is added into three different mailboxes).

```
cmuadd -i "tom,dick,harry" file1 file2 file3
```

cmudelete Examples

The following examples are for **cmudelete**.

After Connect:Enterprise has been loaded, you want to delete a batch called *report summary* from the repository. You know the batch was sent by your branch office in San Francisco (with the assigned mailbox ID of SFO).

The most straightforward way of doing this would be:

```
cmudelete -iSFO -b"report summary"
```

The batch is logically deleted from the repository.

Say you want to do some housecleaning: everything older than May 1, 2002, no matter who sent it, must be deleted from all mailboxes. To do this, enter the following command:

```
cmudelete -t020501
```

All batches older than this date are logically deleted from the repository. The batches are not physically deleted from the repository until the administrator or an authorized operator issues a **cmuerase** command.

The next example demonstrates the use of **cmudelete** with the log batch (called <<ACTIVITY LOG>>). The command **cmudelete** erases the log batch if used with the **--bid** parameter. Used in this way, **cmudelete** physically erases the log batch, not just sets the D flag. The command below erases the log batch.

```
cmudelete -igeorge -b"<<ACTIVITY LOG>>"
```

Because the **--bid** is required, the following command sets the D flag for all batches in the mailbox george *except* for the log batch. The log batch is not affected.

```
cmudelete -igeorge
```

Note: It is also possible to physically erase the log batch using the **cmuerase** command without specifying the **--bid** parameter.

cmuerase Examples

The following examples are for **cmuerase**.

After Connect:Enterprise has been loaded, you want to erase a batch called “report summary” from the repository. You know the batch was sent by your branch office in San Francisco (with the assigned mailbox ID of SFO) and has been logically deleted using the **cmudelete** command. Now you want to remove it from the repository.

The following example shows how to use the **cmuerase** command to remove the report summary from the repository:

```
cmuerase -iSFO -b"report summary"
```

The batch is physically removed from the repository.

To remove every batch older than May 1, 2002, no matter who sent it, use the command:

```
cmuerase -i"*" -t020501
```

All batches older than this date are erased from the repository.

The **cmuerase** command works the same way on the log batch called <<ACTIVITY LOG>> as with any other batch. It will physically erase them. The following command will physically erase everything in directory george, including the log batch. This is the operation of this command for the host user.

```
cmuerase -igeorge
```

Note: It is also possible to erase a log batch using **cmudelete** with the **--bid** parameter specified.

cmuextract Examples

The following examples are for **cmuextract**.

For example, after Connect:Enterprise has been loaded, you want to extract a batch called “report summary” from the repository. You know they were sent by your branch office in San Francisco (with the assigned mailbox ID of SFO). The batch is extracted from the repository to a file called report.23. You can then read report.23 using any text editor.

```
cmuextract -iSFO -b"report summary" -f report.23
```

If you want to move the reports from the repository to an input directory; (delete the input files after the files are added to the repository), your command would look like the following:

```
cmuextract -iSFO -b"report summary" -d -f report.sum -o
```

The file is extracted from the repository and can be read as *report.sum* using any text editor. Only one batch matching the batch ID is extracted.

Unlike other batches, the log batches that show receipt of data do change whenever corresponding remote users add or extract batches into the repository. You can see its timestamp and size change by issuing **$$DIR**. The log batch shows information similar to the following:

```
/*-------------------------------------------------------------------*/
/*LOG Batch for remote user:  Sales                                  */
/*-------------------------------------------------------------------*/
Thu Jan 2 17:32:35 2002
     Add with protocol=FTP:id=CSG,Bid="Annual Report-1998"
      ....Successfully,total bytes=1680,Batch No.=73

Thu Jan 2 17:32:40 1999
     Add with protocol=ASYNC:id=STERLING,Bid="Personnel Report-1998"
      ....Successfully,total bytes=90182450,Batch No.=98
```

In the following example, the record length of the batch being extracted is 80 and the record separator is 0x1E.

```
cmuextract -b#123 -l80:0x1E
```

cmulist Examples

The following examples are for **cmulist**.

After Connect:Enterprise has been started, if you want to list all the batches added to the repository by your branch office in San Francisco (with the assigned mailbox ID of SFO), input the following command:

```
cmulist -iSFO
```

If you want to see every batch added to the repository between April 15 and May 1, 2002, input the following:

```
cmulist -f020415 -t020501
```

cmustatus Examples

The following examples are for **cmustatus**.

The repository contains a file, reports sent from your branch office in San Francisco (mailbox ID of SFO). Previously you deleted it from the repository (labeling it with a D flag) but now you want to make it accessible to other remote sites for requests.

```
cmustatus -iSFO -b"reports" -xD -oR
```

You want to change every batch added to the repository between April 15 and May 1, 2002 from requestable to transmitted.

```
cmustatus -i"*" -f020415 -t020501 -xR -oT
```

Run **cmulist** to review the results of these changes.

The **cmustatus** command handles the log batch (called <<ACTIVITY LOG>>) in several ways. The following example shows how to change the status of the log batch. The **--bid** parameter is required to change the status of the log batch. The example would change the status of the activity log batch from requestable to transmitted.

```
cmustatus -b"<<ACTIVITY LOG>>" -xR -oT
```

The following will not affect the log batch, because **--bid** is not specified. This command will change every other batch in the mailbox smith from requestable to transmitted.

```
cmustatus -ismith -xR -oT
```

cmusslgencsr Example

The following example is for **cmusslgencsr**.

The **cmusslgencsr** command can be asked to prompt the user for a password.

```
cmusslgencsr -p
```

Connect:Enterprise prompts for the password to be entered and verified within the script.

```
# Connect:Enterprise Private Key and CSR Generation Utility  #
# Setting Up  #
# Generating RSA key pair (1024-bit)  #
Enter password used to encrypt private key:
Verify password used to encrypt private key:
# File 'privkey.txt' contains your encrypted private key  #
# Generating Certificate Signing Request  #
2 Letter Country Code (max 2 characters):  US
State/Province (max 128 characters):  Texas
City/Locality (max 128 characters):  Irving
Organization Name (max 128 characters):  Sterling Commerce
Organizational Unit (max 128 characters):  CSG
Common Name (server host name) (max 128 characters):  host.csg.stercomm.com
#You may submit file 'csr.txt' to you CA to request a certificate  #
```

Date and Time Examples

Here are examples of how the **--end**, **--from**, **--start**, and **--to** parameters operate:

Specifying **--end** 011231 selects batches created on or before December 31, 2001. Specifying **--end** 25 on January 25, 2002 indicates an ending date 25 days ago and selects batches created on or before December 31, 2001.

If [CC]=20 and yy=02, the year is 2002. The entry --start 20011231 --end 20020103 lists all batches created on or between December 31, 2001 and January 3, 2002.

--end 011231:1400 selects batches created on or before December 31, 2001 at 2:00 pm. **--end** 011231/1400 selects batches created on or before December 31, 2001 at or before 2:00 pm each day. **--end** 1400 specifies that all batches created on or before 2:00 pm today are selected.

Reports

Reports are the response from Connect:Enterprise whenever a command is executed. They are displayed on your screen unless you redirect them to a file. The advantage of a file is that it can be accessed at any time. To redirect to a file, use the following standard UNIX format:

```
cmucommand > filename
```

where *command* = **add**, **delete**, **erase**, **extract**, **list**, or **status**.

The fields of the output reports are explained in the following table:

Field	Description
Batch #	The batch number assigned to the batch.
Batch ID	The 1–64 character batch ID.
Batch Status Code Values	A summary of the status codes and their meanings.
Bytes	The size of the batch in bytes.
Command Line parameters	The parameters used for this command.
Date	Date the report was issued.
Date-Time	The date and time the batch was successfully added.
File Name	The file name is listed. If this information exceeds the *-cols* value of 80 characters, the file name is wrapped to the next line of the report. The default is 79 characters.

Field	Description
File Size	The size of the file in bytes.
Logically Deleted Batches	The batches that were affected by **cmuerase**.
Mbx ID	Mailbox ID where the batch was added.
Number of Batches Added	The number of batches added.
Number of Input Files	The number of input files processed.
Number of Files Bypassed	The number of input files bypassed.
Number of Input Bytes	The number of bytes of input data read.
Org ID	The Origin ID of the remote site that transmitted the batch (Async or Bisync **$$ADD** or FTP **put**) or the ID of a host site user that added it with **cmuadd**.
Page	Report page number.
PR Status	File processing status. Options are: Processed, indicates success Bypassed, indicates failure. Input file read failures of any kind result in a bypassed status. The reason for failure is displayed. For example, *bypassed file not found* is displayed for a nonexistent input file.
Status	These are the current flags for the batch. The reports generated by the command line utilities use single-character flags for data format and protocol status. See *Flags* on page 6 for definitions of these flags.
Time	Time the report was issued.
Processing Message Values	After all batches have been extracted, the message descriptions for the codes reported are listed. Codes: Processing Message Values 0: Extract Failed 11: Batch bypassed due to transparent data 12: Batch bypassed due to incomplete data 13: Requested incomplete batch extracted 14: Batch bypassed due to delete flag 15: Requested deleted batch extracted 18: Batch Re-Extracted (warning) 98: Batch Bypassed 99: Extract Okay

Chapter 2

Operator Commands

Operator commands are used to control the fundamental operations of Connect:Enterprise. Use the command line utilities to do the following tasks:

- Trace Connect:Enterprise activity (**ceutrace**)
- Display the auto connect queue and query the status or change the priority of individual auto connect entries (**ceuacq**)
- Delete the auto connect queue (**ceuqdel**)
- Start an auto connect (**cmuconnect**)
- Update Connect:Enterprise to include any ACD file changes or communicate the private key password (**cmurefresh**)
- Show the status of the Connect:Enterprise system (**cmusession**)
- Start a communications resource (**cmustart**)
- Stop a repository operation (**cmustop**)
- Trace repository operations (**cmutrace**)

ceushutdown, which is used to shut down Connect:Enterprise, is also classified as an operator command. This command is described in the *Connect:Enterprise UNIX Installation and Administration Guide*.

Note: The **cmureport** command can be issued by the local user, the operator, or the administrator. Refer to *Connect:Enterprise UNIX Installation and Administration Guide* for more information about the **cmureport** command.

Tracing Connect:Enterprise Activity (ceutrace)

You can use the dynamic tracing command (**ceutrace**) to turn tracing on and off and to set tracing levels without restarting Connect:Enterprise. You can also use **ceutrace** to view the current settings. Refer to the following procedures in this section:

- *Turning Tracing On* on page 40
- *Daemon Considerations* on page 43

- *Locating Your Trace Files* on page 44
- *Turning Tracing On* on page 40
- *Clearing and Restarting Your Trace Files* on page 45

Turning Tracing On

Use the following procedure to turn tracing on:

1. Select the appropriate parameters from the following table. The required parameters are in bold.

   All parameters may be input using either the single hyphen or double hyphen format. The parameters may be separated from their associated values by a space or the values may immediately follow.

Parameter	Description
-o --on	Turn trace on Turns tracing on at the target daemons and displays the current trace settings. If –on is specified for a target daemon that is not actively tracing, a trace file is opened according to its current filename prefix.
-x --off	Turn trace off Setting –on or –off will turn tracing on or off at the target daemon(s) and display the current trace settings. Setting neither causes the display of current settings. If –off is sent to a daemon that is actively tracing, it will turn off tracing and close its trace file. If –on is sent to a daemon that is not actively tracing, it will open a trace file according to its current filename prefix. –on and –off are mutually exclusive. If both are specified an error results.
-H *hostname* --host *hostname*	Host computer where the control daemon was started.
-P *portno* --port *portno*	Port number where the control daemon was started.
-p *password* --passwd *password*	Connect:Enterprise password
-d *daemon names* --daemon *daemon names*	Direct the command to those daemons that match the daemon name patterns. If not specified, the command applies to all daemons. The daemon name pattern string may be a combination of File Name Matching patterns, separated by commas. For example, if you have multiple FTP daemons you can trace for all FTP daemons by specifying: ceutrace –d “FTP*” –on Refer to *Daemon Considerations* on page 43.

Parameter	Description
-m on|off --mailbox on|off	Automatically starts mailbox tracing for calls that the specified protocol daemon initiates. This parameter is only available when used with protocol daemons. This affects the mailbox daemon tracing state only when the mailbox daemon was previously not tracing, or was tracing at a lower trace level than the protocol daemon. It cannot cause the mailbox daemon to lower its trace level. The parameter takes an “on” or “off” value to indicate whether mailboxalso processing is to be enabled or disabled.
-f --*filename*	Trace filename prefix Adjusts the file name prefix of the trace file. It overrides the –d filename if specified on the target daemon command line startup. If the target name differs from the current, the old file is closed, and the new one is opened. You can only use this parameter if there is only one target daemon for the command (for example, –daemon FTP).
-l --*level*	Specifies the trace level from 0 to 100 at which the target daemons trace. It overrides the –l level if specified on the target daemon command line startup. Setting a trace level does not imply turning traces on. They must be turned on with the –on parameter. The trace levels are as follows, with each higher number including all tracing for lower numbers. Note: Tracing will affect performance. Do not specify levels higher than 10 unless instructed by Sterling Commerce. 0 High level messages and errors only 1–2 Logon / Logoff activity, statistics 3–4 FTP command traffic, simple program logic flow 5–8 One-line inter-process Communication (SIPS) flow, more program flow 9 SIPS records dumped (first 176 bytes) 10–49 More exhaustive debugging 50-98 Full SIPS records dumped 99 All known debug output 100 Full SIPS records dumped (field by field) The Java daemons (cmuhttp, cmuediintd, and cmuadmind) have the following log4j logging levels: =0 off—No logging <=5 sparse—Errors only <=9 moderate—Errors and warnings only <=50 detailed—Includes infrmational <=99 verbose—Includes trace data

Parameter	Description
-a *--account*	Account names to trace. List multiple account names in a command separated list. You can also match file name patterns and you can use an exclamation (!) to exclude. For example, to trace the remotes RMT1, RMT2, … RMT10, but not RMTTEST, turn on FTP tracing for all but RMTTEST by specifying: ceutrace –d FTP –on –account “RMT*,!RMTTEST”
-L *--listname*	Autoconnect list names to trace. List multiple autoconnects in a command separated list. You can also match file name patterns and you can use an exclamation (!) to exclude. For example, a to trace the list names of LIST1.acd, LIST2.acd, … LIST10.acd, but not LIST10.acd, turn on FTP tracing for just the LIST2.acd-LIST9.acd list names by specifying: ceutrace –d FTP –on –L “LIST*,!*10*”
-D --delimited	Specifies to return pipe delimited output.

2. Enter the **ceutrace** command similar to the following example, using the parameters and values determined in step on page -40.

```
ceutrace -on -d “FTP*” -m -19
```

3. The status of each daemon is displayed. Following is an example:

```
              Command Line Parameters:
                        ceutrace
Name          SID      Trace  Level   Filename
                       Status         Prefix

CONTROL       1         Off     0      CONTROL.out
SVD           8         Off     0      SVD.out
MAILBOX       3         Off     0      MAILBOX.out
EDIINT        16        Off     0      cmuediintd.out
ASYNCD        14        Off     0      ASYNCD.out
HTTP          13        Off     0      cmuhttpd.out
SSHFTP        12        Off     0      SSHFTP.out
FTP2          11        On      9      FTP2.out
FTP           10        On      9      FTP.out
ADMIN         9         Off     0      cmuadmind.out
EXITS         5         Off     0      EXITS.out
SYSLOG        2         Off     0      SYSLOG.out
ACD           4         Off     0      ACD.out
AUTH          6         Off     0      AUTH.out
```

The following example shows the output with the -D option specified. The column correspond to the columns in standard output:

```
CONTROL|1|Off|0|CONTROL.out
SVD|8|Off|0|SVD.out
MAILBOX|3|Off|0|MAILBOX.out
EDIINT|16|Off|0|cmuediintd.out
ASYNCD|14|Off|0|ASYNCD.out
HTTP|13|Off|0|cmuhttpd.out
SSHFTP|12|Off|0|SSHFTP.out
FTP2|11|On|9|FTP2.out
FTP|10|On|9|FTP.out
ADMIN|9|Off|0|cmuadmind.out
EXITS|5|Off|0|EXITS.out
SYSLOG|2|Off|0|SYSLOG.out
ACD|4|Off|0|ACD.out
AUTH|6|Off|0|AUTH.out
```

The trace output is written to one or more files. Refer to *Locating Your Trace Files* on page 44.

Daemon Considerations

When specifying daemons to trace on, you must indicate the name of the daemon with the -d option. Following is a list of default daemon names. If these defaults have changed, you can look at the startup script for current values:

Daemon	Default Name
CONTROL	cmuctld, mailbox control server
AUTH	cmuauthd, authentication server
SVD	cmusvid, service interface server
MAILBOX	cmumboxd, mailbox server
ASYNCD	cmuasyd, async daemon
BISYNC	cmubscdc, bisync daemon
EDIINT	cmuediintd, ediint daemon
HTTP	cmuhttpd, http daemon
SSHFTP	cmusshftpd, ssh daemon
FTP	cmuftpd, ftp daemon
ADMIN	cmuadmind, admin daemon
EXITS	cmuexitd, exits server
SYSLOG	cmulogd, log server
ACD	cmuacd, autoconnect server

With ceutrace, each daemon is managed differently by Connect:Enterprise. Consider the following when tracing the activity of a daemon.

- Business processes—The ceutrace command does not work with the business process daemon. The trace must be set at startup using ceustartup.trace.
- The ceutrace command does not affect the tracing status of child processes for the control daemon, authentication server daemon, mailbox daemon, log daemon, exit daemon, service interface daemon, FTP daemon, SSH daemon.
- The ceutrace command affects the tracing status of child processes of the autoconnect daemon.
- The cmusvid children created to serve a particular GUI user session are active for the life of that session and are not affected by the trace status changes during their lifetimes.
- Async and bisync remote connect child processes run continuously on a given port and do not end after each Remote Connect session. They when an autoconnect request is run on the same port. Any changes to tracing status specified using the ceutrace command for the master async or bisync daemon only take effect for the child processes on a given port when an autoconnect request is run on that port.
- Async and bisync remote connect sessions now have unique session IDs.

Locating Your Trace Files

Connect:Enterprise creates a trace file for each daemon. The name of the files are generated as follows for all master daemons: <*prefix*>.<*pid*>, where <*prefix*> is the trace filename prefix specified on the command line for the master daemon using the –d parameter and <*pid*> is the process ID assigned to the daemon by the operating system.

The name of the trace files for slave and child processes is generated as follows:

Master Daemon	**Child or Slave Daemon Trace File Name**
Auto Connect Daemon (cmuacd)	<*prefix*>.SLV.<*pid*>
Async Daemon (cmuasyd)	Trace files for remote connect sessions are initially created as: <*prefix*>.<*port-name*>.<*pid*>. Once a session is established the file is renamed to <*prefix*>.<*port-name*>.<*account*>.<*session-ID*>.<*pid*>
FTP Daemon (cmuftpd)	For remote connections: <*prefix*>.*svr.*<*account*>.<*sessionID*>.<*pid*> For autoconnects: <*prefix*>.clt.<*account*>.<*sessionID*>.<*pid*>
SSH Daemon (cmusshftp)	<*prefix*>.<*direction*>.<*function*>.<*account*>.<*sessionID*>.<*pid*>
Bisync Daemon (cmubscda, cmubscdc)	Trace files for remote connect sessions are initially created as: <*prefix*>.<*port-name*>.<*pid*>. Once a session is established the file is renamed to <*prefix*>.<*port-name*>.<*account*>.<*session-ID*>.<*pid*>

Clearing and Restarting Your Trace Files

To manage the size of your trace files, you will need to periodically start new trace files. You have two options:

- If you want to save old trace files, you can rename the files in the trace directory and reissue the trace command using *Turning Tracing On* on page 40.

If you do not want to save old trace files, reissue the trace command using *Turning Tracing On* on page 40.

Displaying the Auto Connect Queue Entries (ceuacq)

The **ceuacq** command displays the current entries in the auto connect queue and allows users to query the status and change the priority of individual entries. The complete queue listing provides an indication of the work load present on the system. The auto connect queue is a persistent database to process and monitor active auto connect requests. Inactive requests are not represented in the queue. The auto connect queue is responsible for executing auto connects using the available system resources in the most efficient manner.

The auto connect queue is a persistent database designed to withstand system interruptions. The ACD master process performs queue recovery at startup, which corrects certain situations that may exist in the queue in the event of a system interruption. Queue entries that were connected at the time of the interruption are reset so the session can be reconnected.

To access the auto connect queue, complete the following steps:

1. Select the parameters from the following table as needed. There are no required parameters.

Note: Only the abbreviated format of the parameters is available for this command.

The parameters may be separated from their associated values by a space or the values may immediately follow, without separation.

Parameter	Description
-A *nnnnnnnn*	specifies the reference number of an ACD entry
-a *filename*	specifies the file name of an ACD entry
-l *nn*	specifies the new priority level of the indicated entries
-Q	indicates not to print field headers
-R *nnnnnnnn*	specifies the reference number of an RSD entry
-r *filename*	specifies the file name of an RSD entry
-t	indicates to use the tab character as a field delimiter

Parameter	Description
-v	indicates to display in full verbose mode
-?	displays usage message

2. The **ceuacq** command should be written similar to the following example, using the parameters and values determined in step 1. See *Examples* on page 61 for additional examples using the **ceuacq** command.

```
ceuacq
```

The system will respond with a report displayed on your screen, or you can direct it to a file. The report will look similar to the following example.

```
ACD Name          Ref No     SID   Stat   Pri   Date    Time   Err Stats
--------          ------     ---   ----   ---   ----    ----   ---------
DrJohnson.acd     00000001   001   ACTV   05    021201  1535   00  02
DrWarren.acd      00000004   004   ACTV   05    021201  1545   00  00
DrAddams.acd      00000005   005   ACTV   05    021201  1546   00  00
TomayClinic.acd   00000011   011   WAIT   07    021201  1546   00  00
```

The following table explains each variable.

Field	Description
ACD Name	Name of the associated ACD file that governs the auto connect
Ref No	Identifies the auto connect session within the queue only. This number is unique within an ACD or RSD.
SID	Session ID
Stat	Current status of the auto connect
Pri	Priority setting
Date	Date the auto connect was added to the queue
Time	Time the auto connect was added to the queue
Err Stats	First number - Total number of requeues attempted for the RSD or ACD Second number - Total number of RSD requests that failed and will not complete for an ACD

Deleting the Auto Connect Queue Database (ceuqdel)

The **ceuqdel** command deletes the auto connect queue database. Use this command only if the database becomes corrupt and prevents the Auto Connect daemon from starting up. The ACD master process automatically recreates the queue database files at startup if they do not exist.

Note: **ceuqdel** should only be performed after issuing the **ceushutdown** command. After **ceuqdel** is complete, a new database is created during **ceustartup**.

To delete the auto connect queue, complete the following steps:

1. Enter the following command:

```
ceuqdel
```

   The system responds with the following prompt:

```
ACD queue database location: /usr/cmuhome/database/acdqueue

Do you really want to delete the ACD queue database? (Y or N)
```

2. Type **Y** to delete the database files.

Initiating an Auto Connect (cmuconnect)

The **cmuconnect** command triggers a local site-initiated auto connect function. The auto connect process has a number of options and restrictions.

To initiate an auto connect, complete the following steps:

1. Select the parameters from the following table as needed. Required parameters are in bold.

   All parameters can be input using either the abbreviated or long format.

   The abbreviated parameters, those beginning with a single hyphen, may be separated from their associated values by a space or the values may immediately follow, without separation.

   The long parameters, those beginning with two hyphens, *must* be separated from their associated values with either a space or an equal sign.

Parameter	Description
-L *listname* **--list *listname***	**ACD file to be started**

Parameter	Description
-B *nnn* --block *nnn*	number of records per block for data outbound to a Bisync remote
-b *batchid* --bid *batchid*	specifies a user batch id
-C *y*|*n* --compress *y*|*n*	specifies Bisync compression options
-c *a*|*e*|*n* --conv *a*|*e*|*n*	overrides the translation type
-D *business process ID* --bpid *business process ID*	overrides the business process ID that the ACD notifies.
-d *daemon:port* --resource *daemon:port*	directs an auto connect to a specific session managing daemon
-H *hostname* - -host *hostname*	host computer where cmuctld started
-I *nnn* --interval *nnn*	specifies the time interval to wait before reconnecting
-i *mboxid* --id *mboxid*	changes the method of batch selection for the auto connect list
-m *sr*|*so*|*rs*|*ro* --mode *sr*|*so*|*rs*|*ro*	overrides transfer mode
-o --onebatch	transmits the first batch that meets the criteria
-P *portno* --port *portno*	port number where cmuctld started
-p *password* --passwd password	Connect:Enterprise password
-R *nn* --retry nn	number of retry attempts
-s *n*|*1*|*2*|*3*|*4* --batchsep *n*|*1*|*2*|*3*|*4*	determines the transmission format
-t *y*|*n* --trunc *y*|*n*	trailing blanks removed
-u *username* --user *username*	name other than UNIX login
-v --verbose	displays session activity
-? --help	displays usage message

2. Enter the **cmuconnect** command similar to the following examples, using the parameters and values determined in step 1. See *Examples* on page 61 for additional examples using the **cmuconnect** command.

```
cmuconnect -Lwcoast
```

or

```
cmuconnect --list wcoast
```

Updating the Auto Connect Lists (cmurefresh)

The **cmurefresh** command instructs Connect:Enterprise to reexamine all auto connect lists and update that information. Issue this command each time the administrator adds a new ACD file or modifies an existing one. **cmurefresh** will send a message to the auto connect daemon, **cmuacd**, telling it to reread the entire contents of all ACD files.

To update the auto connect list, complete the following steps:

1. Select the appropriate parameters from the following table. All parameters are optional.

   All parameters can be input using either the abbreviated or long format.

   The abbreviated parameters, those beginning with a single hyphen, may be separated from their associated values by a space or the values may immediately follow, without separation.

   The long parameters, those beginning with two hyphens, *must* be separated from their associated values with either a space or an equal sign.

Parameter	Description
-e --ediintrefresh	refresh AS2 contracts and proxy definitions. This option should only be used when cmurefresh is executed on the Connect:Enterprise UNIX repository host.
-H *hostname* --host *hostname*	host computer where cmuctld started
-N --ftpdname	name of FTP, HTTP, EDIINT, SVID, or SSHFTP daemon
-P *portno* --port *portno*	port number where cmuctld started
-p *password* --passwd *password*	Connect:Enterprise password
-S *password* --sslpass *password*	password used to decrypt the private key used for SSL (FTP, HTTP, SVID, AUTH) and S/MIME (EDIINT).

Parameter	Description
-s *passphrase* --sshpass *passphrase*	passphrase for the SSHFTP hostkey
-u *username* --user *username*	user name other than UNIX login
-? --help	displays usage message

2. Enter the **cmurefresh** command similar to the following example, using the parameters and values determined in step 1. See *Examples* on page 61 for additional examples using the **cmurefresh** command.

```
cmurefresh -Hmachine_6
```

Updating the Private Key Password (cmurefresh)

You can also use the **cmurefresh** command to communicate the private key password to the FTP daemon for Secure FTP, the HTTPS daemon for HTTP, the EDIINT daemon for S/MIME, the SSHFTP daemon for SSH2 (SFTP and SCP). For FTP and HTTP daemons, this updates the password used for SSL. For EDIINT daemons, this updates the private key passwords for signing and encryption.

To send the private key password information, complete the following steps:

1. Select the appropriate parameters from the following table.

   All parameters can be input using either the abbreviated or long format.

   The abbreviated parameters, those beginning with a single hyphen, may be separated from their associated values by a space or the values may immediately follow, without separation.

   The long parameters, those beginning with two hyphens, *must* be separated from their associated values with either a space or an equal sign.

Parameter	Description
-e --ediintrefresh	refresh AS2 contracts and proxy definitions.
-H *hostname* --host *hostname*	host computer where cmuctld started
-N --ftpdname	name of FTP, HTTP, EDIINT, SVID, or SSHFTP daemon. If no daemon is specified, password will be sent to all daemons.

Parameter	Description
-P *portno* --port *portno*	port number where cmuctld started
-p *password* --passwd *password*	Connect:Enterprise password
-S *password* --sslpass *password*	password used to decrypt the private key
-s *password* -ssh password *password*	Passphrase for SSHFTP host key.
-u *username* --user *username*	user name other than UNIX login
-? --help	displays usage message

2. Enter the **cmurefresh** command similar to the following example, using the parameters and values determined in step 1.

```
cmurefresh --sslpass my_pass -N FTP02
```

Updating the AS2 Configuration (cmurefresh)

The **cmurefresh** can also be used to refresh the AS2 contract information and the AS2 proxy configuration. Enter the command similar to the following example.

To show usage information include -? --help.

```
cmurefresh -e
```

or

```
cmurefresh --ediintrefresh
```

To refresh the AS2 port configuration, you must restart the HTTP daemon using **ceustartup**. Refer to *Starting and Stopping Connect:Enterprise* in the *Connect:Enterprise UNIX Installation and Administration Guide*.

Note: The **cmurefresh -e** command updates 2 fields in the AS2 port configuration: **Public IP address or host name** and **Translated listening port**. These fields define the URL that asynchronous MDN are sent to.

Showing the System Status (cmusession)

The **cmusession** utility produces a formatted report on the status of all communications and Auto Connect daemon activity.

To show the system status, complete the following steps:

1. Select the parameters from the following table as needed. All parameters are optional.

   All parameters can be input using either the abbreviated or long format.

   The abbreviated parameters, those beginning with a single hyphen, may be separated from their associated values by a space or the values may immediately follow, without separation.

   The long parameters, those beginning with two hyphens, *must* be separated from their associated values with either a space or an equal sign.

Parameter	Description
-A --all	lists all activity
-a --auto	lists auto connect activity
-b --bsc	lists Bisync daemons
-f --ftp	lists FTP sessions
-h --http	lists HTTP and HTTPS sessions
-e --ediint	lists EDIINT (AS2) sessions
-g --busprocess	lists GIS sessions
-H *hostname* --host *hostname*	host computer where cmuctld started
-M --more	prevents scrolling
-m --mailbox	displays batches currently in transit, current batch size, mailbox ID, and batch ID.
-n *daemonname* --name *daemonname*	resource name for session report
-P *portno* --port *portno*	port number where cmuctld started

Parameter	Description
-p *password* --passwd *password*	Connect:Enterprise password
-q --async	lists Async daemons
- s --sshftp	shows SSHFTP sessions
-u *username* --user *username*	user name other than UNIX login
-X *nnnn* --cols *nnnn*	number of columns available for display
-Y *nnnn* --rows *nnnn*	number of rows available for display
-? --help	displays usage message

2. Enter the **cmussession** command similar to the following examples, using the parameters and values determined in step 1. See *Examples* on page 61 for additional examples using the **cmusession** command.

```
cmusession -f
```

or

```
cmusession --ftp
```

The system will respond with a report displayed on your screen, or you can direct it to a file. The report will look similar to the following example.

```
---------------------------------------------------------------------
Date:  yy/mm/dd    Connect:Enterprise UNIX n.n.nn          Page 0001
Time:  hh:mm:ss    Session Utility


Command Line Parameters:
     cmusession


Name      Type    Host      PID    RmtID Resource   State        SID
xxxxxxxx  xxxxxx  xxxxxxxx  nnnnn  n     xxxxxxxxxx  xxxxxxxxxxx  nnn
xxxxxxxx  xxxxxx  xxxxxxxx  nnnnn  n     xxxxxxxxxx  xxxxxxxxxxx  nnn
xxxxxxxx  xxxxxx  xxxxxxxx  nnnnn  n     xxxxxxxxxx  xxxxxxxxxxx  nnn
xxxxxxxx  xxxxxx  xxxxxxxx  nnnnn  n     xxxxxxxxxx  xxxxxxxxxxx  nnn
xxxxxxxx  xxxxxx  xxxxxxxx  nnnnn  n     xxxxxxxxxx  xxxxxxxxxxx  nnn
xxxxxxxx  xxxxxx  xxxxxxxx  nnnnn  n     xxxxxxxxxx  xxxxxxxxxxx  nnn
xxxxxxxx  xxxxxx  xxxxxxxx  nnnnn  n     xxxxxxxxxx  xxxxxxxxxxx  nnn
xxxxxxxx  xxxxxx  xxxxxxxx  nnnnn  n     xxxxxxxxxx  xxxxxxxxxxx  nnn


Max. Concurrent Sessions: nnn
---------------------------------------------------------------------
```

The following table explains each variable.

Field	Description
Date	Date the command was executed.
Time	Time the command was executed.
Page	Report page number.
Command Line Parameters	The parameters used for this execution of **cmusession.**
Name	Name of the daemon.
Type	Master or Child daemon.
Host	Host system where the daemon is executing.
PID	Process ID of the daemon.
RmtID	For remote connect sessions, this field shows the user that is logged on. For autoconnect sessions, this field shows the RSD name.

Field	Description
Resource	For Async daemons: the Async device name For ARTIC Bisync daemons: the Bisync card#, port# For Cleo Bisync daemons: the device name For FTP daemons: the FTP listening port For HTTP daemons: the port where AS2 listens for HTTP and HTTPS requests For SSHFTP daemons: the SSHFTP listener port. For BP resources, no information appears in this column. For the ADMIN daemon and the WebDAV: the port the ADMIN daemon and WebDAV services are listening on.
State	For Communications daemons: Idle, Connecting, Send batch, Receive batch, Wait for Connect, Disconnecting, Mailbox List, Mailbox Delete, Mailbox Receiving, Mailbox Status, Mailbox Sending, Stopped, Auto Connect, Processing Startup - The port is in a startup or initialization state. This occurs when the product or protocol master daemon is started. Restarting - The master daemon is attempting to restart services on that port. This usually occurs after a scheduled transfer when the port is reinitialized to accept incoming calls. Killed - The communication session and its process were killed. This can occur when an auto connect (scheduled transfer) attempts to start up on a port. Dead - The port is in a nonrecoverable state and is considered offline. This indicates that there is a hardware or configuration problem and the port could not be initialized. For Auto Connect daemon: no output is displayed.
SID	Session ID (useful for **cmustart** and **cmustop**)

Starting a Communications Resource (cmustart)

The **cmustart** program is used to activate a closed communications line and/or session. Reactivation is done by specifying the session id of the line or session to be activated, and the name of the daemon that is controlling the line or session.

To start a communications resource, complete the following steps:

1. Select the parameters from the following table as needed. All parameters are optional.

   All parameters can be input using either the abbreviated or long format.

   The abbreviated parameters, those beginning with a single hyphen, may be separated from their associated values by a space or the values may immediately follow, without separation.

The long parameters, those beginning with two hyphens, *must* be separated from their associated values with either a space or an equal sign.

Parameter	Description
-H *hostname* --host *hostname*	host computer where cmuctld started
-P *portno* --port *portno*	port number where cmuctld started
-p *password* --passwd *password*	Connect:Enterprise password
-s *sessid* --sid *sessid*	session ID associated with a physical resource
-u *username* --user *username*	user name other than UNIX login
-? --help	displays usage message

2. Enter the **cmustart** command similar to the following example, using the parameters and values determined in step 1. See *Examples* on page 61 for additional examples using the **cmustart** command.

```
cmustart -s297
```

or

```
cmustart --sid 297
```

The system will respond with a report displayed on your screen, or you can direct it to a file. The report will look similar to the following example.

```
Session ID = nnn started at ddd mmm DD hh:mm:ss [CC]yy
```

The following table explains each variable.

Variable	Definition
nnn	Session ID
ddd	Day of the Week (for example, Wed)

Variable	Definition
mmm	Month (for example, Dec)
DD	Date (for example, 07)
hh	Hours
mm	Minutes
ss	Seconds
[CC]yy	Year

Stopping an Auto Connect, Comm Daemon, or Child Process of a Comm Daemon (cmustop)

The **cmustop** program is used to stop a communications daemon, a child process spawned by a communications daemon, or an auto connect session. Each communication daemon spawns a child process to manage each physical device the parent daemon has defined to it as a resource. The **cmusession** command can be used to display the session IDs for a communications daemon's ports (Bisync) or devices (Async).

With **cmustop**, the associated communications master daemon signals the associated slave process to enter a wait loop until a **cmustart** is issued. This command is useful in removing a physical port from service. Normally, **cmustop** would be issued against a child daemon's session ID. If issued against a Master daemon's session ID, the daemon is shutdown and must be restarted as it is when the **ceustartup** shell script invokes it.

When **cmustop** is issued against the session ID of a *comm* daemon parent, the parent and all of its children are killed immediately without waiting for active ports to finish current sessions. The **cmustart** command cannot be used to restart the parent communication daemon. It must be started with the **ceustartup** script.

The **cmustop** command can only stop a child *comm* daemon when the daemon is between sessions. If the daemon is active, the command is queued until the current session completes on that port. This is true for auto connect sessions and remote connect sessions. Later, **cmustart** can be used to bring that port back into service without cycling the parent daemon.

When **cmustop** is issued against the session ID of an auto connect in progress, any remotes not yet connected are not dialed, but sessions already started are not affected. The **cmustart** command cannot be used to resume auto connect processing of an auto connect that was stopped. A **cmuconnect** command must be issued instead.

To issue the **cmustop** command, complete the following steps:

1. Select the parameters from the following table as needed. All parameters are optional.

   All parameters can be input using either the abbreviated or long format.

The abbreviated parameters, those beginning with a single hyphen, may be separated from their associated values by a space or the values may immediately follow, without separation.

The long parameters, those beginning with two hyphens, *must* be separated from their associated values with either a space or an equal sign.

Parameter	Description
-H *hostname* --host *hostname*	host computer where cmuctld started
-P *portno* --port *portno*	port number where cmuctld started
-p *password* --passwd *password*	Connect:Enterprise password
-s *sessid* --sid *sessid*	session ID associated with a physical resource
-u *username* --user *username*	user name other than UNIX login
-? --help	displays usage message

2. Enter the **cmustop** command similar to the following examples, using the parameters and values determined in step 1. See *Examples* on page 61 for additional examples using the **cmustop** command.

```
cmustop -s232
```

or

```
cmustop --sid 232
```

The system will respond with a report displayed on your screen, or you can direct it to a file. The report will look similar to the following example.

```
Session ID = nnn stopped at ddd mmm DD hh:mm:ss [CC]yy
```

The following table explains each variable.

Variable	Definition
nnn	Session ID
ddd	Day of the Week (for example, Wed)

Variable	Definition
mmm	Month (for example, Dec)
DD	Date (for example, 07)
hh	Hours
mm	Minutes
ss	Seconds
[CC]yy	Year

Tracing Communication Sessions (cmutrace)

The **cmutrace** command is used to enable or disable traces for communications sessions that use the ARTIC bisync daemon. The **cmutrace** program simply communicates to the COMM daemon (cmubscda) that traces are desired for the specified session ID.

To start or stop a trace, complete the following steps:

1. Select the parameters from the following table as needed. Required parameters are in bold.

   All parameters (required or optional) can be input using either the abbreviated or long format.

   The abbreviated parameters, those beginning with a single hyphen, may be separated from their associated values by a space or the values may immediately follow, without separation.

   The long parameters, those beginning with two hyphens, *must* be separated from their associated values with either a space or an equal sign.

Parameter	Description
-o **--on**	**trace turned on**
-x **--off**	**trace turned off**
-s *sessid* **--sid** *sessid*	**session ID associated with a physical resource**
-H *hostname* --host *hostname*	host computer where cmuctld started
-P *portno* --port *portno*	port number where cmuctld started
-p *password* --passwd *password*	Connect:Enterprise password

Parameter	Description
-u *username* --user *username*	user name other than UNIX login
-? --help	displays usage message

2. Enter the **cmutrace** command similar to the following examples, using the parameters and values determined in step 1. See *Examples* on page 61 for additional examples using the **cmutrace** command.

```
cmutrace -x -s921
```

or

```
cmutrace --off --sid 921
```

ARTIC Bisync Communications Daemon (cmubscda)

To use **cmutrace** with the ARTIC Bisync communications daemon, the daemon must be started in debug mode.

1. Execute a **cmusession** command, identify the card and port you want to trace, and note the session ID for the card and port.
2. Execute the **cmutrace** command as follows to turn tracing on for the port:

```
cmutrace -ssid --on
```

The **-s** argument specifies the Session ID (*sid*) obtained from the **cmusession** command output.

3. Run the session you want to trace.
4. Execute the **cmutrace** command as follows to turn tracing off for the port:

```
cmutrace -ssid --off
```

The **-s** argument specifies the Session ID (*sid*) obtained from the **cmusession** command output.

Cleo Bisync Communications Daemon

The **cmutrace** command generates no output for Bisync sessions using Cleo SYNCcable+ hardware. To obtain a simulated line trace, use the CLEOCMD="-M monitorfile" command in the CPD file.

Async Communications Daemon

The **cmutrace** command generates no output for Async session IDs.

FTP Communications Daemon

The **cmutrace** command generates no output for FTP session IDs.

HTTP Communications Daemon

The **cmutrace** command generates no output for HTTP session IDs.

EDIINT Communications Daemon

The **cmutrace** command generates no output for EDIINT session IDs.

ADMIN Communications Daemon

The **cmutrace** command generates no output for ADMIN session IDs.

WebDAV Communications Daemon

The **cmutrace** command generates no output for WebDAV session IDs.

SSHFTP Communications Daemon

The **cmutrace** command generates no output for SSHFTP session IDs.

Examples

This section provides examples of some operator commands.

ceuacq Examples

To view the full detailed queue entry for an ACD with reference number 00000001, use this command:

```
ceuacq -A -v 00000001
```

The system responds with the following report:

```
ACD Name: DrJohnson.acd  SID: 001
  Ref#: 00000001  Stat: ACTV  Pri: 05  DT: 021201 1535  Total Err Stat: 00 02
    RSD Name:rscfile1.xxx
      Ref#: 00000001  Stat: ACTV  Pri: 05  DT: 021201 1535  Total Err Stat: 00 01
    RSD Name:rscfile2.xxx
      Ref#: 00000002  Stat: WAIT  Pri: 05  DT: 021201 1535  Total Err Stat: 00 01
    RSD Name:rscfile3.xxx
      Ref#: 00000004  Stat: ACTV  Pri: 05  DT: 021201 1535  Total Err Stat: 00 00
```

To change the priority of a queue entry by the ACD file name DrJohnson.acd to level 03, use this command:

```
ceuacq -a DrJohnson.acd -l 03
```

cmustart Example

This example restarts a stopped daemon with session ID 297:

```
cmustart -s297
```

cmuconnect Examples

To start a full auto connect for the auto connect list named WCOAST in the *$CMUHOME/acd* directory, use this command:

```
cmuconnect -Lwcoast
```

or

```
cmuconnect --list wcoast
```

To start an auto connect for the batch named *payroll* for all remote sites in the WCOAST file, use this command:

```
cmuconnect -Lwcoast -bpayroll
```

or

```
cmuconnect --list wcoast --bid payroll
```

cmurefresh Examples

To refresh the system so that it recognizes a new host user password, enter the following command:

```
cmurefresh
```

To communicate with the Control Daemon on another computer that has a control IP address of computer_6, enter the following command:

```
cmurefresh -Hmachine_6
```

To communicate the private key password with a prompt for the password, enter the following command:

```
cmurefresh -S
```

The system will respond with the following prompt:

```
Enter password:
```

To communicate the private key password "my_pass" to the FTP daemon FTP02, enter the following command:

```
cmurefresh --sslpass my_pass -N FTP02
```

To communicate with the EDIINT daemon and to refresh the AS2 contract information and the AS2 proxy configuration:

```
cmurefresh -e
```

cmustop Example

To stop a currently running auto connect session identified by the session ID 232, enter the following command:

```
cmustop -s232
```

or

```
cmustop --sid 232
```

cmusession Examples

To review all the FTP sessions currently running, enter the following command:

```
cmusession -f
```

or

```
cmusession --ftp
```

To review all the Bisync sessions currently running, enter the following command:

```
cmusession -b
```

or

```
cmusession --bsc
```

cmutrace Examples

To turn off trace activities for session ID 921, enter the following command:

```
cmutrace -x -s921
```

or

```
cmutrace --off --sid 921
```

Index

D

E

F

G

K

M

P

R

S

T